Ph.D. DISSERTATION

Signal Processing for NAND Flash Memory Reliability Improvement

낸드 플래시 메모리 신뢰도 향상을 위한 신호 처리 방법 연구

BY

DONGHWAN LEE

FEBRARY 2014

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Signal Processing for NAND Flash Memory Reliability Improvement

낸드 플래시 메모리 신뢰도 향상을 위한 신호 처리 방법 연구

지도교수 성 원 용

이 논문을 공학박사 학위논문으로 제출함

2014년 2월

서울대학교 대학원

전기·정보 공학부

이 동 환

이동환의 공학박사 학위논문을 인준함

2014년 2월

위 원 장: 이 재 홍

부위원장: 성 원 용

위 원: 노 종 선

위 원: 이 정 우

위 원: 조 준 호

Abstract

The capacity of NAND flash memory has been continuously increased by aggressive technology scaling and multi-level cell (MLC) data coding. However, it becomes more challenging to maintain the current growth rate of the memory density mainly because of degraded signal quality of sub-20 nm NAND flash memory. This dissertation develops signal processing techniques to improve the signal reliability of MLC NAND flash memory.

In the first part of this dissertation, we develop two threshold voltage distribution estimation algorithms to compensate the effect of program-erase (PE) cycling and charge loss in MLC NAND flash memory. The sensing directed estimation (SDE) utilizes the output of multi-level memory sensing to estimate the means and the variances of the threshold voltage distribution that is modeled as a Gaussian mixture. In order to reduce memory sensing overheads for the SDE algorithm, we develop a decision directed estimation (DDE) that uses error corrected bit patterns for more frequent updates of the model parameters. We also present a combined estimation scheme that employs both the SDE and the DDE approaches to minimize the number of memory sensing operations while maintaining the estimation accuracy. The effectiveness of the SDE and the DDE algorithms is evaluated by using both simulated and real NAND flash memory, and it is demonstrated that the proposed algorithms can estimate the statistical information of threshold voltage distribution accurately.

The cell-to-cell interference (CCI) is one of the major sources of bit errors in

sub-20 nm NAND flash memory and becomes more severe as the size of memory cell decreases. In the second part of this dissertation, we develop a CCI cancellation algorithm that is similar to interference cancellers employed in conventional communication systems. We first provide the experimental characterization of the CCI by measuring the coupling coefficients from actual NAND flash memory with a 26 nm process technology. Then, we present a CCI cancellation algorithm that consists of the coupling coefficient estimation and the CCI removal steps. To reduce the number of memory sensing operations, the optimal quantization schemes for the proposed CCI canceller are also studied.

This dissertation also develops soft-information computation schemes in order to apply soft-decision error correction to NAND flash memory. The probability density function (PDF) of the CCI removed signal is quite different from that of the original threshold voltage, which can be modeled as a Gaussian mixture. Thus, computing soft-information, such as LLR (log likelihood ratio), with the CCI removed signal is not straightforward. We propose two soft-information computation schemes that combine CCI cancellation and soft-decision error correction. In the first approach, we derive a mathematical formulation for the PDF of the CCI removed signal and directly compute the LLR values by using it. In the second approach, CCI cancellation and soft-information computation are jointly conducted. Based on the intensive simulations, it is demonstrated that the reliability of NAND flash memory is significantly improved by applying the proposed signal processing algorithms as well as soft-decision error correction.

Keywords : NAND flash memory, signal processing, threshold voltage distribution estimation, cell-to-cell interference, soft-decision error correction

Student Number : 2009-20856

Contents

List of Figures

List of Tables

Chapter 1

Introduction

High-density NAND flash memory devices are critical components for many applications from mobile devices to solid state drives (SSDs). The advances in the semiconductor process technology have propelled the continued density growth of NAND flash memory, which is well known as Moore's law. Figure 1.1 shows the scaling trend of two-bit MLC (multi-level cell) NAND flash memory [1]. In this figure, we can find that the size of memory cell has been reduced by half approximately every 2.5 years. Besides technology scaling, the MLC data coding scheme that stores more than one bits per cell doubles or even triples the capacity of NAND flash memory. As a result, the capacity of NAND flash memory has been increased nearly 1,000 times during the last decade.

Although the growth rate of NAND flash memory density has been successfully maintained until the feature size of 20 nm, further process technology scaling is considered to be quite challenging due to the reliability issue. The threshold voltage disturbance becomes quite large for sub-20 nm NAND flash memory because the

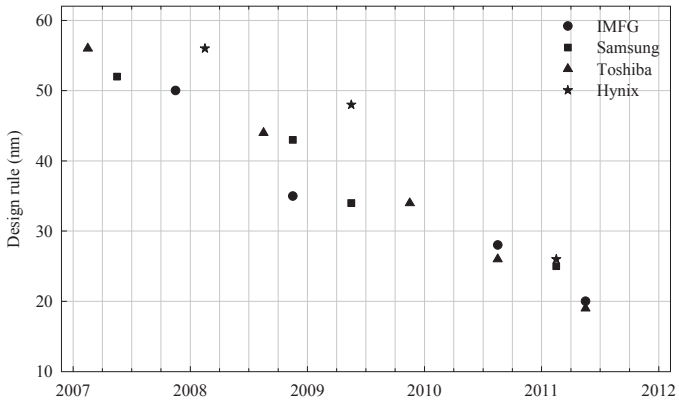


Figure 1.1: Scaling trend of two-bit MLC NAND flash memory [1].

number of charges at each floating gate is too small. Figure 1.2 shows the expected number of electrons per voltage level at the floating gate according to the design rule [1]. For two-bit MLC NAND flash memory with 20 nm process technologies, around 100 electrons (about 25 electrons per level) can be stored at the floating gate, but this number is reduced almost half for the 10 nm one. The number of electrons that discriminates each voltage level is even more reduced by employing MLC data coding, which requires additional voltage margins. As a result, loss of a single electron due to the data retention process causes a substantial change in the threshold voltage distribution.

The cell-to-cell interference (CCI), which is caused by the capacitance coupling, becomes one of the major sources of bit errors for sub-20 nm NAND flash memory. It is well known that the amount of CCI is proportional to threshold voltage shifts of the neighboring cells as well as the coupling coefficients [2]. Since the programming and the erase voltages, which are usually quite high, are not scaled well, the amount of threshold voltage shift of the interfering cell remains almost the same. As the distance between two adjacent cells decreases, however, the coupling coef-

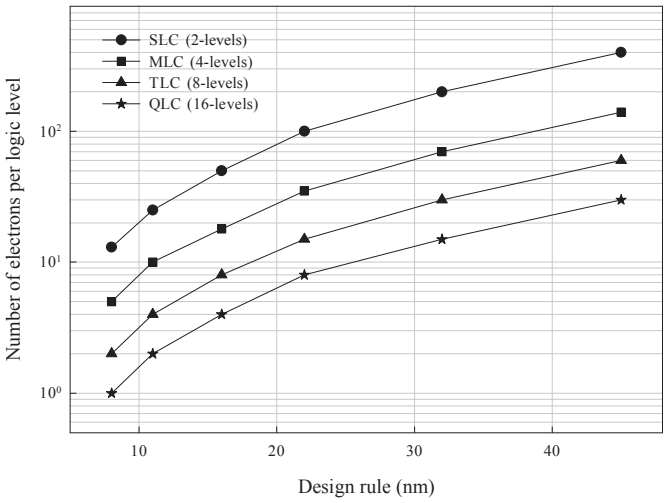


Figure 1.2: The expected numbers of electrons per voltage level at the floating gate of NAND flash memory [1].

ficients become larger, thus the amount of CCI increases rapidly. According to [3], it is expected that the amount of the CCI is over 50 % of the total noise in NAND flash memory with 20 nm process technologies. Although some cell structures have been devised to reduce the CCI, removing the CCI is still very critical for sub-20 nm NAND flash memory devices [3].

In order to solve the reliability problems of NAND flash memory, hard-decision error correction employing BCH (Bose-Chaudhuri-Hocquenghem) or RS (Reed-Solo mon) code has been widely used. This is because these codes can fix a small number of bit errors quite efficiently and are relatively simple to implement [4, 5, 6, 7, 8]. As the decoding capability increases, however, the area and the power consumption of the error correcting circuitry grow rapidly. Thus, hard-decision error correction is not efficient for sub-20 nm NAND flash memory devices where the signal quality is too low [9, 10, 11]. As a result, signal processing techniques as well as soft-decision

error correction are very needed.

This dissertation proposes several signal processing techniques for reliability improvement of high-density NAND flash memory. This dissertation addresses the following techniques:

- Threshold voltage distribution estimation to compensate the effect of charge loss due to data retention process.
- Cell-to-cell interference cancellation to mitigate the effect of capacitance coupling.
- Reliability information computation to apply soft-decision error correction to NAND flash memory.

To provide statistical information on reliability, such as SNR (signal-to-noise ratio), to signal processing and/or error correcting units, we have developed threshold voltage distribution estimation algorithms. In the proposed schemes, the threshold voltage distribution of actual NAND flash memory is modeled as a Gaussian mixture, and the means and the variances of the Gaussian model are found. The sensing directed estimation (SDE) algorithm conducts multi-level memory sensing and utilizes the high precision sensing output to find the statistical information. Since the SDE method requires extra energy consumption and latency for the memory sensing operations, we also have developed a decision directed estimation (DDE) that utilizes error corrected bit patterns.

We also provide the experimental characterization of the CCI that are observed from actual NAND flash memory with a 26 nm process technology and develop a CCI cancellation algorithm that is similar to the interference canceller employed in conventional communication systems. The proposed algorithm consists of the coupling

coefficient estimation and CCI removal steps. The optimal memory sensing schemes for CCI cancellation were also studied. In order to apply soft-decision error correction to NAND flash memory, soft-information, such as LLR (log-likelihood ratio), computation schemes are also studied in this dissertation.

This dissertation is organized as follows. Chapter 2 contains brief introduction to NAND flash memory and modeling of threshold voltage distribution. In Chapter 3, threshold voltage distribution estimation algorithms are proposed. Chapter 4 addresses the statistical characterization of the cell-to-cell interference and proposes a CCI cancellation algorithm. The soft-information computation schemes with/without CCI cancellation are explained in Chapter 5. Finally, Chapter 6 concludes this dissertation.

Some of materials in this dissertation were presented in [12, 11, 13, 14, 15, 16, 17].

Chapter 2

NAND Flash Memory Basics

This chapter contains a brief review of NAND flash memory. In Section 2.1, the memory structures, the multi-page programming scheme, and various noises of NAND flash memory are explained. The threshold voltage distribution of two-bit MLC (multi-level cell) NAND flash memory and its modeling are shown in Section 2.2.

2.1 Basics of NAND Flash Memory

2.1.1 NAND Flash Memory Structure

A block in NAND flash memory is a two-dimensional cell array that consists of multiple word- and bit-lines as shown in Fig. 2.1. According to the organization of bit-lines, NAND flash memory can be categorized as either the even/odd bit-line or the all bit-line structure. In the even/odd bit-line structure, the cells on the even bit-lines form even pages, while those on the odd bit-lines become odd ones [18, 19]. On the other hand, there is no such distinction in the all bit-line structure [20, 21]. The

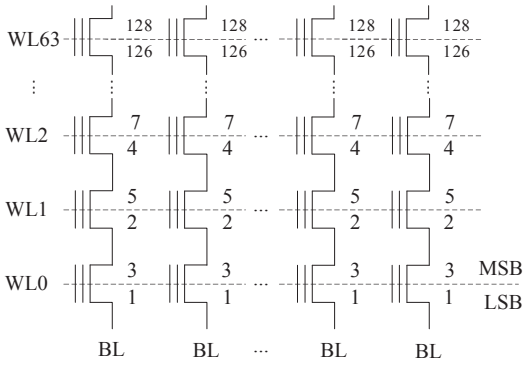


Figure 2.1: Two-bit MLC NAND flash memory structure (all bit-line) and the programming order.

even/odd structure requires less hardware than the all bit-line one because the page buffers and the peripheral circuits are shared by the even and the odd pages. However, the all bit-line structure induces smaller cell-to-cell interference (CCI), which is one of the major sources of bit errors in sub-20 nm NAND flash memory, than the even/odd one.

2.1.2 Multi-Page Programming

In MLC NAND flash memory, the multi-page programming scheme is widely used to reduce the variance of the threshold voltage signals while achieving a high write throughput [19]. In this programming scheme, LSB (least significant bit) and MSB (most significant bit) pages are programmed sequentially as shown in Fig. 2.2. During the LSB page programming, a temporal state (‘0’ state in Fig. 2.2) is used instead of ‘00’ and ‘10’ states, and the result is similar to that of SLC (single-level cell) NAND flash memory. At the MSB page programming step, the cells on the temporal state are programmed to either the symbol *00* or *10*, while those of the erased one (‘1’ state) are programmed to the symbol *01* or remain as the symbol *11*. By adopting the

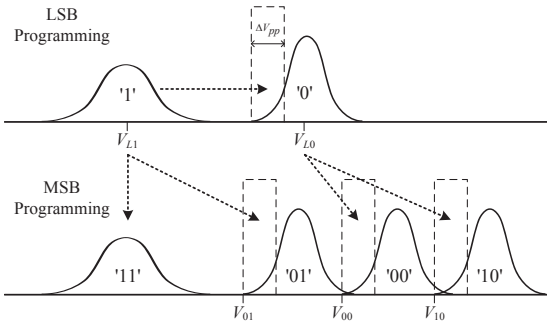


Figure 2.2: Multi-page programming scheme.

multi-page programming method, the threshold voltage shift during the MSB programming is reduced much when compared to the method that directly changes the threshold voltage from the erased one to the symbol *01*, *00*, or *10*. Since the amount of CCI is proportional to the amount of threshold voltage shift for the interfering cell during the MSB programming, the multi-page programming scheme is advantageous in reducing the CCI.

When the cells are programmed, the incremental stair pulse programming (ISPP) is widely used to achieve a tight threshold voltage bound [22]. In the ISPP, a verification process is followed by the incremental programming to ensure that the threshold voltage of the programmed cell is higher than the target voltage. In this programming scheme, the threshold voltage of each target cell increases as much as ΔV_{pp} and is compared to the target voltage at each iteration. If the programmed voltage is higher than the target one, the programming operation stops. It is well known that the ideal ISPP results in a uniform distribution as shown in Fig. 2.2. In this dissertation, the target voltages are denoted as V_{01}, V_{00}, and V_{10} depending on the symbol of the programmed cell.

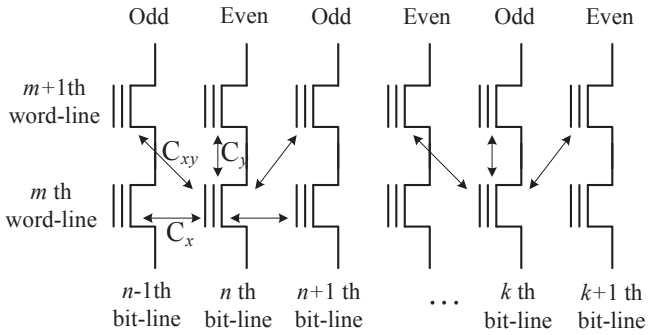


Figure 2.3: Cell-to-cell interference model in the even/odd bit-line structure.

2.1.3 Cell-to-Cell Interference

The cell-to-cell interference is caused by the parasitic capacitor coupling effect between the adjacent cells. Thus, when one cell is programmed, the threshold voltages of both the target and the surrounding cells increase [2]. During the MSB programming of the target cell, the ISPP neutralizes the CCI induced by the previously programmed neighboring cells. Thus, only the surrounding cells that are programmed after the victim cell cause the CCI. The number of interfering cells depends on the bit-line structure. In the even/odd bit-line structure where the even cells are programmed earlier than the odd ones, the even victim cells are affected by not only the three cells from the next word-line, which is the $(m+1)$-th word-line in Fig. 2.3, but also the two cells on the same word-line. On the other hand, the odd victim cells receive the interference only from the three cells on the next word-line [18]. In the all bit-line structure, in which the even and the odd pages are unified, a victim cell receives the CCI only from the three cells on the next word-line [20, 21].

The amounts of interference that the (m,n)-th (on the even bit-line) and (m,k)-th (on the odd bit-line) victim cells receive, where m denotes the word-line index and n

and k represent the bit-line indices, can be represented as the linear combinations of the threshold voltage shifts of neighboring cells. Thus, the CCI for the even and the odd victim cells become

$$V_{CCI,even}[m,n] = C_x \cdot (\Delta V[m,n-1] + \Delta V[m,n+1]) + C_y \cdot \Delta V[m+1,n] \\ + C_{xy} \cdot (\Delta V[m+1,n-1] + \Delta V[m+1,n+1]) \quad (2.1)$$

and

$$V_{CCI,odd}[m,k] = C_{xy} \cdot (\Delta V[m+1,k-1] + \Delta V[m+1,k+1]) + C_y \cdot \Delta V[m+1,k], \quad (2.2)$$

respectively, where $\Delta V[m,n-1]$ is the threshold voltage shift of the left neighboring cell for the (m,n)-th victim cell, and so on. The coefficients, C_x, C_y, and C_{xy}, are the coupling ratio, and they are determined by the geometry and the dielectric constant of the material for spacing.

2.1.4 Data Retention

The data retention problem is caused by charge loss at the floating gate of each memory cell. As the feature size of NAND flash memory decreases, the number of electrons at each cell's floating gate becomes smaller, and as a result, the data retention induced distortion becomes more serious [23, 24]. As illustrated in Fig. 2.4, the data retention process not only negatively shifts the threshold voltage but also increases the variance of the distribution.

According to [24], the data retention induced noise can be modeled as a Gaussian

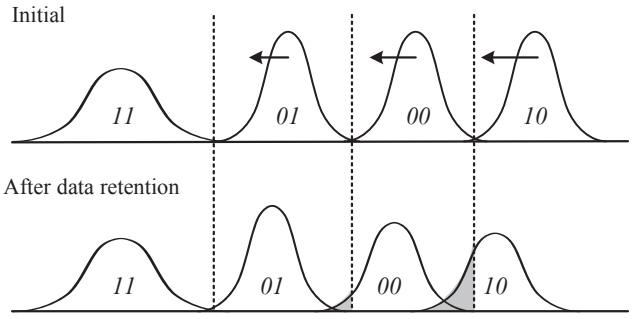


Figure 2.4: Threshold voltage distribution shift due to data retention in MLC NAND flash memory.

random variable. The mean and the standard deviation shifts are

$$m_r = -\alpha_m(N_{pe}, v_i) \cdot \ln\left(1 + \frac{t}{t_0}\right) \tag{2.3}$$

and

$$\sigma_r = \alpha_\sigma(N_{pe}, v_i) \cdot \ln\left(1 + \frac{t}{t_0}\right), \tag{2.4}$$

respectively, where α_m and α_σ are positively valued functions of the number of PE (program-erase) cycles N_{pe} and the initial threshold voltage v_i. Note that t_0 is the initial time and can be set to 1 hour. In this model, the mean and the standard deviation shifts are expressed via power law functions of the initial threshold voltage and the number of PE cycles. Also, more importantly, both the mean and the standard deviation shifts are proportional to the logarithms of the data retention time. Thus, we have

$$\sigma_r = \beta(N_{pe}, v_i) \cdot m_r. \tag{2.5}$$

From (2.5), we can find that σ_r is linearly proportional to m_r, and $\beta(N_{pe}, v_i)$ is the

coefficient of proportionality. For simplicity, we denote $\beta(N_{pe}, v_i)$ as β in this dissertation.

2.2 Threshold Voltage Distribution of NAND Flash Memory and Signal Modeling

This section illustrates the threshold voltage distributions obtained from actual two-bit MLC NAND flash memory devices with 20 nm process technologies, and also describes the signal modeling.

2.2.1 Threshold Voltage Distribution and Gaussian Approximation

The threshold voltage distributions of NAND flash memory cells with various PE cycling and data retention conditions are shown in Fig. 2.5. The threshold voltages are measured with the precision of 0.1 V by using a manufacturer defined function that can alter the memory sensing reference voltages (MSRVs). The initial threshold voltage distribution is obtained from the fresh memory cells. The threshold voltage distributions of non-initial states were obtained after 1.0 K, 1.5 K, or 3.0 K times of PE cycling and 10 hours of baking process at 125 °C, which is equivalent to one year data retention. Since the memory device cannot measure the voltages below -1 V, this voltage region is not drawn in this figure. As the number of PE cycles grows, the distribution becomes wider, which significantly increases the raw bit error rate (RBER) of the memory devices. Also, the baking process or the data retention causes negative shifts of the threshold voltage distributions.

The threshold voltage distribution of NAND flash memory cells can be approximated to a mixture of Gaussian distributions. When the threshold voltage distribution

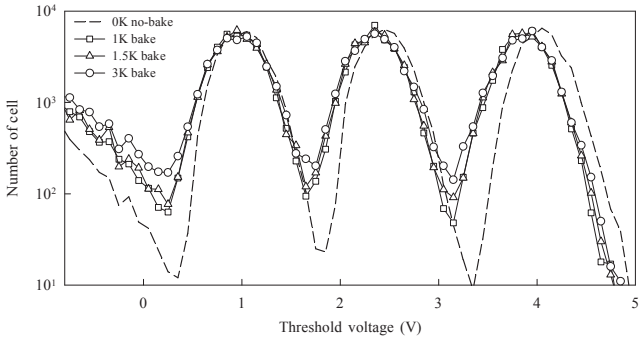


Figure 2.5: Threshold voltage distribution of real NAND flash memory according to the number of PE cycles.

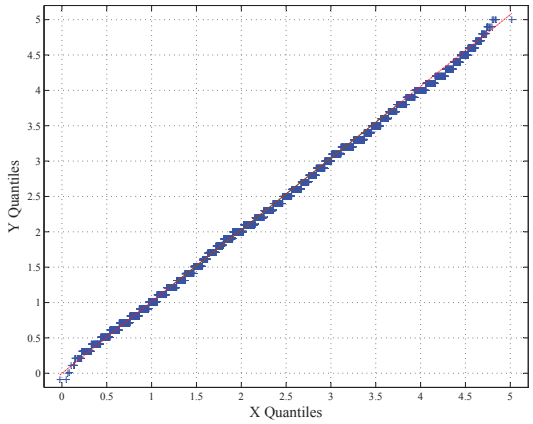


Figure 2.6: Q-Q plot for the threshold voltage distribution versus its Gaussian mixture model.

of each symbol is modeled as a Gaussian function, the likelihood function for a given input symbol j can be written as

$$f(y|x=j) = \frac{1}{\sqrt{2\pi\sigma_j^2}} e^{-\frac{(y-m_j)^2}{2\sigma_j^2}}, \tag{2.6}$$

for $j = 0, 1, 2,$ and 3. Note that x is the input symbol and y represents the threshold

voltage, while m_j and σ_j are the mean and standard deviation of the Gaussian distribution, respectively. In order to compare the true threshold voltage distribution with its Gaussian mixture model, the Q-Q plot is illustrated in Fig. 2.6. In this figure, the x quantiles are sampled from the modeled distribution, while the y values are the observed threshold voltages. Since the points in the Q-Q plot almost lie on the line of $y = x$, we consider that the two distributions are quite close [25]. Since a Gaussian distribution is completely characterized by only two parameters, the approximation using the Gaussian mixture simplifies the modeling process very much.

2.2.2 Modeling of Threshold Voltage Signal

The output signal obtained from two-bit MLC NAND flash memory can be modeled as shown in Fig. 2.7. Before programming the memory cells, an erase operation is needed to remove charges stored in each cell's floating-gate. The threshold voltage distribution of the erased cells tends to be the Gaussian due to the variability in the erase process [26]. When programming the memory cells, the multi-page programming explained in Section 2.1 is used. In this programming scheme, the LSB pages are programmed before the MSB ones. The programming operations in NAND flash memory induce the CCI. We denote the output signal after the LSB programming as V_L. The mean of V_L is either V_{L0} or V_{L1} depending on the cell's LSB as shown in Fig. 2.2. When the MSB page programming is conducted, the threshold voltage becomes V_M. The target cell is also affected by the CCI that is induced during the MSB page programming of neighboring cells. Note that we denote the amount of interference as V_{CCI}. The data retention problem results in negative shift of the threshold voltage, V_R. Usually, the CCI increases the threshold voltages, and its magnitude becomes larger as the feature size of memory cells decreases. On the other hand, V_R has a negative

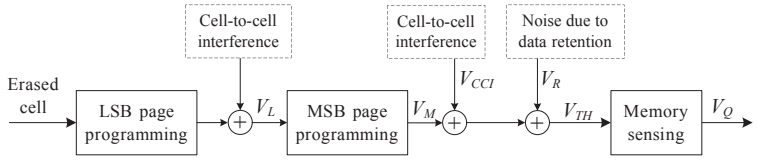


Figure 2.7: Signal modeling for two-bit MLC NAND flash memory.

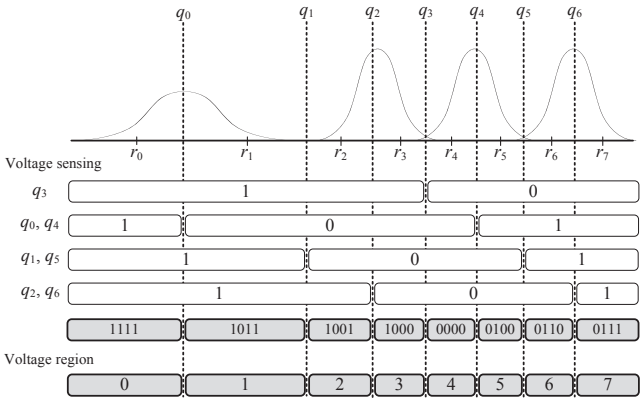
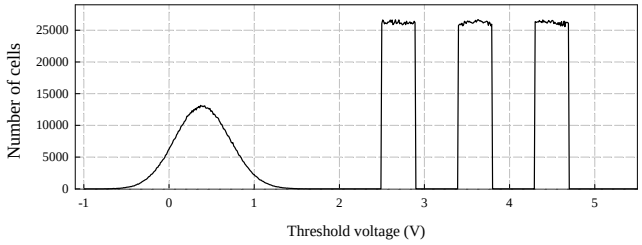


Figure 2.8: 8-level voltage sensing in two-bit MLC NAND flash memory.

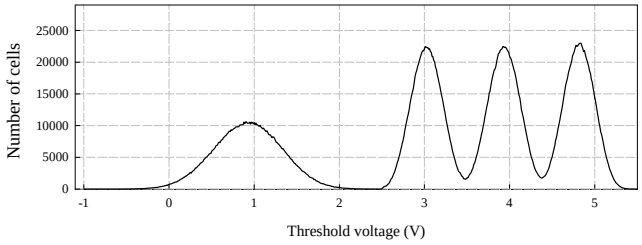
value, and its magnitude increases with the number of PE cycles and the retention time. After experiencing charge loss, the threshold voltage of one cell becomes

$$V_{TH} = V_M + V_{CCI} + V_R. \tag{2.7}$$

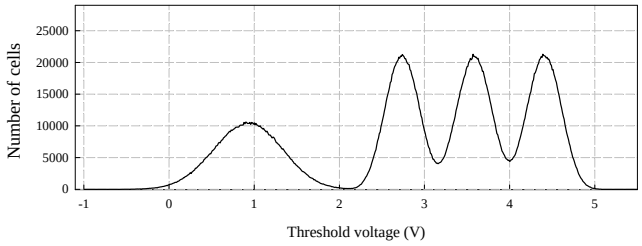
The memory read operation induces the quantization effect on the threshold voltage signal, and as a result, V_Q is observed after memory sensing operations. Note that V_Q depends on the number of memory sensing operations. Figure 2.8 illustrates 8-level memory sensing in two-bit MLC NAND flash memory. While changing the reference voltage from q_0 to q_6, the memory sensing results, which determine the



(a) Ideal programming



(b) After cell-to-cell interference



(c) After data retention

Figure 2.9: Threshold voltage distributions of simulated two-bit MLC NAND flash memory.

voltage regions of the cells, are obtained. For example, the memory sensing output of '1001' indicates that the cell has a threshold voltage in the range of $(q_1, q_2]$, and so on. Each voltage region has one of eight representative values $r_0 \sim r_7$, which becomes the quantized threshold voltage V_Q.

In order to visualize the above modeling process, an example of simulated threshold voltage distribution is shown in Fig. 2.9. One block of hypothetical NAND flash memory has 64 K bit-lines and 64 word-lines. The erase operation yields the symbol

11 whose PDF is a Gaussian distribution with the mean and the standard deviation of -2.0 V and 0.40 V, respectively. The symbols *01*, *00*, and *10* are programmed with the target voltages of 0.4 V, 1.9 V, and 3.5 V, respectively. We assume that the coupling coefficients for the *x*, *y*, and *xy* directions are the Gaussian random variables whose means are 0.0810, 0.1231, and 0.023, respectively. The standard deviation values of the coupling coefficients are set to 20 % of their means. Note that the mean and standard deviation values of the coupling coefficients are obtained from actual NAND flash memory chips with a 26 nm process technology, which will be explained in Section 4.2. The data retention induced noise is approximated as a Gaussian random variable according to [27].

Chapter 3

Threshold Voltage Distribution Estimation

3.1 Introduction

The reliability of data in NAND flash memory employing sub-20 nm process technology is seriously challenged by the data retention problem. As the memory cell size decreases, not many electrons are stored at the floating-gate of each cell, and as a result, even a small number of leakage charges can cause significant distortion in the threshold voltage distribution [3]. Especially, this problem becomes more serious as MLC (multi-level cell) and TLC (triple-level cell) data coding schemes are employed for density increase. Since the data retention process not only widens but also shifts the threshold voltage distribution, it is difficult to control the amount of distortion by employing the ISPP (incremental step pulse programming) [22] that is widely used for minimizing the PE (program-erase) cycling induced noises.

Several works have been conducted to solve the data retention problem. In the

flash correct-and-refresh (FCR) scheme, every programmed page is periodically read and remapped before it accumulates data retention induced bit errors [28]. This scheme also reprograms or refreshes the erroneous memory cells to conduct remapping less frequently. However, reprogramming itself induces cell-to-cell interference (CCI), which is one of the major sources of bit errors in sub-20 nm NAND flash memory. Thus, the FCR method yields diminishing returns as the amount of CCI becomes larger in high-density NAND flash memory. Most of all, the FCR scheme can only be applied to systems that are always powered on. In [29], a moving read technique that adjusts the memory sensing reference voltages (MSRVs) was proposed. This technique observes the changes in the number of cells that are sensed as the symbol *10*, which corresponds to the highest threshold voltage level in MLC NAND flash memory. Even though this method shows improved BER (bit error rate) performance when applied to hard-decision error correction, it does not provide SNR (signal-to-noise ratio) information that is essential for soft-decision error correction [30]. Note that conventional hard-decision error correcting algorithms, such as BCH (Bose-Chaudhuri-Hocquenghem) and RS (Reed-Solomon) codes, are no more efficient for high-density NAND flash memory devices [5, 8].

In this chapter, we have developed parameter estimation algorithms to find the statistical information of the threshold voltage distribution. The sensing directed estimation (SDE) algorithm approximates the threshold voltage distribution as a Gaussian mixture and finds the best-fit mean and standard deviation values by comparing the actual distribution and its model. The SDE scheme does not utilize any pilots or known bit patterns but employs extra memory sensing operations. Since the SDE algorithm requires extra delay and energy consumption for memory sensing operations, we also develop a decision directed estimation (DDE) algorithm that does not demand

any sensing overheads. The DDE algorithm also adopts the Gaussian mixture model and compares the input and the output data of the error correction circuit to find the best-fit parameters of the model. Since the DDE method assumes successful error correction, we need to use the SDE algorithm when the error correction fails. To this end, we also propose a combined threshold voltage distribution estimation scheme that utilizes both the SDE and DDE methods. The proposed algorithms are evaluated by using the data samples obtained from both simulated and actual NAND flash memory, and the accuracy of estimated means and standard deviations is assessed.

This chapter is organized as follows. In Section 3.2, we propose a sensing directed estimation algorithm and show experimental results. A decision directed estimation algorithm is proposed in Section 3.3. The experimental results for the DDE algorithm are also shown in this section. Finally, concluding remarks are made in Section 3.4.

3.2 Sensing Directed Estimation of Threshold Voltage Distribution

One simple approach for estimating the threshold voltage distribution is to build a relative frequency histogram by conducting memory sensing operations many times while changing the reference voltage. For example, if the MSRV is altered by 0.1 V at each sensing operation, we can obtain a quite accurate threshold voltage distribution, but this straightforward approach demands more than 50 times of memory sensing operations. Apparently, it is desired to reduce the number of memory sensing operations for the sake of minimizing the access time and energy consumption. According to [31], one voltage sensing operation takes approximately 15 μs and consumes the energy of 0.81 μJ per page for a 20 nm-class NAND flash memory device whose

page size is 8 KBytes.

In this section, we propose the sensing directed estimation algorithm to estimate the mean and standard deviation values that are needed to model the threshold voltage distribution. The measured data are obtained by conducting a fairly small number of voltage sensing operations for a page with different MSRVs and used to determine the parameters of the Gaussian mixture model. We use the gradient descent (GD) and the Levenberg-Marquardt (LM) methods for the parameter search. The proposed SDE algorithms are invoked only when the currently stored means and standard deviations are no longer valid due to PE cycling and/or the data retention induced distortion, which causes error correction failures or too many iterations for iterative decoding [11]. If the proposed algorithms find accurate signal statistics, more reliable hard-decision data or soft-information can be fed to a hard- or soft-decision decoder, thus error correction is more likely to be successful.

3.2.1 Cost Function

The cost function is needed to guide the search algorithms by assessing the closeness between the data and the model. In this work, the cost function is defined by using the squared Euclidean distance between the true distribution and the modeled one. Consider two-bit NAND flash memory whose threshold voltage distribution is shown in Fig. 3.1. By conducting seven voltage sensing operations with the reference voltages of $q_1, q_2, \cdots$, and q_7, we can divide the voltage region into eight levels. Note that q_0 and q_8 correspond to $-\infty$ and ∞, respectively. Let us define Nr_i (for $i = 0, 1, 2, \cdots$, and 7) as the number of cells in a page whose threshold voltages are in the range of $(q_i, q_{i+1}]$, then it can be determined by counting the number of cells at each voltage level. This process is equivalent to building a relative frequency histogram for the

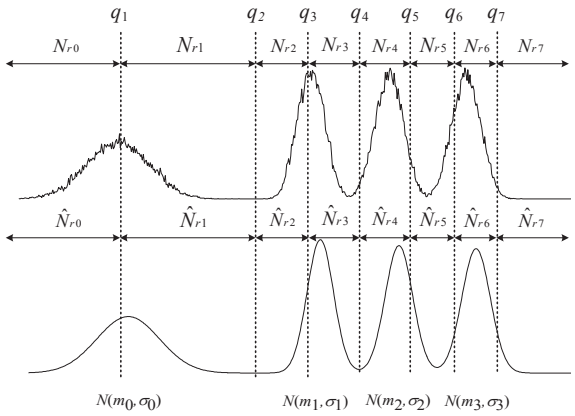


Figure 3.1: Pictorial representation of the cost function.

threshold voltage. Similarly, let $\hat{N}r_i$ be the estimator of Nr_i, and it provides the relative frequency histogram obtained from the Gaussian mixture model. If the modeled distribution is close enough to the measured one, the difference between Nr_i and $\hat{N}r_i$ will be minimized.

In the Gaussian mixture model, $\hat{N}r_i$ can be computed as

$$\hat{N}r_i = Nw_0 \cdot \left\{ Q(\frac{q_i - m_0}{\sigma_0}) - Q(\frac{q_{i+1} - m_0}{\sigma_0}) \right\} + Nw_1 \cdot \left\{ Q(\frac{q_i - m_1}{\sigma_1}) - Q(\frac{q_{i+1} - m_1}{\sigma_1}) \right\} \tag{3.1}$$

$$+ Nw_2 \cdot \left\{ Q(\frac{q_i - m_2}{\sigma_2}) - Q(\frac{q_{i+1} - m_2}{\sigma_2}) \right\} + Nw_3 \cdot \left\{ Q(\frac{q_i - m_3}{\sigma_3}) - Q(\frac{q_{i+1} - m_3}{\sigma_3}) \right\},$$

where

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} \exp(-\frac{u^2}{2}) du, \tag{3.2}$$

In Eq. (3.1), m_j and σ_j are the mean and standard deviation values of the threshold voltage for the input symbol j, respectively. Nw_j is the number of cells that were written as the symbol j during the programming operation. In this work, we assume

that Nw_j is counted before the programming operation and written at the spare region of the page. By letting $z_{i,j} = \frac{q_i - m_j}{\sigma_j}$, we can simplify Eq. (3.1) as

$$\hat{Nr}_i = \sum_{j=0}^{3} Nw_j \cdot \left\{ Q(z_{i,j}) - Q(z_{i+1,j}) \right\}. \tag{3.3}$$

When we apply memory sensing operations with the MSRVs of q_1 to q_{Ns-1}, the voltage region is divided into N_s distinct ones. In the determination of q_i, the currently stored mean and variance values that were obtained during the latest parameter estimation process can be used, thus choosing q_i is inherently an adaptive process. In this work, the hard-decision boundaries as well as the points that evenly divide these boundaries are used as q_i. Using the above equations, we can define a cost function as

$$C_{Ns} = \frac{1}{2} \sum_{i=0}^{N_s-1} \left(\frac{Nr_i - \hat{Nr}_i}{N} \right)^2, \tag{3.4}$$

where N is the total number of cells in each page. The cost function corresponds to the squared Euclidean distance between the two histograms [32]. Note that the cost function also depends on the number of voltage levels N_s. For example, C_8 and C_{12} are two different cost functions. By minimizing the cost function, we can find the best-fit parameters of the Gaussian mixture model.

$$(m^*_{0\sim3}, \sigma^*_{0\sim3}) = \arg\min C_{Ns}(m_{0\sim3}, \sigma_{0\sim3}) \tag{3.5}$$

3.2.2 Gradient Descent Method based Parameter Search

Since the exhaustive search to find the global minimum of the cost function is impractical, we employ the gradient descent method, which is one of the simplest optimiza-

tion techniques. First, let $\mathbf{x} = [m_0, \quad \cdots, \quad m_3, \quad \sigma_0, \quad \cdots, \quad \sigma_3]^T$ be the parameter vector in Eq. (3.5), then the cost function can be rewritten as

$$C_{Ns}(\mathbf{x}) = \frac{1}{2} G_{Ns}(\mathbf{x})^T \cdot G_{Ns}(\mathbf{x}), \text{ where } G_{Ns}(\mathbf{x}) = \frac{1}{N} \begin{bmatrix} Nr_0 - \hat{N}r_0(\mathbf{x}) \\ Nr_1 - \hat{N}r_1(\mathbf{x}) \\ \vdots \\ Nr_{N_s-1} - \hat{N}r_{N_s-1}(\mathbf{x}) \end{bmatrix}. \quad (3.6)$$

The gradient descent method produces a sequence $\mathbf{x}^{(k)}$ that minimizes $C_{Ns}(\mathbf{x})$ by using the following equation:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \mu \nabla C_{Ns}(\mathbf{x}^{(k)}). \quad (3.7)$$

Note that μ represents the constant step size. The gradient of $C_{Ns}(\mathbf{x})$ or $\nabla C_{Ns}(\mathbf{x})$ is equal to $J_{G_{Ns}}(\mathbf{x})^T \cdot G_{Ns}(\mathbf{x})$, where $J_{G_{Ns}}(\mathbf{x})$ denotes the Jacobian matrix of $G_{Ns}(\mathbf{x})$. By reformulating the equations, we can obtain

$$J_{G_{Ns}}(\mathbf{x})^T = \frac{1}{N} \begin{bmatrix} \alpha(z_{0,0}) & \alpha(z_{1,0}) & \cdots & \alpha(z_{N_s-1,0}) \\ \vdots & \vdots & \vdots & \vdots \\ \alpha(z_{0,3}) & \alpha(z_{1,3}) & \cdots & \alpha(z_{N_s-1,3}) \\ \beta(z_{0,0}) & \beta(z_{1,0}) & \cdots & \beta(z_{N_s-1,0}) \\ \vdots & \vdots & \vdots & \vdots \\ \beta(z_{0,3}) & \beta(z_{1,3}) & \cdots & \beta(z_{N_s-1,3}) \end{bmatrix}, \quad (3.8)$$

where

$$\alpha(z_{i,j}) = -\frac{Nw_j}{\sqrt{2\pi}\sigma_j}(e^{-\frac{z_{i,j}{}^2}{2}} - e^{-\frac{z_{i+1,j}{}^2}{2}}) \quad (3.9)$$

and

$$\beta(z_{i,j}) = -\frac{Nw_j}{\sqrt{2\pi}\sigma_j}(z_{i,j}e^{-\frac{{z_{i,j}}^2}{2}} - z_{i+1,j}e^{-\frac{{z_{i+1,j}}^2}{2}}). \tag{3.10}$$

If N_s is equal to 4, $J_{G_{Ns}}(\mathbf{x})^T$ is an 8 by 4 matrix whose rank is at most 4. Thus, $\nabla C_{Ns}(\mathbf{x})$ $(= J_{G_{Ns}}(\mathbf{x})^T \cdot G_{Ns}(\mathbf{x}))$ can be a zero vector even if $G_{Ns}(\mathbf{x})$ is not a zero vector, which means that the proposed algorithm can converge to non-optimal points. To make $J_{G_{Ns}}(\mathbf{x})$ be a full rank matrix, N_s needs to be larger than or equal to 8.

The entire algorithm is shown in Algorithm 1. Note that $\|\nabla C_{Ns}(\mathbf{x})\|^2$ is used as the stopping criterion, thus the iteration stops when this value is smaller than η. The initial trial parameter is denoted as $\mathbf{x}^{(\mathbf{0})}$, and the currently stored mean and standard deviation values can be used for it. For example, if the proposed estimation methods are invoked for the first time, the means and standard deviations of initial threshold voltage distribution are used for $\mathbf{x}^{(\mathbf{0})}$. Since the GD based method uses only the first order derivatives, it usually converges slowly.

Algorithm 1 Gradient descent method based parameter search.

Initialization: $\mu = 1.0$ and $\mathbf{x} = \mathbf{x}^{(\mathbf{0})}$
$k \leftarrow 0$
while $\|\nabla C_{Ns}(\mathbf{x})\|^2 \geq \eta$ and $k < Max\_iter$ **do**
  compute $\nabla C_{Ns}(\mathbf{x})$ using Eq. (3.8)
  $\mathbf{x} \leftarrow \mathbf{x} - \mu \nabla C_{Ns}(\mathbf{x})$
  $k \leftarrow k + 1$
end while

3.2.3 Levenberg-Marquardt Method based Parameter Search

The Levenberg-Marquardt method is a hybrid of the gradient descent and the Newton algorithms, and it is widely used in many applications for solving non-linear least

squares problems [33]. Usually, the LM method shows faster convergence speed than the GD one.

The LM optimization method operates as follows [33, 34]:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - (H_{G_{Ns}}(\mathbf{x}) + \lambda I)^{-1} \cdot \nabla C_{Ns}(\mathbf{x}^{(k)}). \tag{3.11}$$

Note that $H_{G_{Ns}}(\mathbf{x})$ and I represent the Hessian of $G_{Ns}(\mathbf{x})$ and an 8 by 8 identity matrix, respectively, and λ is a positively valued step size. In order to compute $H_{G_{Ns}}(\mathbf{x})$ exactly, the second order derivatives are required, which demands a high computational overhead. However, $H_{G_{Ns}}(\mathbf{x})$ can be approximated to $J_{G_{Ns}}(\mathbf{x})^T \cdot J_{G_{Ns}}(\mathbf{x})$, thus Eq. (3.11) can be rewritten as

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - (J_{G_{Ns}}(\mathbf{x})^T \cdot J_{G_{Ns}}(\mathbf{x}) + \lambda I)^{-1} \cdot \nabla C_{Ns}(\mathbf{x}^{(k)}). \tag{3.12}$$

During each iteration, the step size λ is updated, and we adopt a simple step size updating algorithm presented in [35]. The LM method based parameter estimation is described in Algorithm 2. If the estimation error value of the current parameter vector, $C_{Ns}(\mathbf{x_{lm}})$, is smaller than that of the previous one, err_p, we accept the update and decrease λ by a factor of ν. Otherwise, we retract the update and increase λ by the same factor. In this work, ν is set to 10 by referring [35].

Table 3.1 shows the number of arithmetic operations for each iteration of the GD and the LM based parameter estimation algorithms. Note that except for the matrix inversion, small constant terms are ignored. We assume that the exponential and the Q-functions are implemented by using the look-up tables (denoted as 'LUT' in Table 3.1). The total number of arithmetic operations for the GD based algorithm is the

Algorithm 2 Levenberg-Marquardt method based parameter search.

Initialization: $\lambda = 0.1$, $\nu = 10$ and $\mathbf{x} = \mathbf{x}^{(\mathbf{0})}$
$updateFlag \leftarrow 1$
$err_p \leftarrow C_{Ns}(\mathbf{x})$
$k \leftarrow 0$
while $\|\nabla C_{Ns}(\mathbf{x})\|^2 \geq \eta$ and $k < Max\_iter$ **do**
  if $updateFlag == 1$ **then**
    compute $J_{G_{Ns}}(\mathbf{x})$, $G_{Ns}(\mathbf{x})$ and $\nabla C_{Ns}(\mathbf{x})$
  end if

  solve $(J_{G_{Ns}}(\mathbf{x})^T \cdot J_{G_{Ns}}(\mathbf{x}) + \lambda I) \cdot \triangle \mathbf{x} = -\nabla C_{Ns}(\mathbf{x})$
  $\mathbf{x_{lm}} \leftarrow \mathbf{x} + \triangle \mathbf{x}$
  $err \leftarrow C_{Ns}(\mathbf{x_{lm}})$

  if $err < err_p$ **then**
    $\mathbf{x} \leftarrow \mathbf{x_{lm}}$, $\lambda \leftarrow \frac{\lambda}{\nu}$, $err_p \leftarrow err$
    $updateFlag \leftarrow 1$
  else
    $\lambda \leftarrow \nu\lambda$
    $updateFlag \leftarrow 0$
  end if
  $k \leftarrow k+1$
end while

Table 3.1: The number of arithmetic operations for each iteration of the GD and the LM based parameter estimation algorithms

	ADD	MUL	DIV	LUT
$z_{i,j}$, $Q(z_{i,j})$, $\exp(\frac{z_{i,j}^2}{2})$	$4N_s$	0	$4N_s$	$8N_s$
$\alpha(z_{i,j})$, $\beta(z_{i,j})$	$8N_s$	$12N_s$	0	0
$G_{Ns}(\mathbf{x})$	$8N_s$	$4N_s$	0	0
$\nabla C_{Ns}(\mathbf{x})$, $\|\nabla C_{Ns}(\mathbf{x})\|^2$	$9N_s$	$9N_s$	0	0
GD total	$29N_s$	$25N_s$	$4N_s$	$8N_s$
LM total	$93N_s + \lceil \frac{8^3}{3} \rceil$	$89N_s + \lceil \frac{8^3}{3} \rceil$	$4N_s + \lceil \frac{8^3}{3} \rceil$	$8N_s$

sum of data in the first four rows. In the LM based method, the number of arithmetic operations for the matrix-matrix multiplication ($64N_s$) and the matrix inversion ($\lceil \frac{8^3}{3} \rceil$) is added to that of the GD based one. We assume that the matrix inversion employs the Gaussian-Jordan elimination.

3.2.4 Experimental Results

We conducted experiments in order to evaluate the performance of the SDE algorithms. We used data samples that were obtained from the simulated NAND flash memory model and the actual NAND flash memory devices. The simulated two-bit MLC NAND flash memory is generated according to the model described in Chapter 2.2 [36, 27], while NAND flash memory with a 20 nm technology is used for the real devices. In both cases, we assume the worst case scenario that the parameter estimation process is invoked for the first time, thus the means and variances of initial threshold voltage distribution for NAND flash memory devices are used for $\mathbf{x}^{(\mathbf{0})}$.

3.2.4.1 Parameter Estimation with Simulated NAND Flash Memory

To study the convergence capability and speed, the proposed estimation methods are applied to simulated NAND flash memory whose PE cycle and retention time are set to 5,000 times and 64 K hours (7.3 years), respectively. Note that most two-bit MLC flash memory devices only allow 3,000 times of PE cycles [37]. In this condition, the mean values of the symbol 1, 2, and 3 are shifted as much as -0.36 V, -0.48 V, and -0.60 V, respectively, as shown in Fig. 3.2. The standard deviations are changed as much as 0.027 V, 0.037 V, and 0.048 V, respectively. While changing N_s from 8 to 16, we assess the convergence characteristics of the proposed algorithms. The vertical lines in Fig. 3.2 represent the sensing reference voltages. Among them, the solid lines

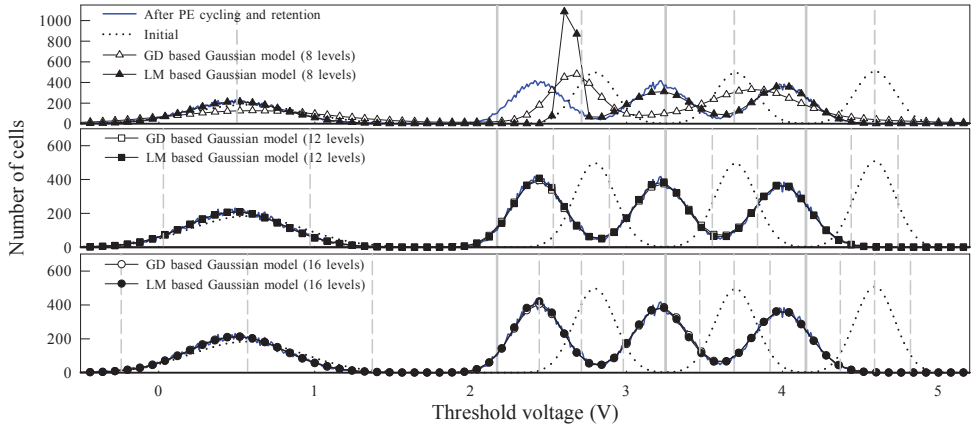


Figure 3.2: The simulated and estimated threshold voltage distributions of NAND flash memory with PE cycles of 5,000 times and the data retention time of 64K hours.

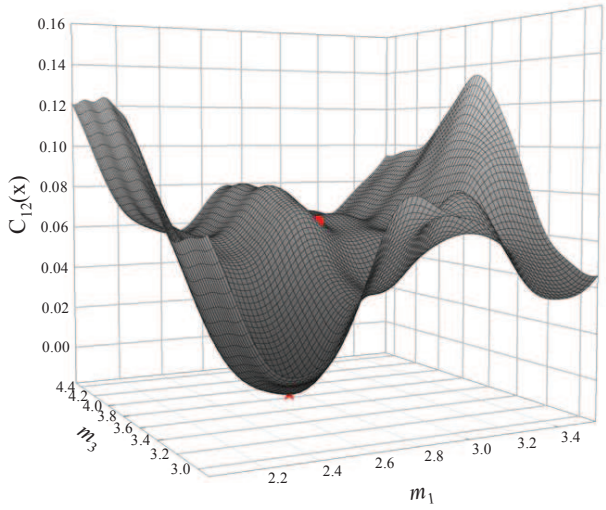
are the hard-decision boundaries of the initial distribution, while the other reference voltages (the dotted lines) are determined by uniformly dividing two adjacent hard-decision boundaries.

The distributions of simulated (blue solid curves), initial (dotted curves), and estimated (curves with markers) threshold voltages are shown in Fig. 3.2. When the number of voltage levels is larger than or equal to 12, the estimated distributions follow the simulated ones very closely, and we can find that both estimation algorithms converge to the global optimum points. However, no algorithm can find the optimum points when the number of voltage levels is 8.

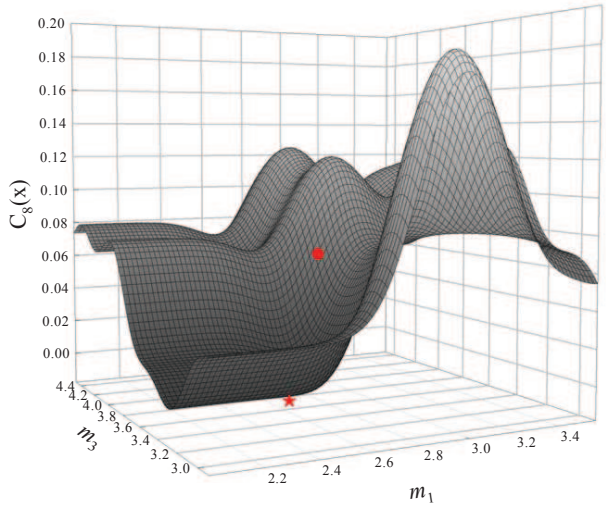
In Fig. 3.2, it is shown that the threshold voltage distribution shifts to the left and widens quite substantially after 64 K hours of data retention. When the distribution is extremely left-shifted, the memory sensing operations with high reference voltages yield no additional information about the true distribution. It is because the number of cells between these voltage levels is almost zero. We can consider the

voltage level with no information as the degenerate one. For example, when N_s is 16, almost no cells exist in the three right-most voltage levels, and they provide not much additional information for the estimation. When the ith voltage level is the degenerate one, the ith column of $J_{G_{Ns}}(\mathbf{x})^T$ in Eq. (3.8) becomes a zero vector because $\alpha(z_{i,j})$ and $\beta(z_{i,j})$ are almost zeros as $\mathbf{x}$ approaches the optimal point. If the number of degenerate voltage levels increases, the rank of $J_{G_{Ns}}(\mathbf{x})$ can be less than 8 and the proposed algorithms cannot converge. However, even for the 8 voltage-level case, the optimal parameters can be found if there exist no degenerate voltage levels.

Figure 3.3(a) shows the surface plot for $C_{12}(\mathbf{x})$, which measures the distribution in 12 levels, when the number of PE cycles and retention time are 5,000 times and 64 K hours, respectively. We obtained this plot by changing m_1 (x axis) and m_3 (y axis) while the other parameters were set to the optimal ones. Since the cost function is not a convex function, the proposed algorithms can settle down to a local minimum if the initial point $\mathbf{x}^{(\mathbf{0})}$ (red dot in Fig. 3.3(a)) is far away from the optimal one. In most cases, however, $\mathbf{x}^{(\mathbf{0})}$ is located near the optimal point because the parameter estimation algorithms can utilize the previously determined mean and standard deviation values. When the parameter estimation fails, we may retry it after changing the sensing reference voltages to eliminate degenerate voltage levels. One simple strategy for adjusting the MSRVs in the retrial process is to shift them into the negative direction with a fixed amount. For example, the right most degenerate voltage level of the 8-level case can be removed if we move all the reference voltages by -0.2 V. Then, the proposed algorithms can converge. Figure 3.3(b) shows the surface plot for $C_8(\mathbf{x})$ with one degenerate voltage level. Since at least eight non-degenerate voltage levels are needed to find the unique solution vector of eight parameters, the rank deficiency due to the degenerate voltage level causes infinitely many solutions for the equation



(a) $C_{12}(\mathbf{x})$



(b) $C_8(\mathbf{x})$

Figure 3.3: Error surface plot for (a) $C_{12}(\mathbf{x})$ and (b) $C_8(\mathbf{x})$.

of $\nabla C_8(\mathbf{x}) = 0$. As shown in Fig. 3.3(b), there exist many minimum points approximately along the straight line, and the proposed algorithms cannot find the optimum one.

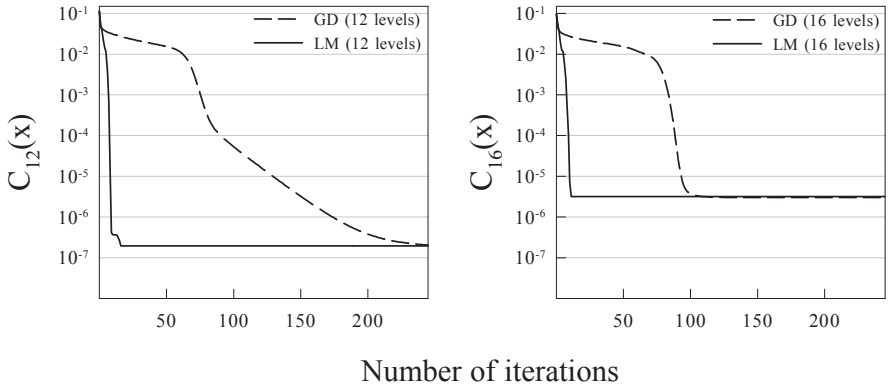


Figure 3.4: Learning curves of the proposed parameter estimation algorithms with *Ns* of 12 or 16.

In Fig. 3.4, $C_{12}(\mathbf{x})$ and $C_{16}(\mathbf{x})$ are plotted according to the iteration. Regardless of N_s, the LM based estimation algorithm converges much faster than the GD based one. However, the computational complexity for each iteration of the LM based method is a few times larger than that of the GD based one due to the matrix multiplication and inversion. In this figure, we can also find that the final estimation error value of the 12 voltage-level case is much smaller than that of the 16 voltage-level one. But this does not imply that the estimation with 12 voltage levels is better than that with 16 levels because $C_{12}(\mathbf{x})$ and $C_{16}(\mathbf{x})$ are basically different cost functions. For small N_s, the true and the estimated distributions are compared less strictly, while the comparison is conducted more strictly when N_s is large. When $C_{12}(\mathbf{x})$ and $C_{16}(\mathbf{x})$ are computed with the same distributions, the former is usually smaller than the latter.

We apply the proposed parameter estimation algorithms to hypothetical NAND flash memory whose retention time varies from 1 K to 256 K hours (29.2 years) while the number of PE cycles is fixed to 5,000 times. The amounts of mean and standard deviation shifts due to the data retention time are plotted in Fig. 3.5 and 3.6. When

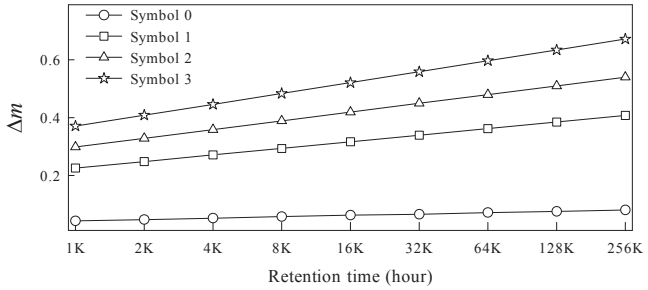


Figure 3.5: The amount of mean shift for each symbol when increasing the data retention time.

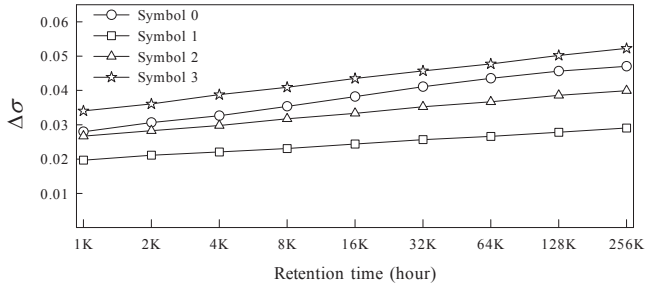


Figure 3.6: The amount of standard deviation change for each symbol when increasing the data retention time.

we observe the mean of the symbol 3 in Fig. 3.5, it is changed about 14 %, from 4.60 V to 3.94 V, by the retention of 256 K hours. The standard deviation of the symbol 3 shown in Fig. 3.6 grows about 41 %, from 0.128 V to 0.181 V, which can be considered as 3.0 dB increase in noise power. When the retention time is larger than 256 K hours, the parameter estimation algorithms fail to converge, thus we did not draw the figure beyond this point.

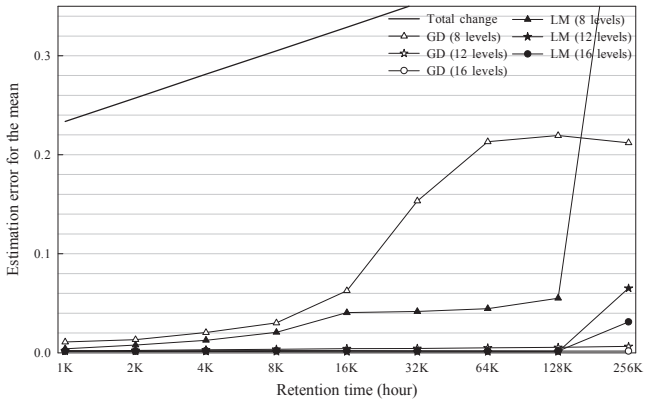


Figure 3.7: Estimation errors for the mean when increasing the data retention time.

In order to evaluate the accuracy of the estimated parameters, the estimation errors for the mean and the standard deviation are defined as follows:

$$\text{Estimation error for the mean} = \frac{1}{4}\sum_{j=0}^{3}\left|m_j - \hat{m}_j\right|, \tag{3.13}$$

$$\text{Estimation error for the standard deviation} = \frac{1}{4}\sum_{j=0}^{3}\left|\sigma_j - \hat{\sigma}_j\right|.$$

In Eq. (3.13), m_j and σ_j are the actual mean and standard deviation, while $\hat{m}_j$ and $\hat{\sigma}_j$ are the estimated ones. For comparison purpose, we also compute the estimation errors when the means and standard deviations of the initial voltage distribution are used for $\hat{m}_j$ and $\hat{\sigma}_j$, respectively.

Figure 3.7 shows the estimation error (V) for the mean when increasing the data retention time. When there is no parameter estimation (denoted as 'Total change' in Fig. 3.7), the estimation errors are in the range of 0.2 V to 0.5 V, and they grow almost linearly with the retention time. When the retention time is smaller than 256 K hours, the parameter estimation with 12 or 16 voltage levels results in the estimation error

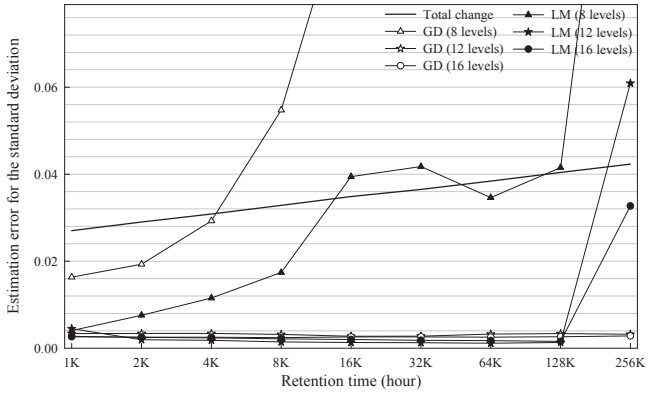


Figure 3.8: Estimation errors for the standard deviation when increasing the data retention time.

below 0.01 V, while the error is quite large for the 8 voltage-level case.

The estimation error (V) for the standard deviation is drawn in Fig. 3.8. The errors for the unestimated distributions are in the range of 0.02 V to 0.05 V and linearly increase with the retention time. When N_s is 8, the estimation errors for the proposed algorithms are even larger than those of the initial distribution, which means that the proposed algorithms converge to the non-optimum points. We find that at least 12 voltage sensing levels are needed for fairly correct estimation of the standard deviation.

Figure 3.9 illustrates the required number of iterations until both methods converge. Note that the maximum number of iterations (Max_iter in Algorithm 1 and 2) is set to 200. In most cases, the GD based method requires more iterations than the LM based one, and their difference becomes larger as the data retention time increases. According to Table 3.1, one iteration of the LM based method requires a few times more arithmetic operations than the GD based one. Thus, in most cases, the total amount of arithmetic operations for the LM based method is smaller than that

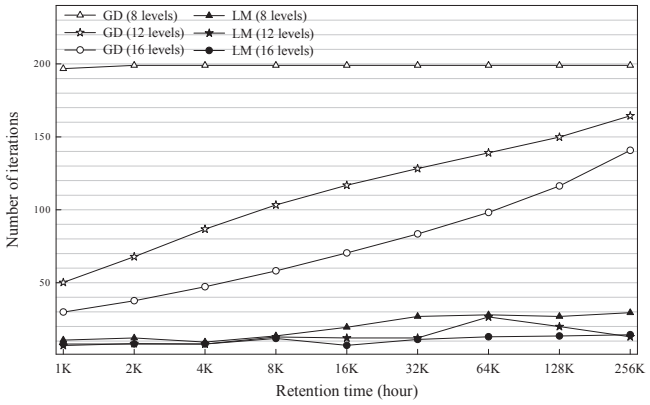


Figure 3.9: Number of iterations when increasing the data retention time.

of the GD based one.

3.2.4.2 Parameter Estimation with real NAND Flash Memory

We apply the proposed parameter estimation algorithms to the data samples that are obtained from the actual two-bit MLC NAND flash memory devices with a 20 nm-class process technology. The numbers of word- and bit-lines are the same as those of the simulated one; however, this memory employs the even/odd bit-line structure. In order to obtain the real mean and standard deviation data, the threshold voltages are measured with the precision of 0.1 V. Note that the memory device used in this work supports a manufacturer defined function that can alter the sensing reference voltages. The initial threshold voltage distribution is obtained from the fresh memory cells. After 1.0 K, 1.5 K, or 3 K times of PE cycling and 10 hours of baking process at 125 °C, the threshold voltage distributions of non-initial states were also obtained.

Table 3.2 shows the estimation errors. During the experiments, the number of voltage levels is fixed to 12. Note that 'm' and 'σ' on the 'Total change' columns denote the average mean and standard deviation shifts from the initial threshold voltage

Table 3.2: The estimation errors and the number of iterations (Iter.) of the SDE algorithm when applied to real NAND flash memory

		Even page			Odd page		
		Total change	GD	LM	Total change	GD	LM
1.0K	m	0.1091	0.0095	0.0094	0.1023	0.0180	0.0193
	σ	0.0125	0.0103	0.0101	0.0256	0.0160	0.0125
	Iter.	N/A	67.7	4.02	N/A	51.8	4.51
1.5K	m	0.1095	0.0112	0.0111	0.1013	0.0175	0.0186
	σ	0.0190	0.0102	0.0100	0.0338	0.0121	0.0113
	Iter.	N/A	80.6	4.02	N/A	65.7	4.47
3.0K	m	0.0889	0.0095	0.0095	0.0878	0.0165	0.0167
	σ	0.0327	0.0077	0.0076	0.0553	0.0129	0.0128
	Iter.	N/A	94.8	4.02	N/A	90.8	4.47

distribution after applying PE cycling and the data retention process. Similar to the simulated memory case, the proposed parameter estimation algorithms work well. We can find that the estimation errors for the mean and the standard deviation are less than 0.02 V in every case.

3.3 Decision Directed Estimation of Threshold Voltage Distribution

In this section, we propose a decision directed estimation algorithm that does not require extra memory sensing operations. We explain the basic idea of the proposed method, and then apply it to two-bit MLC NAND flash memory. We also propose a combined threshold voltage distribution estimation scheme that employs both the SDE and the DDE based approaches.

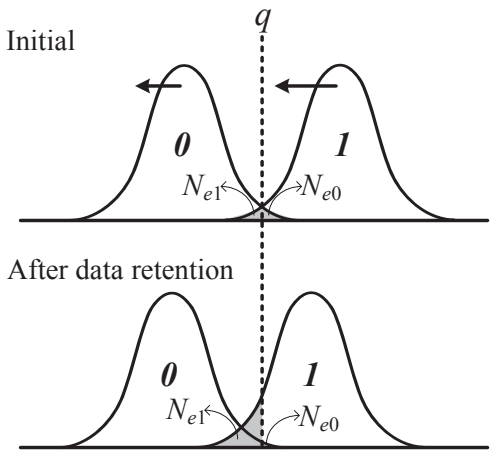


Figure 3.10: Threshold voltage distribution shift due to data retention in SLC NAND flash memory

3.3.1 Basic Idea

The decision directed estimation algorithm is inspired by the adaptive decision feedback equalizers that are widely used in communication systems [38, 39]. In the decision feedback equalizers, the reference data is provided by decoding the received data. The proposed DDE algorithm also updates the parameters of the model using the bit error patterns. Unlike the SDE algorithm that is triggered by unsuccessful error correction, the proposed method operates at successful page read operations and allows more frequent parameter updates. Besides the error corrected bit patterns, this method assumes the threshold voltage distribution as a Gaussian mixture and utilizes the linear relation between the mean and standard deviation shifts.

Consider a threshold voltage distribution of SLC (single-level cell) NAND flash memory shown in Fig. 3.10. Assume that the initial threshold voltage distribution is shifted to the negative direction because of the data retention problem, which causes

unbalanced bit errors if the quantization boundary remains the same. Let N_{e1} be the number of cells that were initially programmed as the symbol *1* but read as the symbol *0*. N_{e0} is defined as the number of bit errors converted from *0* to *1*. If error correction is successful, we can obtain N_{e1} and N_{e0} by comparing the input and the output of the error correction circuit. As data retention time elapses, N_{e1} increases gradually but N_{e0} may decrease and become 0. As a result, N_{e1} provides more reliable information than N_{e0}. We estimate N_{e1} by using the partial CDF (cumulative density function) of the modeled Gaussian distribution as follows:

$$N_{e1} = N_{w1} \cdot \{1 - Q(z_1)\}, \tag{3.14}$$

where

$$z_1 = \frac{q - m_1}{\sigma_1}. \tag{3.15}$$

Note that N_{w1} is the number of cells that were written as the symbol *1* during the programming operation. N_{w1} can be counted from the error corrected bit pattern. In Eq. (3.15), m_1 and σ_1 represent the mean and the standard deviation of the threshold voltage distribution for the input symbol *1*, and they can be represented as

$$m_1 = m_{i1} + \Delta m_1 \tag{3.16}$$

and

$$\sigma_1 = \sigma_{i1} + \Delta\sigma_1, \tag{3.17}$$

where m_{i1} and σ_{i1} denote the initial mean and standard deviation values, respectively. Since the inverse Q-function, $Q^{-1}(x)$, is well defined in the region of $[0, 1]$, z_1 in Eq.

(3.14) and (3.15) can be obtained as follows:

$$z_1 = Q^{-1}\left(1 - \frac{N_{e1}}{N_{w1}}\right) = \frac{q - (m_{i1} + \Delta m_1)}{\sigma_{i1} + \Delta\sigma_1}. \tag{3.18}$$

Since there are two unknown variables, it is not possible to uniquely determine the values of Δm_1 and $\Delta\sigma_1$ by using Eq. (3.18) alone. Recall that the amount of standard deviation shift due to the data retention process is linearly proportional to that of the mean shift. By using Eq. (2.5), we have

$$Q^{-1}\left(1 - \frac{N_{e1}}{N_{w1}}\right) = \frac{q - (m_{i1} + \Delta m_1)}{\sigma_{i1} + \beta_1 \Delta m_1}. \tag{3.19}$$

Thus, the mean and standard deviation shifts become

$$\Delta m_1 = \frac{q - m_{i1} - \sigma_{i1} Q^{-1}\left(1 - \frac{N_{e1}}{N_{w1}}\right)}{1 + \beta_1 Q^{-1}\left(1 - \frac{N_{e1}}{N_{w1}}\right)} \tag{3.20}$$

and

$$\Delta\sigma_1 = \beta_1 \Delta m_1, \tag{3.21}$$

respectively. Note that Eq. (3.20) is not singular because both β_1 and $Q^{-1}\left(1 - N_{e1}/N_{w1}\right)$ in the denominator are negative. The same approach can be applied to obtain Δm_0 and $\Delta\sigma_0$, thus we have

$$\Delta m_0 = \frac{q - m_{i0} - \sigma_{i0} Q^{-1}\left(\frac{N_{e0}}{N_{w0}}\right)}{1 + \beta_0 Q^{-1}\left(\frac{N_{e0}}{N_{w0}}\right)} \tag{3.22}$$

and

$$\Delta\sigma_0 = \beta_0 \Delta m_0. \tag{3.23}$$

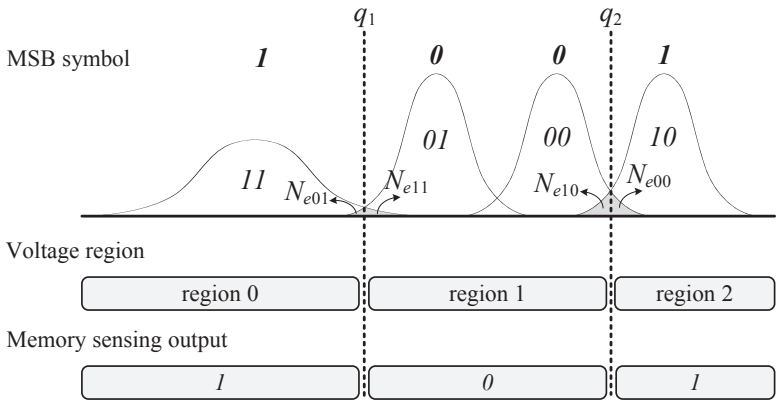


Figure 3.11: MSB page read operation in two-bit MLC NAND flash memory.

Unlike Eq. (3.20), computing Eq. (3.22) can cause large estimation errors because the inverse Q-function, $Q^{-1}(x)$, is sensitive to small perturbations of x when x is near zero. Note that N_{e0} is close to zero after the data retention process. Also, more importantly, Eq. (3.22) can be singular because the $\beta_0 Q^{-1}(N_{e0}/N_{w0})$ term in the denominator is negative. In order to obtain Δm_0 and $\Delta\sigma_0$ with small estimation errors, we indirectly find them by using Δm_1 and $\Delta\sigma_1$, which will be explained in the following subsection.

3.3.2 Applying to Two-Bit MLC NAND Flash Memory

The proposed method can be directly extended to two-bit MLC NAND flash memory. Let us assume that an MSB (most significant bit) page read operation is conducted as shown in Fig. 3.11. For reading an MSB page, two memory sensing operations are required, where the sensing with q_1 discriminates the symbol *11* and *01*, while the other with q_2 is for the symbol *00* and *10*.

By comparing the results of memory sensing and ECC decoding, we can count

the number of bit errors and also identify the type of them. Let us assume that the MSB of a cell was programmed as the symbol *0* and the cell is sensed at the voltage region 0, where the memory sensing output bit is the symbol *1*. Then, this cell causes the $01 \rightarrow 11$ type of bit error and increases N_{e01} by one. In this manner, we can obtain N_{e01} and N_{e00} as follows:

N_{e01} **:** Sensed at region 0 but decoded as 0 by error correction

N_{e00} **:** Sensed at region 2 but decoded as 0 by error correction

We can compute N_{e11} and N_{e10} by using N_{e01} and N_{e00}. Note that both N_{e11} and N_{e10} are in the voltage region 1. Let us denote the number of cells in the voltage region 0 as $N_{region0}$. Then, the following equation holds:

$$N_{region0} = N_{w11} - N_{e11} + N_{e01}. \tag{3.24}$$

By solving Eq. (3.24), we can obtain N_{e11}. Similarly, N_{e10} also can be computed by the following equation:

$$N_{region2} = N_{w10} - N_{e10} + N_{e00}. \tag{3.25}$$

Since the symbol *01* and *10* correspond to the symbol *1* in (3.10), the mean and standard deviation shifts for these symbols can be computed by using Eq. (3.20) and (3.21). However, finding Δm_{00} and $\Delta\sigma_{00}$ using Eq. (3.22) and (3.23) is subject to large estimation errors because N_{e00} is quite close to zero after the data retention process. Instead, we can compute the mean and standard deviation shifts of threshold voltage for the symbol *00* by considering that the amount of mean shift is proportional to the

initial threshold voltage, thus we have

$$\Delta m_{00} = \frac{1}{2}\left(\Delta m_{01} + \Delta m_{10}\right) \tag{3.26}$$

and

$$\Delta \sigma_{00} = \beta_{00} \Delta m_{00}. \tag{3.27}$$

Since the threshold voltage distribution of the erased state (symbol *11*) only changes slightly with the data retention process [23, 24], we assume that Δm_{11} and $\Delta \sigma_{11}$ are zeroes.

To lower the estimation error for the proposed method, we need to find the accurate value of β, which is the coefficient between the mean and standard deviation shifts. According to [24], β is a function of the input symbol and the number of PE cycles. Thus, it is needed to find the value of β while changing the number of PE cycles (e.g. 0, 0.5K, 1.0K, 1.5K, and so on) for each symbol. The values of β according to the PE cycles can be stored at a look-up table, and linear interpolation determines the intermediate values of look-up table entries. Figure 3.12 shows an example of finding β for each symbol. In this work, the least squares method is used for fitting straight lines to the observed data samples.

The proposed DDE algorithm can also be extended to TLC (triple-level cell) NAND flash memory. The read operation of NAND flash memory involves address decoding, cell array accessing, and data output. According to the timing parameters in [40], the SDE with 12-level voltage sensing requires approximately 300 μs per page only for the memory sensing operations. In order to apply the SDE algorithm to TLC NAND flash memory, in which 16 parameters of 8 symbols need to be found, more

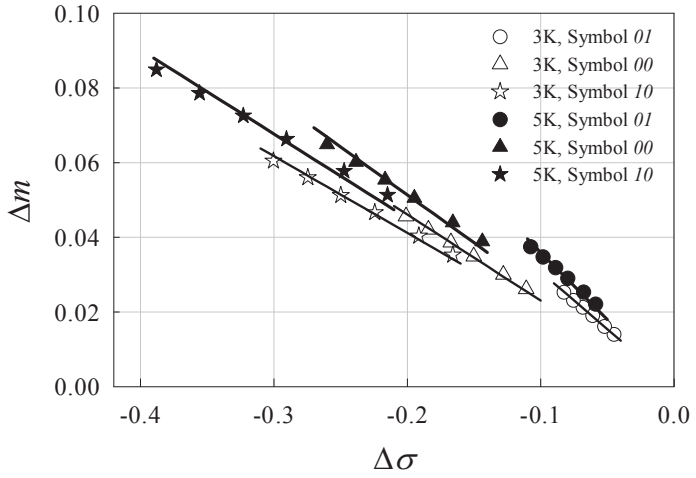


Figure 3.12: $\Delta m - \Delta\sigma$ plots when the numbers of PE cycles are 3 K and 5 K times.

than 16-level voltage sensing is required. Note that 16-level voltage sensing demands approximately 325 μs. Moreover, the computational complexity of the SDE algorithm is proportional to the square of the number of parameters. On the other hand, the DDE algorithm demands no memory sensing overheads because this method only uses the output of the MSB page read request. Also, the computational complexity of the DDE algorithm is linearly proportional to the number of parameters. As a result, using the DDE algorithm becomes more beneficial over the SDE one as the TLC technology replaces the MLC one.

3.3.3 Combined Threshold Voltage Distribution Estimation

The proposed DDE algorithm needs a condition of successful error correction to update the model parameters gradually. The DDE algorithm is effective when the parameter update is conducted frequently. If the data retention time is too long to decode all the data without errors, we have to employ the SDE algorithm. By combining both

the SDE and the DDE algorithms, we can reduce the overhead of extra memory sensing operations while maintaining the estimation accuracy. If the amount of threshold voltage shift is very small, the parameter update needs not to be conducted. If the data retention induced distortion is a little bit severe but error correction is still successful, the DDE algorithm is conducted to update the parameters. In order to measure the amount of threshold voltage shifts, N_{ed}, which is defined as $N_{e1} - N_{e0}$, is used. When N_{ed} is larger than a threshold value N_{α}, which means a significant amount of shift, the DDE algorithm is invoked. The SDE algorithm is conducted when the decoding failure occurs.

3.3.4 Error Analysis

In this subsection, we assess the estimation accuracy by analyzing the sources of errors in the DDE algorithm. If the variance of the threshold voltage distribution is small, which is quite common for the NAND flash memory cells with small PE cycles, the number of unbalanced bit errors is close to zero as shown in Fig. 3.13, thus it is quite difficult to find the model parameters by only using the DDE algorithm. However, in this case, finding the optimal MSRVs is not critical to lower the raw bit error rate (RBER) because only a small number of unbalanced bit errors is generated even after the data retention process. Therefore, we confine the error analysis to the cases when the number of PE cycles is not small so that the RBER is fairly affected by the estimation accuracy.

3.3.4.1 Modeling Error in the Threshold Voltage Distribution

Even though a Gaussian mixture can model the actual threshold voltage distribution quite closely [41, 13], the two distributions do not match perfectly especially at the

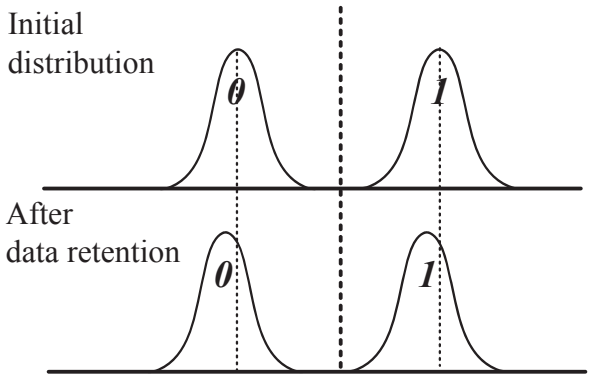


Figure 3.13: Threshold voltage distribution shift due to data retention for the memory cells with small PE cycles.

symbol boundaries. Figure 3.14 shows a threshold voltage distribution of 20 nm-class NAND flash memory and its Gaussian approximation. We can notice that the two distributions do not match well at the symbol boundaries, which results in inaccurate numbers of bit errors N_e. In fact, the Gaussian mixture model in Fig. 3.14 underestimates the number of bit errors for MSB pages, and N_{e1} in Eq. (3.20) is smaller than the true value.

Let us assume that the Gaussian mixture model causes a small error denoted as ΔN_e. If we define m_e as the estimation error caused by ΔN_e, Eq. (3.20) becomes

$$\begin{aligned}\Delta m + m_e &= \frac{q - m_i - \sigma_i Q^{-1}\left(1 - \frac{N_e + \Delta N_e}{N_w}\right)}{1 + \beta Q^{-1}\left(1 - \frac{N_e + \Delta N_e}{N_w}\right)} \\ &= \frac{q - m_i - \sigma_i Q^{-1}(1 - \gamma - \Delta\gamma)}{1 + \beta Q^{-1}(1 - \gamma - \Delta\gamma)},\end{aligned} \tag{3.28}$$

where

$$\gamma = \frac{N_e}{N_w} \text{ and } \Delta\gamma = \frac{\Delta N_e}{N_w}. \tag{3.29}$$

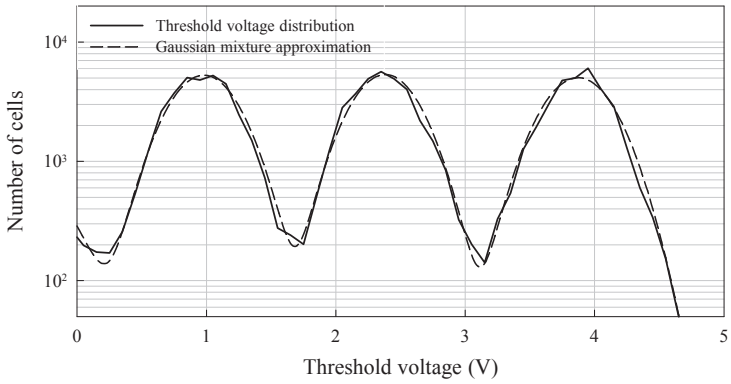


Figure 3.14: Threshold voltage distribution of an actual NAND flash memory device and its Gaussian approximation.

Recall that the DDE algorithm is conducted only when the decoding process is succeeded. Thus, ΔN_e is smaller than the error correcting capability. Moreover, N_w is much larger (around 10 K) than ΔN_e, thus $\Delta\gamma$ is a small value. We can approximate $Q^{-1}(1-\gamma-\Delta\gamma)$ by employing the first order Taylor expansion as follows:

$$\begin{aligned} Q^{-1}(1-\gamma-\Delta\gamma) &\simeq Q^{-1}(1-\gamma) - \left.\frac{dQ^{-1}(y)}{dy}\right|_{y=1-\gamma} \cdot \Delta\gamma \\ &= z - \frac{\Delta\gamma}{\left.\frac{dQ(x)}{dx}\right|_{x=z}} \\ &= z + \sqrt{2\pi} e^{\frac{z^2}{2}} \cdot \Delta\gamma \end{aligned} \tag{3.30}$$

where

$$z = Q^{-1}(1-\gamma). \tag{3.31}$$

Note that z represents the z-score that expresses the divergence of the observed data from its mean value. By using the approximation of $Q^{-1}(1-\gamma-\Delta\gamma)$, Eq. (3.28) can

be simplified as

$$\begin{aligned}
\Delta m + m_e &= \frac{q - m_i - \sigma_i z - \sqrt{2\pi}\sigma_i e^{\frac{z^2}{2}}\Delta\gamma}{1+\beta z + \sqrt{2\pi}\beta e^{\frac{z^2}{2}}\Delta\gamma} \\
&= \left(\frac{q - m_i - \sigma_i z}{1+\beta z}\right)\left(\frac{1+\beta z}{1+\beta z + \sqrt{2\pi}\beta e^{\frac{z^2}{2}}\Delta\gamma}\right) - \frac{\sqrt{2\pi}\sigma_i e^{\frac{z^2}{2}}\Delta\gamma}{1+\beta z + \sqrt{2\pi}\beta e^{\frac{z^2}{2}}\Delta\gamma} \\
&= \Delta m \cdot \left(1 - \frac{\sqrt{2\pi}\beta e^{\frac{z^2}{2}}\Delta\gamma}{1+\beta z + \sqrt{2\pi}\beta e^{\frac{z^2}{2}}\Delta\gamma}\right) - \frac{\sqrt{2\pi}\sigma_i e^{\frac{z^2}{2}}\Delta\gamma}{1+\beta z + \sqrt{2\pi}\beta e^{\frac{z^2}{2}}\Delta\gamma} \\
&= \Delta m - \frac{\sqrt{2\pi}e^{\frac{z^2}{2}}\Delta\gamma(\beta\Delta m + \sigma_i)}{1+\beta z + \sqrt{2\pi}\beta e^{\frac{z^2}{2}}\Delta\gamma}.
\end{aligned} \tag{3.32}$$

Thus, the estimation error for the mean caused by ΔN_e becomes

$$m_e = -\left\{\frac{\beta\Delta m + \sigma_i}{\frac{1+\beta z}{\sqrt{2\pi}}\exp\left(\frac{-z^2}{2}\right) + \beta\Delta\gamma}\right\}\Delta\gamma. \tag{3.33}$$

In Eq. (3.33), the denominator is a function of z. When the magnitude of z is large, the second term, $\beta\Delta\gamma$, is dominant because of the exponential function in the denominator. On the other hand, when the magnitude of z is small, *e.g.* below 4, the magnitude of the first term is much larger than the second one because $\Delta\gamma$ is a small value. As shown in Fig. 3.10, for the symbol *1*, the magnitude of z decreases as the amount of shift increases. Thus, when we apply the combined estimation scheme, in which the DDE algorithm is conducted when the amount of shift is larger than the pre-defined threshold, $\beta\Delta\gamma$ term in the denominator of Eq. (3.33) becomes negligible. As a result, Eq. (3.33) can be further simplified as

$$m_e \simeq C(z)\Delta\gamma, \tag{3.34}$$

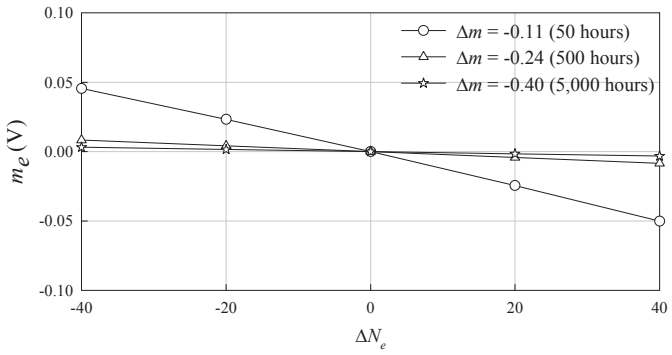


Figure 3.15: The estimation errors for the mean m_e computed while changing ΔN_e from -40 to 40.

where

$$C(z) = -\sqrt{2\pi}\left(\frac{\beta\Delta m + \sigma_i}{1+\beta z}\right)\exp(\frac{z^2}{2}). \tag{3.35}$$

According to Eq. (3.34), the estimation error for the mean is linearly proportional to $\Delta\gamma$ with a slope of $C(z)$. Thus, the estimation errors grows as the magnitude of z increases. Since the DDE algorithm is conducted only when the magnitude of z is small, estimation errors are not large.

Figure 3.15 shows the estimation errors for the mean when ΔN_e changes. By observing the threshold voltage distributions of 20 nm-class NAND flash memory, we find that the magnitude of ΔN_e is no more than 20 in most cases. Considering this observation, we change ΔN_e from -40 to 40. Note that the estimation errors are obtained by computing Eq. (3.33). We consider three cases depending on the amount of threshold voltage shifts Δm. In this figure, we can find that the estimation error decreases as the magnitude of Δm increases. The estimation errors for $\Delta m = -0.40$ and -0.24 cases are below 0.01 V even when ΔN_e is 40. On the other hand, the estimation error for $\Delta m = -0.11$ case is over 0.02 V even when ΔN_e is equal to 20.

Thus, in the combined estimation scheme, N_α needs to be carefully selected so that the DDE algorithm is invoked when the amount of mean shift is larger than 0.11 V.

3.3.4.2 Sensitivity of the Accuracy on β

In the DDE algorithm, we assume that the standard deviation shift is linearly proportional to the amount of mean shift with the coefficient of β. Since the values of β vary with the number of PE cycles and the initial threshold voltages, we need to find β through off-line training. However, the actual value of β in each device may vary, thus we need to quantify the effects on the estimation accuracy when the value of β is not known exactly.

Let $\Delta\beta$ be the error in estimating β, then Eq. (3.20) becomes

$$\Delta m + m_e = \frac{q - m_i - \sigma_i Q^{-1}(1-\gamma)}{1 + (\beta + \Delta\beta) Q^{-1}(1-\gamma)}. \tag{3.36}$$

Note that m_e is the estimation error for the mean caused by the $\Delta\beta$. Equation (3.36) can be simplified as

$$\begin{aligned} \Delta m + m_e &= \left(\frac{q - m_i - \sigma_i z}{1+\beta z}\right) \cdot \left(\frac{1+\beta z}{1+\beta z + \Delta\beta z}\right) \\ &= \Delta m \cdot \left(1 - \frac{\Delta\beta z}{1+\beta z + \Delta\beta z}\right). \end{aligned} \tag{3.37}$$

Thus, we have

$$m_e = -\left(\frac{\Delta m}{1/z + \beta + \Delta\beta}\right) \Delta\beta. \tag{3.38}$$

The error curves for the symbol *01* and *10* are plotted in Fig. 3.16 by computing Eq. (3.38). Note that $\Delta\beta$ changes from -50 % to 50 % of β. When the error in β is below 30 % of the true value, it is expected that the estimation errors are smaller than

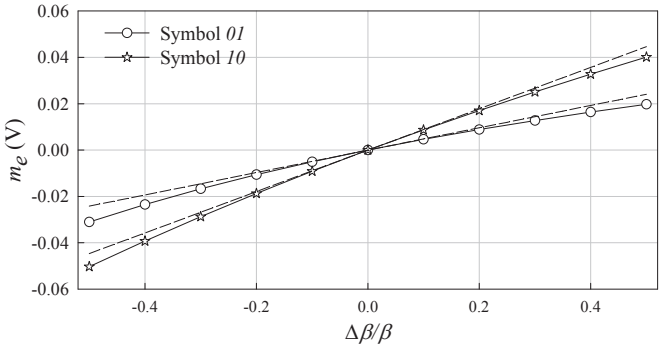


Figure 3.16: The estimation errors for the mean m_e computed while changing $\Delta\beta$ from -50 % to 50 % of β.

0.03 V. When comparing the estimation errors for both symbols, the symbol *01* shows smaller errors than the symbol *10*. This is because the estimation error is proportional to the amount of the mean shift Δm.

3.3.5 Experimental Results

3.3.5.1 Parameter Estimation with Simulated NAND Flash Memory

We apply the DDE algorithm while changing the retention time from 10 to 10 K hours. Note that the number of PE cycles is fixed to 5,000 times. To evaluate the accuracy of the DDE algorithm, we measure the magnitudes of the estimation errors for the means and the standard deviations. Since the threshold voltage distribution of the symbol *11* changes very slightly after the data retention process, we do not measure the estimation errors for this symbol.

The estimation errors for the mean are shown in Fig. 3.17. When no threshold voltage estimation is applied, the mean shift from the initial value is denoted as 'Total change'. As the data retention time elapses, the amount of mean shift from the initial

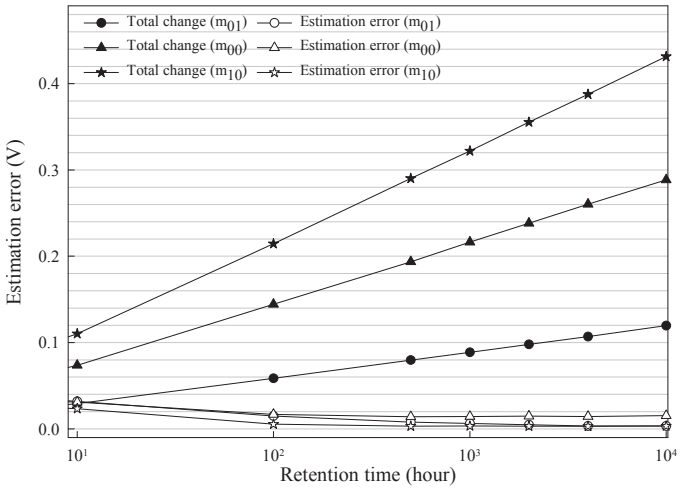


Figure 3.17: Estimation errors for the mean when PE cycles is 5,000, and the data retention time increases from 10 to 10K hours.

value also increases. For example, when the data retention time is 100 hours, the amounts of mean shifts become 0.06 V, 0.14 V, and 0.22 V for the symbol *01*, *00*, and *10*, respectively. After 10 K hours of the data retention time, the amounts of shifts increase up to 0.12 V, 0.29 V, and 0.43 V, respectively. By applying the DDE method, the estimation errors are reduced much and remain below 0.04 V for all the cases. In Fig. 3.17, it is also shown that the error levels decrease as the data retention time increases. As analyzed at the previous section, the DDE shows a relatively small estimation error when the amount of shift is large. For the same reason, the estimation error for symbol *10* is the smallest.

The estimation errors for the standard deviations are plotted in Fig. 3.18. The DDE algorithm can find the standard deviation values quite accurately with the estimation errors below 0.03 V. When the data retention time is around 10 hours, the estimation errors are larger than 'Total change', which means that the estimation process is not helpful to reduce the RBER. However, the estimation error decreases as

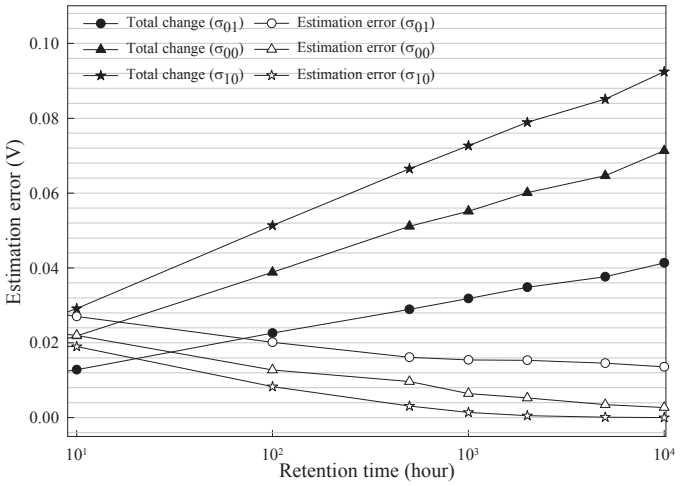


Figure 3.18: Estimation errors for the standard deviation when PE cycles is 5,000, and the data retention time increases from 10 to 10 K hours.

the data retention time elapses, and the DDE algorithm can find accurate standard deviation values after 100 hours of the retention time. Thus, in the combined estimation method, N_α needs to be properly selected so that the DDE algorithm is conducted after at least 100 hours of data retention time. Recall that the DDE algorithm is invoked only when $N_{e1} - N_{e0}$ is larger than N_α in the combined estimation scheme. In order to select the proper value of N_α, an off-line training is needed. For the given data retention time, *e.g.* 100 hours in this example, we count N_{e1} and N_{e0} and set N_α as $N_{e1} - N_{e0}$.

Figure 3.19 shows the RBER of simulated NAND flash memory with the PE cycles of 5 K times. The horizontal dashed and solid lines indicate the RBERs of 0.005 and 0.0062, respectively. If we employ a (72306, 65536, 400) BCH code, whose frame error rate (FER) is 0.015 and 1.0 at the RBERs of 0.005 and 0.0062, respectively, the DDE algorithm can be used until 2,000 hours of the retention time. When the data retention time is around 1,800 hours, the probability that the DDE algorithm

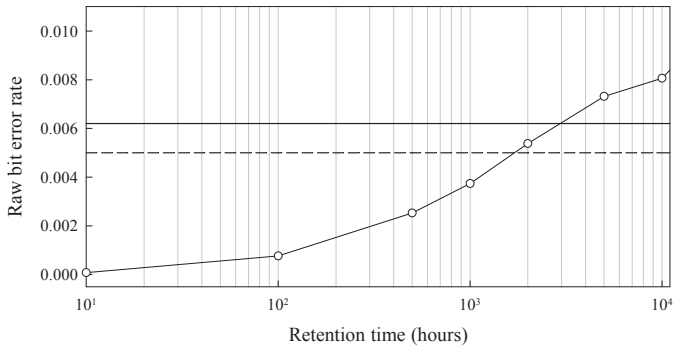


Figure 3.19: Raw bit error rate when PE cycles is 5,000 and the data retention time increases from 10 to 10 K hours.

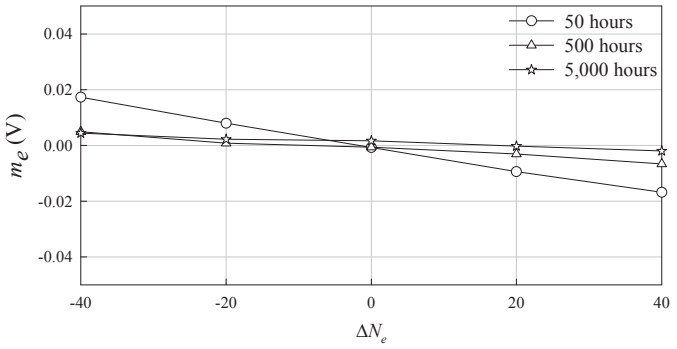


Figure 3.20: Estimation errors for the mean while changing ΔN_e from -40 to 40.

updates the model parameters is about 98.5 %. However, when the data retention time is over 4,000 hours, error correction always fails, thus the SDE needs to be conducted.

To evaluate the additional estimation errors caused by imprecise modeling of the threshold voltage distribution, we apply the DDE algorithm while changing ΔN_e from -40 to 40. In the experiments, the number of PE cycles is fixed to 5,000 times, while the data retention time changes from 50 to 5 K hours. Figure 3.15 shows the estimation errors for the mean of the symbol *10*. The experimental results are quite similar to Fig. 3.15. When the data retention time is larger than 50 hours, the estimation er-

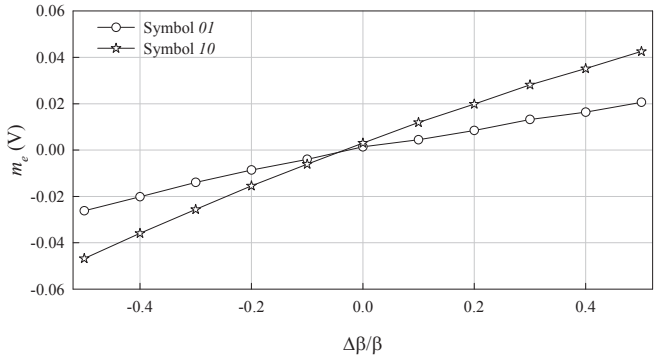


Figure 3.21: Estimation errors for the mean while changing $\Delta\beta$ from -50 % and 50 % of β.

rors are quite small (below 0.02 V), which demonstrates that the DDE algorithm is robust in these cases.

The estimation errors caused by inaccurate β are also evaluated while changing $\Delta\beta$ from -50 % to 50 % of β. In here, the number of PE cycles and the data retention time are set to 5,000 times and 5,000 hours, respectively. Figure 3.21 shows the estimation error curves for the mean. As expected in Fig. 3.16, the estimation errors are almost linearly proportional to $|\Delta\beta/\beta|$. Even when β is 50 % overestimated, the DDE algorithm can find the mean values with only 0.04 V of the estimation error.

3.3.5.2 Parameter Estimation with Real NAND Flash Memory

We apply the proposed parameter estimation algorithms to the data samples that are obtained from two-bit MLC NAND flash memory devices fabricated with a 20 nm-class process technology. The initial threshold voltage distribution is obtained from the fresh memory cells. The threshold voltage distributions of non-initial states were obtained after 1.0 K, 1.5 K, or 3 K times of PE cycling and 10 hours of baking process at 125 °C, which is equivalent to one year data retention. The RBER of the

Table 3.3: The estimation errors of the DDE algorithm when applied to real NAND flash memory

		Total change			DDE		
		01	*00*	*10*	*01*	*00*	*10*
m	1.0 K	0.0397	0.0987	0.1787	0.0286	0.0143	0.0121
	1.5 K	0.0390	0.0984	0.1841	0.0238	0.0120	0.0122
	3.0 K	0.0253	0.0836	0.1842	0.0152	0.0164	0.0146
σ	1.0 K	0.0195	0.0203	0.0174	0.0122	0.0043	0.0062
	1.5 K	0.0289	0.0281	0.0235	0.0153	0.0050	0.0064
	3.0 K	0.0572	0.0476	0.0400	0.0286	0.0080	0.0077

1.0 K, 1.5 K, and 3 K baked cells are 0.0022, 0.0031, and 0.0053, respectively. If we employ the (72306, 65536, 400) BCH code, the DDE algorithm can be applied with the probability of about 90 % when reading 3 K baked cells. The need of conducting SDE algorithm that demands extra memory sensing is quite small, about 10 %, even for the memory with 3 K PE cycles.

Table 3.3 shows the estimation errors for each symbol. In Table 3.3, 'm' and 'σ' on the 'Total change' columns denote the mean and standard deviation shifts from the initial threshold voltage distribution after applying PE cycling and the data retention process. The DDE algorithm results in relatively large estimation errors for the symbol *01* especially when the number of PE cycles is 1.0 K. In this case, the variance of the threshold voltage distribution is small, and the amount of shift is below 0.04 V. Thus, only a small number of unbalanced bit errors occurs, which degrades the estimation accuracy of the DDE algorithm much. However, the RBER for the 1.0 K baked memory cells is only 0.0022. If we employ the (72306, 65536, 400) BCH code, the corrected BER is much lower than 10^{-15}, which is considered as 'error-free' in NAND flash memory. Thus, for the fresh or 1.0 K baked cells, the

effect of inaccurate parameter estimation is not critical. Overall, the proposed DDE algorithm works well on real NAND flash memory, and the estimation errors for the mean and the standard deviation are less than 0.03 V.

3.4 Concluding Remarks

We have developed parameter estimation algorithms for modeling the threshold voltage distribution of MLC NAND flash memory. The SDE algorithm that employs the gradient descent or Levenberg-Marquardt based optimization requires about 11 memory sensing operations to estimate all the mean and variance values of four symbols in two-bit MLC NAND flash memory. The DDE algorithm utilizes error corrected bit patterns and requires no additional memory sensing operations. We also have developed a combined estimation scheme that employs both the DDE and the SDE algorithms depending on the amount of threshold voltage shift. The effectiveness of the proposed algorithms is evaluated by using both simulated and real NAND flash memory. The SDE algorithm shows small estimation errors even when the threshold voltage distribution varies significantly, but demands several memory sensing overheads. On the other hand, the DDE algorithm requires no additional memory sensing operations and can be conducted only when the amount of shift is small so that error corrected bit patterns are available.

Chapter 4

Cell-to-Cell Interference Cancellation

4.1 Introduction

The capacitance coupling effect or cell-to-cell interference (CCI) becomes the dominant source of bit errors for sub-20 nm NAND flash memory. According to [3], the amount of CCI exceeds 50 % of the total noise when the feature size of the semiconductor process is 20 nm. Although some cell structures have been devised to reduce the CCI, removing the CCI is still very critical for sub-20 nm NAND flash memory devices.

There have been several works to develop signal processing solutions for CCI cancellation [36, 42, 43]. In [36], data post-compensation and pre-distortion techniques were proposed. Also, in [42], an adaptive LMS (least mean square) filter based coupling canceller was studied. Even though these techniques offer promising solutions for CCI cancellation, they require high resolution input data [36, 42], which leads to longer latency and more energy for data acquisition and transfer. More im-

portantly, detailed characterization of capacitance coupling, which is necessary for designing effective CCI cancellation algorithms, has not been provided in those previous works.

In this chapter, we present the experimental characterization of the cell-to-cell interference in NAND flash memory. To this end, we measure the coupling coefficients of an actual NAND flash memory chip with a 26 nm process technology by using a simple NAND flash memory controller implemented on an FPGA board [44]. We also develop a CCI cancellation algorithm that consists of the coupling coefficient estimation and the CCI removal steps. In the proposed CCI cancellation algorithm, the coupling coefficients can be found by using either specific programming patterns or ordinary data. In order to reduce the number of memory sensing operations for CCI cancellation, we study the optimal quantization schemes. The proposed CCI cancellation algorithm is evaluated by using the data samples obtained from both simulated and actual NAND flash memory, and the BER (bit error rate) performance is presented.

This chapter is organized as follows. Section 4.2 addresses the statistical characteristics of the CCI and proposes a direct approach to measure the coupling coefficients. In Section 4.3, we develop a least squares method based coupling coefficient estimation algorithm. The optimal multi-level memory sensing schemes for the proposed CCI cancellation algorithm are studied in Section 4.4. The experimental results are shown in Section 4.5. Finally, concluding remarks are made in Section 4.6.

4.2 Direct Measurement of Coupling Coefficients

The cell-to-cell interference cancellation algorithm removes the correlation among the cells by employing signal processing algorithms that are similar to interference cancellers employed in conventional communication systems. In this dissertation, we propose a CCI cancellation algorithm that estimates the coupling coefficients by using either specific programming patterns or ordinary data and removes the CCI with simple arithmetics. In the proposed CCI cancellation algorithm, the amount of CCI that the (m,n)-th victim cell receives $V_{CCI}[m,n]$ is modeled as

$$\begin{aligned} V_{CCI}[m,n] = {} & C_x \cdot (\Delta V[m,n-1] + \Delta V[m,n+1]) + C_y \cdot \Delta V[m+1,n] \\ & + C_{xy} \cdot (\Delta V[m+1,n-1] + \Delta V[m+1,n+1]), \end{aligned} \tag{4.1}$$

where $\Delta V[m,n-1]$, $\Delta V[m,n+1]$, $\Delta V[m+1,n]$, $\Delta V[m+1,n-1]$, and $\Delta V[m+1,n+1]$ are the threshold voltage shifts of the left, right, upper, upper-left, and upper-right neighbor cells, respectively. If the multi-page programming scheme shown in Fig. 4.1 is employed, $\Delta V[i,j]$ is equal to $V_{TH}[i,j] - V_L[i,j]$, where $V_L[i,j]$ and $V_{TH}[i,j]$ are the threshold voltages of the (i,j)-th cell after the LSB and the MSB programming, respectively. Note that Eq. (4.1) is for the even victim cells.

In Eq. (4.1), C_x, C_y, and C_{xy} are the coupling coefficients for the x, y, and xy directions, respectively, which need to be found during the coefficient estimation step of the proposed CCI cancellation algorithm. Once the amount of CCI is estimated by using Eq. (4.1), it needs to be subtracted from the memory sensing output signal V_Q during the CCI removal step. As a result, the output of the proposed CCI canceller

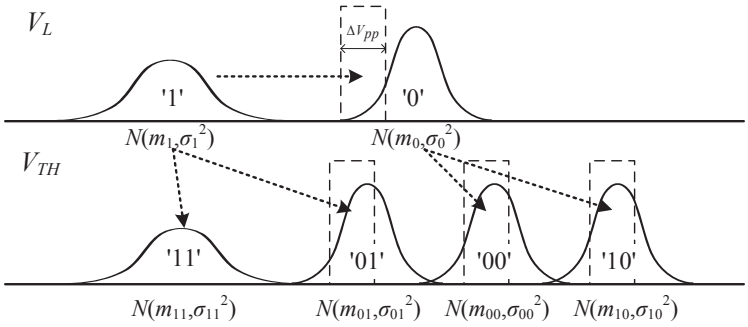


Figure 4.1: Multi-page programming scheme.

becomes

$$V_O[m,n] = V_Q[m,n] - V_{CCI}[m,n]. \tag{4.2}$$

In order to achieve a satisfactory BER performance, the CCI cancellation algorithm needs to estimate the coupling coefficients precisely. In this section, we propose a direct method that measures the coupling coefficients of actual NAND flash memory devices.

4.2.1 Measurement Procedure

In the direct approach, the coupling coefficient of each direction is measured by using one of the programming patterns shown in Fig. 4.2. In these programming patterns, only one of the interfering cells is programmed, and the threshold voltage shift of the victim cell is measured to obtain the coupling coefficient. Note that the other neighboring cells are in the erased state. In Fig. 4.2, the left, middle, and right programming patterns are designed to measure the coupling coefficients of the y, x, and xy directions or C_y, C_x, and C_{xy}, respectively. When using the programming patterns of Fig. 4.2-(a), only the upper neighboring cell of a victim cell is programmed. Thus,

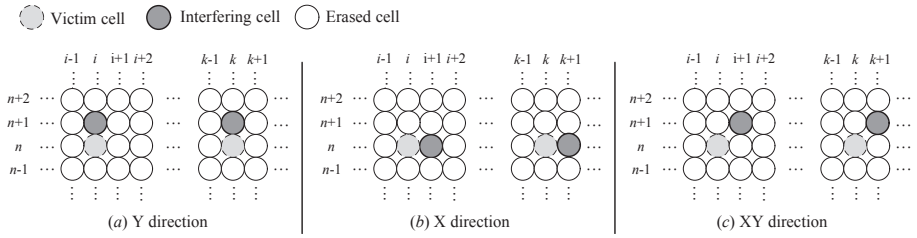


Figure 4.2: Programming patterns to measure the coupling coefficients.

$\Delta V[m+1,n]$ has a non-zero value in Eq. (4.1), from which we can directly compute C_y. Similarly, the programming patterns shown in Fig. 4.2-(b) and 4.2-(c) can be used to measure C_x, and C_{xy}, respectively. Note that we do not plot the right and right-upper interfering cells in Fig. 4.2 for simplicity. In these programming patterns, the victim cells are located sparsely so that they do not interfere with each other. Actually, the (n,i) and the (n,k)-th victim cells in Fig. 4.2 are 8 cell-distance away from each other. The other cells besides victim and interfering ones are not programmed, thus they do not affect the threshold voltages of the victim and the interfering cells.

In the proposed direct approach, it is important for obtaining the coupling coefficients to measure the threshold voltages in high precision. In this research, we use a simple NAND flash memory controller implemented on a Vertex 6 FPGA board [44] to measure the coupling coefficients from actual NAND flash memory chips. This NAND flash memory controller offers basic functionalities, such as erase, program, and read operations. To measure the threshold voltage in high precision, we added a manufacturer defined function that can alter the memory sensing reference voltage (MSRV). The NAND flash memory controller can measure the threshold voltages within the range of 0.1 V to 4.4 V with the precision of 0.04 V.

Figure 4.3 shows the procedure to measure the coupling coefficients from raw

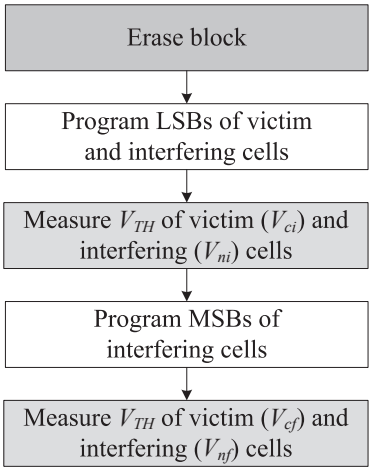


Figure 4.3: Procedure to measure the coupling coefficients.

NAND flash memory chips. The NAND flash memory chip used in the experiments can sense the threshold voltage only in the range of 0.1 V to 4.4 V, in which the erased cells are not included. Thus, we first program the LSB (least significant bit) of the victim and the interfering cells to the symbol *0* so that their threshold voltages are in the measurable range. Let us denote the initial threshold voltages of the victim and the interfering cells as V_{ci} and V_{ni}, respectively. After measuring V_{ci} and V_{ni}, we program only the MSBs (most significant bits) of the interfering cells and read the threshold voltages once again. Let us define V_{cf} and V_{nf} as the threshold voltages of the victim and the interfering cells after the MSB programming. Then, the threshold voltage shifts of the victim and the interfering cells become

$$\Delta V_c = V_{cf} - V_{ci} \tag{4.3}$$

and

$$\Delta V_n = V_{nf} - V_{ni}, \tag{4.4}$$

respectively. Note that ΔV_c and ΔV_n correspond to V_{CCI} and $\Delta V[i, j]$ in Eq. (4.1). During the MSB programming, each interfering cell can be programmed to either the symbol *00* or *10* because the LSB is already programmed to the symbol *0*. When the interfering cell is programmed to the symbol *10*, its threshold voltage shift is larger than the other case. By using ΔV_c and ΔV_n, the coupling coefficient can be computed as follows:

$$C = \frac{\Delta V_c}{\Delta V_n}. \tag{4.5}$$

4.2.2 Experimental Results

We conducted experiments to measure the values of coupling coefficients by using the proposed direct measurement procedure and a NAND flash memory chip with a 26 nm process technology. Figure 4.4 shows the threshold voltage shifts of the victim and the interfering cells when the programming patterns shown in Fig. 4.2-(a) are used. Note that these patterns are designed to measure C_y. The X and Y axes represent ΔV_n and ΔV_c, respectively. Thus, the coupling coefficient can be determined by applying the tangent function to the angle of the data points. The triangles and the circles in Fig. 4.4 represent the cases that the interfering cells are programmed to the symbol *00* and *10*, respectively. In this figure, it is demonstrated that the threshold voltage shift of the victim cell is almost linearly proportional to that of the interfering cell, which is in accordance with the experimental results of previous studies [2, 36, 45].

By using Eq. (4.5) and the data samples shown in Fig. 4.4, we can compute C_y and plot its probability density function (PDF) as depicted in Fig. 4.5. Similar to the results in [36, 45], C_y varies depending on the physical locations of the victim cells

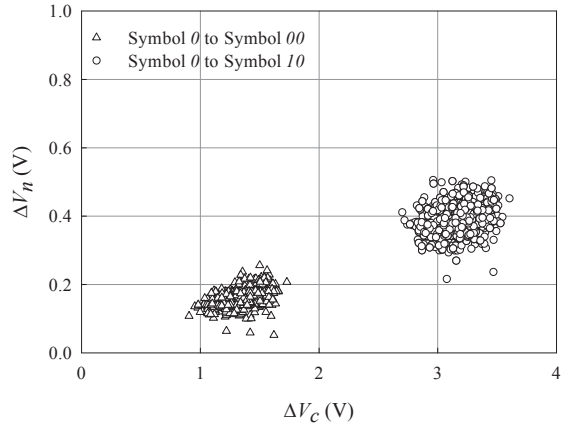


Figure 4.4: Threshold voltage shifts of the victim (y-axis) and the interfering (x-axis) cells when using the programming patterns that are designed to measure C_y.

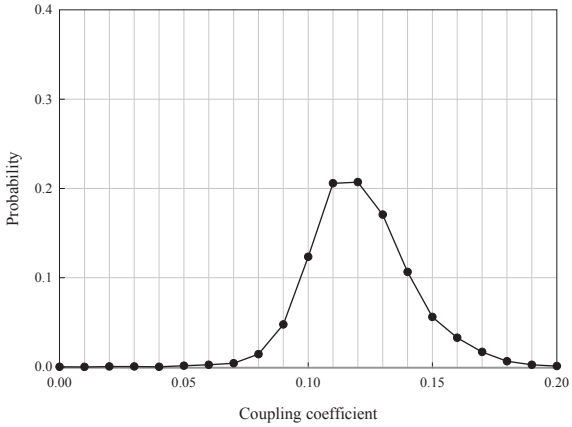


Figure 4.5: Measured probability density function of C_y.

and has the PDF that is similar to the Gaussian function. It is well known that the random line edge roughness effect caused by lithography and etching is the main reason of the variation for coupling coefficients [45]. For the NAND flash memory chip that we use, the mean and the standard deviation values of C_y are 0.1239 and 0.0201, respectively.

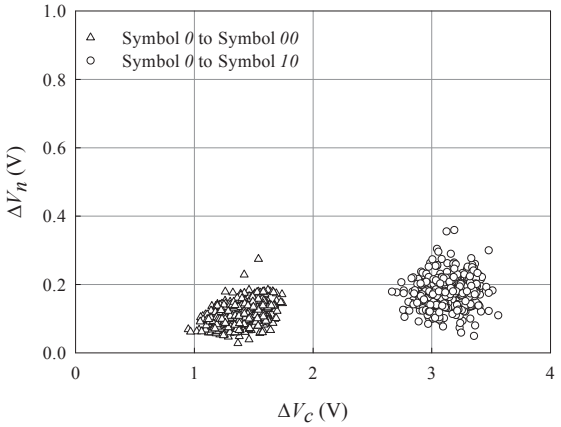


Figure 4.6: Threshold voltage shifts of the victim (y-axis) and the interfering (x-axis) cells when using the programming patterns that are designed to measure C_x.

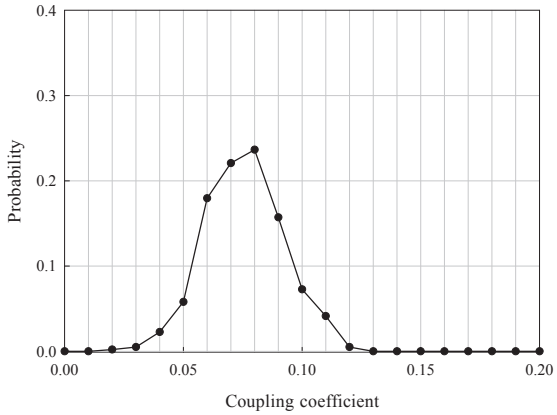


Figure 4.7: Measured probability density function of C_x.

Figure 4.6 shows ΔV_n and ΔV_c when applying the programming patterns shown in Fig. 4.2-(b). Similar to Fig. 4.4, ΔV_c and ΔV_n show a linear relationship, which is also in accordance with the experimental results of the previous researches [2, 36, 45]. As shown in Fig. 4.7, the PDF of C_x also can be approximated to the Gaussian function. The mean and the standard deviation values of C_x are 0.081 and 0.0169, respectively,

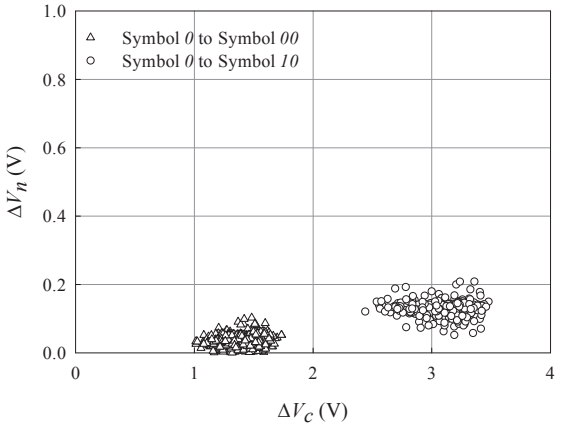


Figure 4.8: Threshold voltage shifts of the victim (y-axis) and the interfering (x-axis) cells when using the programming patterns that are designed to measure C_{xy}.

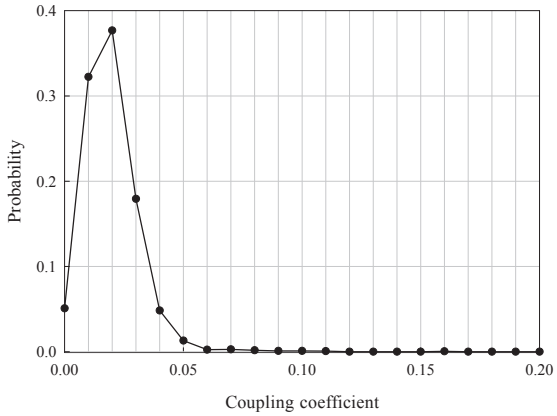


Figure 4.9: Measured probability density function of C_{xy}.

and they are smaller than those of C_y. Note that the similar results were reported in [3, 45].

Figures 4.8 and 4.9 show threshold voltage shifts and the measured PDF of C_{xy} when using the programming patterns shown in Fig. 4.2-(c). The experimental results show that C_{xy} is much smaller than C_y or C_x. Note that the mean and the standard

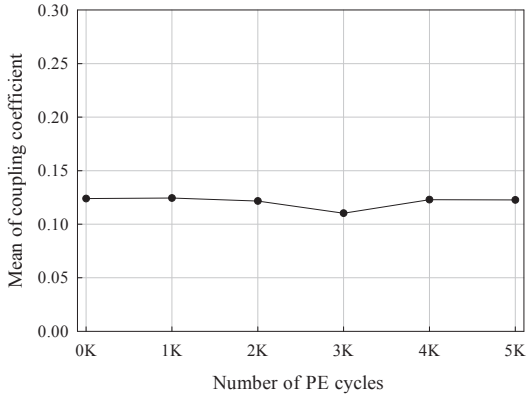


Figure 4.10: Mean values of C_y when increasing the number of PE cycles from 0 K to 5 K.

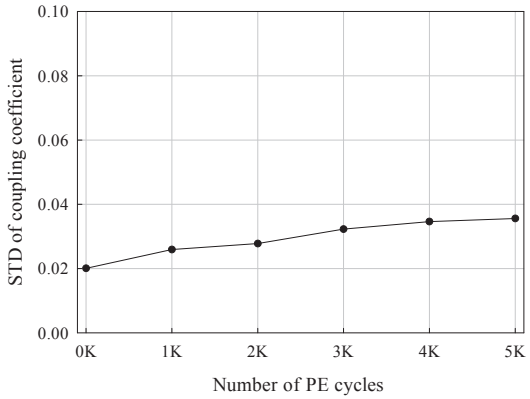


Figure 4.11: Standard deviation values of C_y when increasing the number of PE cycles from 0 K to 5 K.

deviation values of C_{xy} are 0.0273 and 0.0102, respectively. The standard deviation of C_{xy} is relatively large and is almost 40 % of the mean value. This is because the amount of CCI induced by the interfering cells in the *xy* direction is too small, and as a result, the quantization effect causes relatively large errors when measuring ΔV_c.

In order to characterize the effect of program-erase (PE) cycling to coupling coefficients, we measure C_y while increasing the number of PE cycles from 0 K to 5 K.

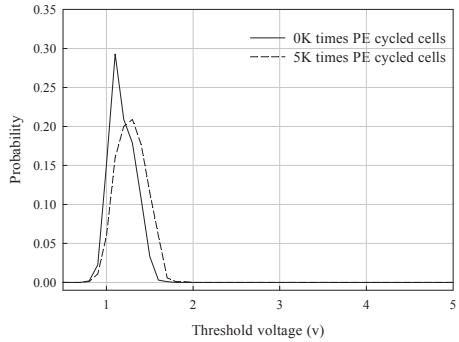


Figure 4.12: Threshold voltage distributions when only the LSB of victim cell is programmed to the symbol *0*.

We can observe that the mean of C_y remains almost the same as shown in Fig. 4.10. This experimental result demonstrates that PE cycling does not alter the coupling coefficients. Figure 4.11 shows the standard deviation values of C_y when increasing the number of PE cycles from 0 K to 5 K. Unlike the mean, the standard deviation increases as the number of PE cycles grows.

To identify the main reason of the variance increase in C_y, we program only the LSBs of the victim cells to the symbol *0* and measure the threshold voltage distribution. Figure 4.12 shows the threshold voltage distribution of fresh and 5 K times PE-cycled victim cells. As the number of PE cycles increases, the threshold voltage distribution becomes wider even without the effect of CCI. This result shows that the read disturb and the RTN (random telegraph noise) induce variations in the measured threshold voltages. The mean values of these noises are cancelled out when computing ΔV_c and ΔV_n. However, the variance of ΔV_c (ΔV_n) becomes larger as V_{cf} (V_{nf}) is subtracted by V_{ci} (V_{ni}).

In summary, the coupling coefficients vary depending on the physical locations of the memory cells. However, they do not seem to be affected by PE cycling. For

a NAND flash memory chip with a 26 nm process technology, we obtained 0.1239, 0.0810, and 0.023 as the mean values of coupling coefficients in the y, x, and xy directions, respectively. Note that two neighboring cells in each of x and xy directions induce the CCI, while only one cell in the y direction causes the capacitance coupling effect to the victim cell. We also found that the standard deviation of the coupling coefficient is approximately 20 % of the mean value in the NAND flash memory chip that we use.

4.3 Least Squares Method based Coupling Coefficient Estimation

In this section, we develop a least squares (LS) method based coupling coefficient estimation algorithm. In the direct approach, only one of the coupling coefficients is measured at a time by sensing the threshold voltages of the victim and the interfering cells multiple times. As a result, this approach requires quite large overheads for the memory sensing operations. Unlike the direct approach that employs specific programming patterns, the LS algorithm utilizes ordinary data patterns for fast estimation of the coupling coefficients. In the LS approach, the coupling coefficients of the x, y, and xy directions can be obtained at once by measuring the threshold voltages only one time, which reduces the memory sensing overheads.

In order to derive the least squares based approach, let us rewrite the amount of CCI that the n-th victim cell receives as follows:

$$V_{CCI}[n] = \sum_{i=0}^{M-1} C_i \cdot \Delta V[i], \tag{4.6}$$

where the number of interfering cells M is either five or three for the even or odd page. In Eq. (4.6), $\Delta V[i]$ denotes the threshold voltage shift of the i-th neighboring cell during the MSB programming. Recall that $\Delta V[i]$ is equal to $V_{TH}[i] - V_L[i]$ as shown in Fig. 4.1. In the direct approach, we can obtain $V_L[i]$, which corresponds to V_{ni}, by measuring the threshold voltage of the interfering cell after the LSB programming. When using the ordinary data patterns, however, we only measure $V_{TH}[i]$, which is the threshold voltage after the MSB programming. Thus, the LS algorithm uses $E[V_L[i]|X_L[i]]$ instead of $V_L[i]$. As a result, Eq. (4.6) becomes

$$\hat{V}_{CCI}[n] = \sum_{i=0}^{M-1} C_i \cdot \Big\{ V_{TH}[i] - E\big[V_L[i]|X_L[i]\big] \Big\}, \tag{4.7}$$

where $X_L[i]$ represents the pre-determined LSB symbol of the i-th neighbor cell. Depending on $X_L[i]$, which is either the symbol *1* or *0*, $E\big[V_L[i]|X_L[i]\big]$ can be either V_{L1} or V_{L0} as shown in Fig. 4.1. We assume that V_{L1} and V_{L0} are known in advance. In order to determine the LSB of the i-th neighbor cell, the observed threshold voltage $V_{TH}[i]$ can be used.

To find accurate values of C_i, we need to know $V_{CCI}[n]$ precisely. Since $V_{CCI}[n]$ is the threshold voltage shift of the n-th victim cell, we can estimate it by using the following equation:

$$V_{CCI}[n] = V_{TH}[n] - V_M[n], \tag{4.8}$$

where $V_M[n]$ is the threshold voltage of the n-th victim cell before affected by the CCI. If we use the programming patterns shown in Fig. 4.2, we can achieve accurate values of $V_{CCI}[n]$, which corresponds to ΔV_c, by measuring the threshold voltage of the victim cell (V_{ci} and V_{cf}). On the other hand, it is not straightforward to directly

measure $V_{CCI}[n]$ in the LS approach. Thus, the LS algorithm uses $E[V_M|X[n]]$ instead of $V_M[n]$ as follows:

$$\bar{V}_{CCI}[n] = V_{TH}[n] - E[V_M|X[n]], \tag{4.9}$$

where $X[n]$ denotes the pre-determined symbol of the n-th victim cell. Computing $E[V_M|X[n]]$ is simple because the incremental step pulse programming (ISPP) scheme results in a uniform distribution with the width of ΔV_{pp}. For example, if the input symbol $X[n]$ is *01*, $E[V_M|X[n] = 01]$ becomes $V_{01} + \frac{1}{2}\Delta V_{pp}$.

Since both $\hat{V}_{CCI}[n]$ and $\bar{V}_{CCI}[n]$ are estimators of $V_{CCI}[n]$, their difference can be used to define a cost function:

$$J = \sum_{n=0}^{N_s-1} \left(\hat{V}_{CCI}[n] - \bar{V}_{CCI}[n]\right)^2. \tag{4.10}$$

The coupling coefficient C_i can be determined by minimizing the cost function. Since both $\hat{V}_{CCI}[n]$ and $\bar{V}_{CCI}[n]$ are not the exact but estimated ones, a large number of data is required. By averaging the estimation errors of N_s data samples, we can expect a more reliable solution in the proposed LS approach. Note that N_s can be quite large because the coupling coefficients need to be estimated only once in the life time of the target NAND flash memory chip.

In order to find the solution for Eq. (4.10), let us define a vector as

$$\mathbf{a}[n] = \left[\Delta V_0[n], \Delta V_1[n], \cdots, \Delta V_{M-1}[n]\right]^T, \tag{4.11}$$

where $\Delta V_i[n]$ is the threshold voltage shift of the i-th neighboring cell for the n-th

victim cell. Then, we can rewrite the right hand side of Eq. (4.7) as

$$\hat{V}_{CCI}[n] = \mathbf{a}[n]^T \cdot \mathbf{x}, \tag{4.12}$$

where

$$\mathbf{x} = \left[C_0, C_1, \cdots, C_{M-1}\right]^T. \tag{4.13}$$

By using the above definition, Eq. (4.10) can be transformed into a matrix-vector form:

$$J_m(\mathbf{x}) = (\mathbf{A} \cdot \mathbf{x} - \mathbf{b})^T \cdot (\mathbf{A} \cdot \mathbf{x} - \mathbf{b}), \tag{4.14}$$

where

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}[0]^T \\ \mathbf{a}[1]^T \\ \vdots \\ \mathbf{a}[N_s - 1]^T \end{bmatrix} \text{ and } \mathbf{b}_m = \begin{bmatrix} \bar{V}_{CCI}[0] \\ \bar{V}_{CCI}[1] \\ \vdots \\ \bar{V}_{CCI}[N_s - 1] \end{bmatrix}. \tag{4.15}$$

Note that $\mathbf{A}$ is an N_s by M matrix and $\mathbf{b}$ is an N_s dimensional vector. Equation (4.14) is known as the least squares problem, and many algorithms have been developed to find $\mathbf{x}$ that minimizes the cost function. Since Eq. (4.14) is a linear system, the analytic solution can be derived as

$$\mathbf{x}^* = (\mathbf{A}^T \cdot \mathbf{A})^{-1} \cdot \mathbf{A}^T \cdot \mathbf{b}. \tag{4.16}$$

Once the optimal solution $\mathbf{x}^*$ is computed, the estimation step does not need to be re-conducted because the coupling coefficients remain almost the same even with excessive PE cycling. The coupling coefficient estimation step demands $N_s M^2$ and

M^3 arithmetic operations for computing $\mathbf{A}^T \cdot \mathbf{A}$ and the matrix inverse, respectively. Between them, the former term, $N_s M^2$, is dominant because N_s is usually much larger than M, thus the time complexity of the coefficient estimation step can be modeled as $O(N_s M^2)$.

4.4 Multi-Level Memory Sensing Schemes for CCI Cancellation

The bit error rate performance of the proposed CCI cancellation algorithm is strongly affected by the precision of the sensed threshold voltage signals. As the number of memory sensing operations increases, the quantized threshold voltage V_Q is close to the original threshold voltage, and the quantization noises in the CCI removal step become smaller. However, increasing precision of the sensed signals requires additional memory sensing operations. Therefore, it is very needed to find the optimal quantization schemes for reducing the number of memory sensing operations, while maintaining the performance of the proposed CCI cancellation algorithm.

The optimal memory sensing schemes for the proposed CCI cancellation algorithm depend on the physical locations of the sensed cells; the victim and the neighboring cells may require different quantization schemes. For the coefficient estimation step, which is conducted only once, we can apply many memory sensing operations to obtain reliable coupling coefficients. On the other hand, the CCI removal step is conducted at every page read request, thus it is important to reduce the number of memory sensing operations. In this research, we consider two memory sensing schemes for the CCI removal step.

The main purpose of CCI removal is to achieve a satisfactory post-FEC (forward

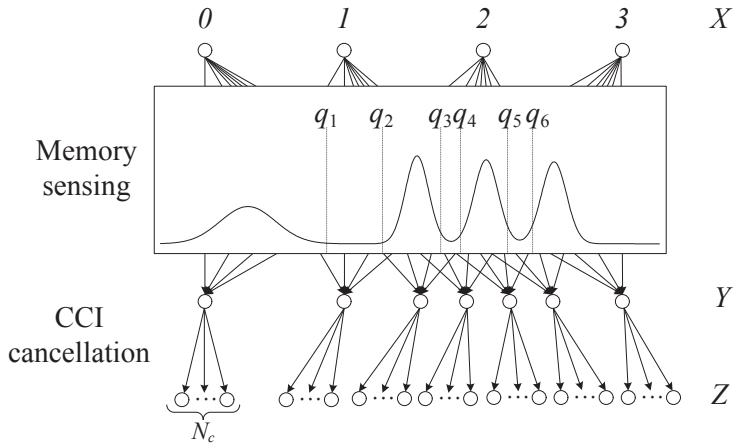


Figure 4.13: Channel model for the 7-level memory sensing and the CCI cancellation.

error correction) BER performance. According to [30], the maximizing mutual information (MMI) quantization scheme that employs unequal quantization steps can yield near optimal error correcting performance for soft-decision error correction. Based on this observation, we also assume that the proposed CCI cancellation algorithm can result in the optimal error performance when the mutual information between the input and the CCI canceller output is maximized.

Figure 4.13 illustrates the quantized channel model that includes the CCI canceller and 7-level memory sensing that uses the MSRVs of q_1, ... , q_6 for the victim cells. Among the seven voltage levels, $(q_1, q_2]$, $(q_3, q_4]$, and $(q_5, q_6]$ are erasure regions. In this figure, *X*, *Y*, and *Z* denote the input, the memory sensing output, and the CCI canceller output symbols, respectively. Note that *X* can be either *0*, *1*, *2*, or *3*, which corresponds to the symbol *11*, *01*, *00*, or *10* in two-bit NAND flash memory. The input symbol undergoes the memory channel and is mapped to one of the seven output symbols *Y*. In Fig. 4.13, the erasures as well as the symbols *0*, *1*, *2*, and *3* can

be Y. Suppose that each memory sensing output symbol Y is further divided into N_c distinct symbols. Thus, there are $7N_C$ distinct symbols of Z in this channel model.

Let us denote $P_{i,j}$ as the probability that the input symbol i is mapped to the quantization output symbol j. If X is equally likely to be *0*, *1*, *2*, or *3*, the mutual information between X and Y can be computed as follows [30]:

$$\begin{aligned} I(X;Y) &= H(Y) - H(Y|X) \\ &= H(P_0,\, P_1,\, \cdots,\, P_6) - \frac{1}{4}\sum_{i=0}^{3} H(P_{i,0},\, P_{i,1},\, \cdots,\, P_{i,6}), \end{aligned} \tag{4.17}$$

where $P_j = \frac{1}{4}\sum_{i=0}^{3} P_{i,j}$. Note that $H(X)$ represents the entropy of the discrete random variable X. Let us assume that N_C distinct symbols that are originated from the same Y are equiprobable. Then, the mutual information between X and Z becomes

$$\begin{aligned} I(X;Z) &= H(Z) - H(Z|X) \\ &= H\Big(\frac{P_0}{N_C},\cdots,\frac{P_0}{N_C},\cdots,\frac{P_6}{N_C},\cdots,\frac{P_6}{N_C}\Big) - \frac{1}{4}\sum_{i=0}^{3} H\Big(\frac{P_{i,0}}{N_C},\cdots,\frac{P_{i,0}}{N_C},\cdots,\frac{P_{i,6}}{N_C},\cdots,\frac{P_{i,6}}{N_C}\Big). \end{aligned} \tag{4.18}$$

The right hand side of Eq. (4.18) can be further simplified as

$$\begin{aligned} &-\sum_{j=0}^{6} N_C\Big(\frac{P_j}{N_C}\log\frac{P_j}{N_C}\Big) + \frac{1}{4}\sum_{i=0}^{3}\sum_{j=0}^{6} N_C\Big(\frac{P_{i,j}}{N_C}\log\frac{P_{i,j}}{N_C}\Big) \\ &= I(X;Y). \end{aligned} \tag{4.19}$$

According to the data processing inequality [46], $I(X;Z)$ is always smaller than or equal to $I(X;Y)$. Thus, the maximum value of $I(X;Z)$ can be achieved when every symbol of Z is equally likely. Let us denote the symbol of the i-th neighbor cell as Z_i,

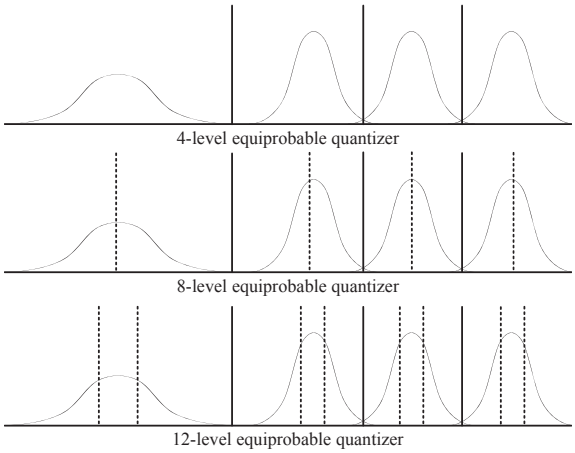


Figure 4.14: 4-, 8-, 12-level equiprobable quantizers for the neighboring cells.

then the joint PMF (probability mass function) of the neighboring symbols becomes

$$f_{Z_1,\cdots,Z_M}(k_1,\cdots,k_M) = f_{Z_1}(k_1)\cdots f_{Z_M}(k_M), \tag{4.20}$$

where $f_{Z_i}(k_i)$ is the PMF of the i-th neighbor and k_i is 0, 1, $\cdots$, or $N_{qn}-1$ assuming that an N_{qn}-level quantization is applied. If each symbol of the neighbor cells is equally likely, the joint PMF $f_{K_1,\cdots,K_M}(k_1,\cdots,k_M)$ is also a constant value and Eq. (4.19) holds.

The optimal memory sensing schemes for the CCI removal step can be found by maximizing the mutual information between the input and the CCI canceller output. When each symbol of the neighboring cells is equally likely, $I(X;Z)$ is equal to its maximum value $I(X;Y)$. Thus, the equiprobable quantizers shown in Fig. 4.14 become the optimal memory sensing schemes for the neighboring cells. When the number of quantization levels is 4 and 8, the equiprobable qauntizers are almost the same as the uniform ones. In the meanwhile, the optimal quantizer for the victim cells

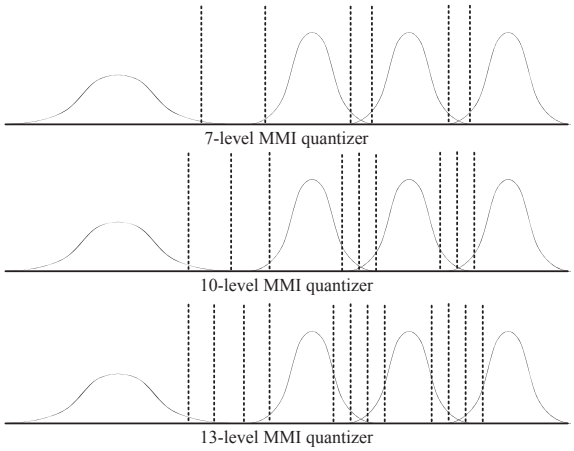


Figure 4.15: 7-, 10-, 13-level MMI quantizers for the victim cells.

is the MMI quantizer that maximizes the mutual information between the input X and the memory sensing output Y as shown in Fig. 4.15.

4.5 Experimental Results

We conducted experiments to evaluate the performance of the proposed CCI cancellation algorithm. We used data samples that were obtained from the simulated NAND flash memory model described in Chapter 2 and the actual NAND flash memory devices with a 20 nm process technology.

4.5.1 CCI Cancellation with Simulated NAND Flash Memory

We apply the proposed CCI cancellation algorithm to a two-bit MLC NAND flash memory model described in Chapter 2. We assume that C_x, C_y, and C_{xy} are the Gaussian random variables whose means are $0.0810s$, $0.1231s$, and $0.023s$, respectively, where the coupling coefficient factor, s, varies from 0.6 to 1.6. The standard deviation

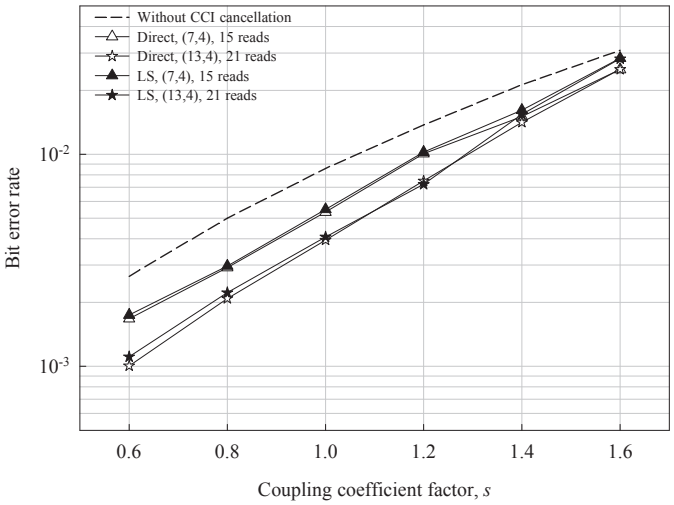
values of coupling coefficients are set to 20 % of their means. Note that the mean and the standard deviation values are obtained from an actual NAND flash memory chip with a 26 nm process technology as explained in Section 4.2. As the feature size of NAND flash memory decreases, the coupling coefficient factor, s, increases. When s is around 1.0, the simulated model is closed to NAND flash memory with a 26 nm process technology.

For the CCI removal step, we need to apply multi-level memory sensing to measure the threshold voltages in the victim and the neighboring cells. The required numbers of memory sensing operations are

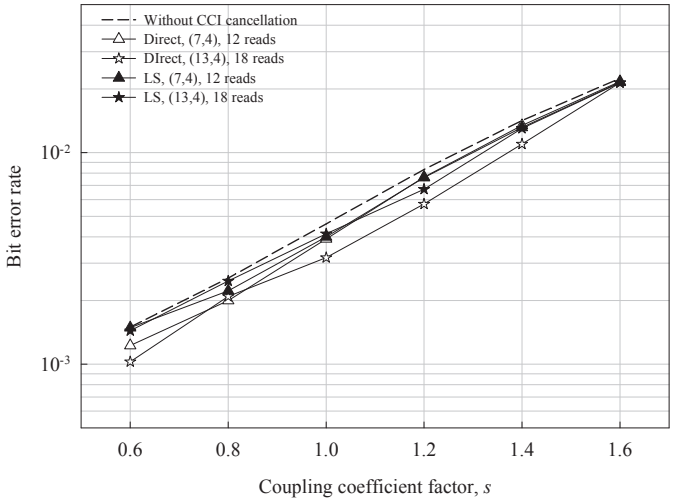
$$\begin{cases} N_{qc} + 3 \cdot N_{qn} - 4 & \text{for the even victim cells,} \\ N_{qc} + 2 \cdot N_{qn} - 3 & \text{for the odd victim cells,} \end{cases} \tag{4.21}$$

where N_{qc} and N_{qn} are the numbers of quantization levels for the victim and the neighboring cells, respectively. Increasing N_{qn} demands three times or twice many memory sensing operations than that of N_{qc}, thus it is needed to minimize N_{qn}. For the CCI removal step, the 7-, 13-, and 19-level MMI quantizers are applied to the victim cells, while only the 4-level uniform quantization scheme is used for the neighboring cells. To evaluate the BER performance of NAND flash memory with various feature sizes, we apply the proposed CCI cancellation algorithm to the simulated memory model whose coupling coefficient factor, s, varies from 0.6 to 1.6.

Figure 4.16 shows the BER performances for the even and the odd victim cells when applying the proposed CCI cancellation algorithm. We apply both the direct (denoted as 'Direct' in Fig. 4.16) and least squares method (denoted as 'LS' in Fig. 4.16) based approaches to obtain the coupling coefficients. Note that the numbers



(a) even pages



(b) odd pages

Figure 4.16: BERs of the proposed CCI cancellation algorithm when applied to (a) even pages and (b) odd pages of the simulated memory model.

inside the parenthesis represent quantization levels for the victim and the neighboring cells, respectively. For example, '(7,4), 15 reads' represents the case when the 7-level MMI and the 4-level uniform quantizers are used for the victim and the neighboring

cells, respectively, which requires 15 memory sensing operations as a total. In Fig. 4.16, we can find that the proposed CCI cancellation algorithm can significantly lower the BER of even pages especially when s is below 1.4. However, the performance gain becomes smaller as s increases. In these cases, the threshold voltage distribution of each input symbol is heavily overlapped because of strong coupling between adjacent cells. Thus, the number of bit errors is not dramatically reduced by solely applying the proposed CCI cancellation algorithm. We also compare the BER curves for the direct and the LS algorithms. Even though the LS algorithm uses ordinary data rather than well designed programmed patterns shown in Fig. 4.2, the BER performance of the LS approach is almost comparable to that of the direct one especially for the even pages.

When comparing the BERs of the even and the odd victim cells, the proposed CCI cancellation algorithm can correct more bit errors when applied to the even pages. As a result, the BERs of both pages become comparable. Usually, the even victim cells receive more severe CCI from its neighbors, thus we can expect an improved BER performance on the even cells once the CCI is removed. On the other hand, the odd pages are less severely affected by the CCI, and removing the CCI does not lead to large improvement on the BER performance. Since the endurance and the lifetime of NAND flash memory are limited by the worst case BER, which is usually determined by the even pages, we can achieve enhanced PE cycle endurance and increased retention time limit by applying the proposed CCI cancellation algorithm. It is also noteworthy that the '(13,4)-level' one shows a satisfactory error performance and demands a reasonable number of memory sensing operations among the various quantization schemes.

In order to assess the accuracy of coupling coefficients estimated by the LS ap-

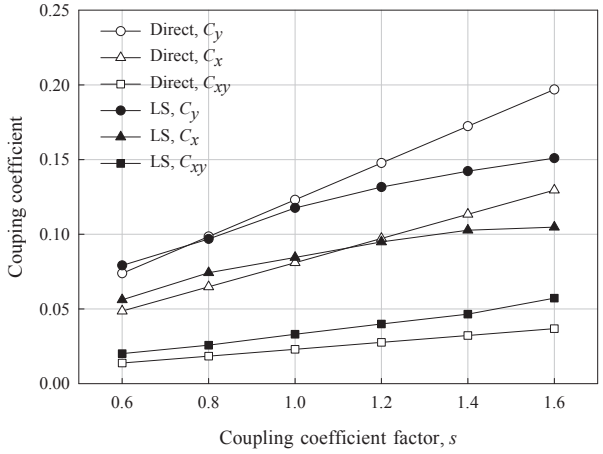
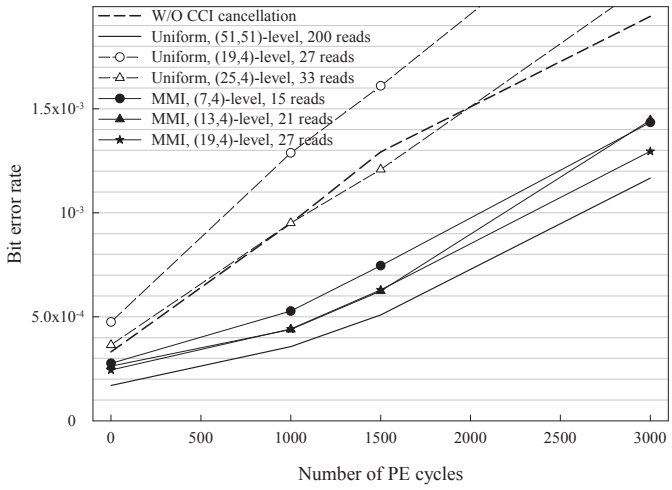
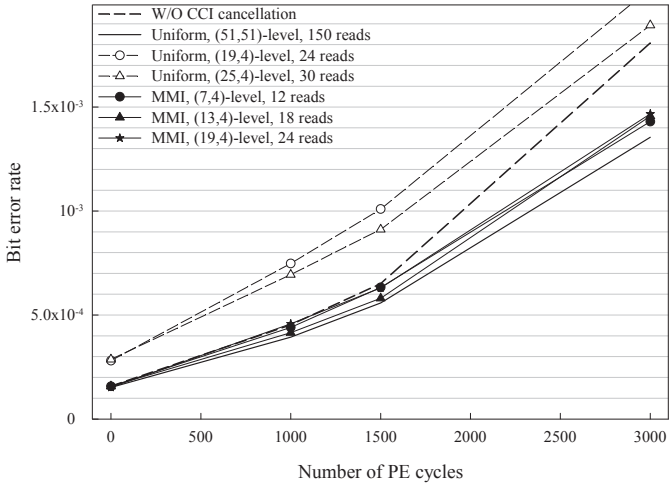


Figure 4.17: Mean values of the coupling coefficients that are obtained by employing the direct ('Direct') and the least squares ('LS') based approaches.

proach, we compare them with the coupling coefficients that are obtained by employing the direct measurement as shown in Fig. 4.17. Note that the direct approach can find the true mean values of coupling coefficients. In this experiment, we change the coupling coefficient factor s from 0.6 to 1.6. When s is smaller than 1.4, where the amount of the CCI is not large, the LS algorithm can find the means of coupling coefficients quite accurately, which is the main reason that the BER performances of both approaches are quite similar. In these cases, the estimation errors for C_x and C_y are smaller than 10 %. As s increases, however, the LS algorithm under-estimates C_x and C_y, thus the estimation errors grow. In the LS algorithm, the pre-determined symbols of the victim and the neighboring cells are used to compute Eq. (4.7) and (4.8). When the CCI is extremely severe, those pre-determined symbols become erroneous, which degrades the estimation accuracy much.



(a) even pages



(b) odd pages

Figure 4.18: BERs of the proposed CCI cancellation when applied to (a) even pages and (b) odd pages of actual NAND flash memory.

4.5.2 CCI Cancellation with Real NAND Flash Memory

We also apply the proposed CCI cancellation algorithm to the data samples that are obtained from the actual two-bit MLC NAND flash memory devices with a 20 nm

process technology. The coupling coefficients are obtained by using the least squares method based estimation algorithm. We measure the BERs after conducting CCI cancellation while changing the number of quantization levels for the CCI removal step. For the victim cells, 7-, 13-, and 19-level MMI quantizers (denoted as 'MMI' in Fig. 4.18) and 19- and 25-level uniform quantization (denoted as 'Uniform' in Fig. 4.18) schemes are employed, while only the 4-level uniform quantizer is used for the neighboring cells. The BER performance bound for the proposed CCI cancellation algorithm is achieved when using the '(51,51)-level' uniform quantizer.

Figure 4.18 shows the error performance of the proposed CCI cancellation algorithm. As N_{qc} increases, the error performances of both the uniform and the MMI quantizers are improved. However, applying the uniform quantizer for memory sensing generates more errors rather than correct them unless the number of quantization levels is fairly large. In Fig. 4.18-(a), we can find that more than 25 levels are required to achieve improved BER performance for the even pages when using the uniform quantization scheme. On the other hand, the MMI quantizer is suitable for CCI cancellation, thus only the 7-level quantization can lead to significant BER reduction. Considering the error performance and the memory sensing overhead, we can find that the 'MMI, (7,4)-level' and 'MMI, (13,4)-level' cases are the optimal memory sensing schemes.

4.6 Concluding Remarks

We provided the statistical characterization of the cell-to-cell interference based on the measured data from an actual NAND flash memory chip with a 26 nm process technology. From the observations, it is demonstrated that the values of coupling co-

efficients vary depending on the physical locations of the memory cells. Moreover, it is also shown that programming-erase cycling does not alter the coupling coefficients. We also developed a CCI cancellation algorithm that estimates the coupling coefficients by using either specific programming patterns or ordinary data and removes the CCI with simple arithmetics. To reduce the memory sensing operations that are needed for the CCI removal step, we studied the optimal multi-level memory sensing schemes. The developed algorithm is applied to both simulated and real NAND flash memory devices and results in significant improvement on the BER performance even with a limited number of memory sensing operations. We also show that the proposed CCI cancellation algorithm improves the worst case BER and extends the lifetime of MLC NAND flash memory.

Chapter 5

Soft-Decision Error Correction in NAND Flash Memory

5.1 Introduction

Hard-decision error correction employing BCH (Bose-Chaudhuri-Hocquenghem) or RS (Reed-Solomon) codes has been widely used in NAND flash memory because these codes can correct a small number of bit errors with low implementation complexity [4, 5, 6, 7, 8]. As the feature size for manufacturing of NAND flash memory decreases, the error correcting capability needs to be increased and hard-decision decoding is no more efficient [9, 10, 11].

In this dissertation, we propose soft-information computation schemes in order to apply soft-decision error correction, which usually shows much improved BER (bit error rate) performance, to NAND flash memory. According to Chapter 2, the threshold voltage distribution of NAND flash memory can be modeled as a Gaussian mixture, thus computing soft-information, such as log likelihood ratio (LLR), is quite

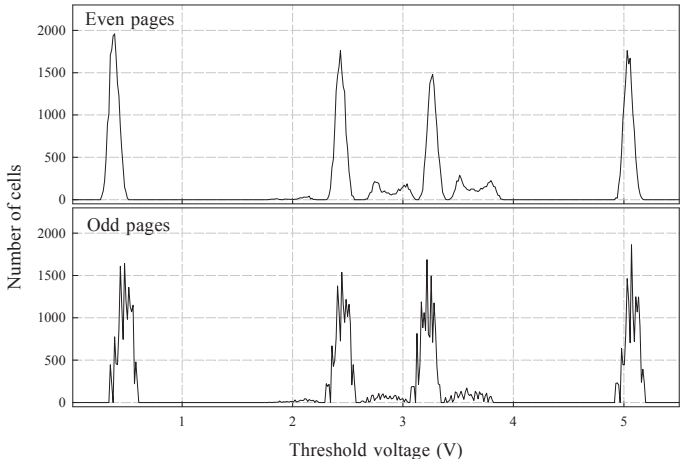


Figure 5.1: Distribution of the CCI removed signals.

straightforward. However, the probability density function (PDF) of the CCI (cell-to-cell interference) removed signal is quite different from that of the original threshold voltage as shown in Fig. 5.1. In this dissertation, we propose two soft-information computation schemes that combine the CCI cancellation and the soft-decision error correction. The first approach derives the PDF of the CCI removed signal, which is shown in Fig. 5.1, and uses them to compute LLRs. The second algorithm jointly conducts CCI cancellation and soft-input computation. The proposed methods are applied to simulated NAND flash memory, and the post-FEC (forward error correction) BER performances are evaluated.

This chapter is organized as follows. Section 5.2 explains an LLR computation scheme when no CCI cancellation techniques are applied. In Section 5.3, we address soft-decision error correction that considers CCI cancellation. Finally, concluding remarks are made in Section 5.4.

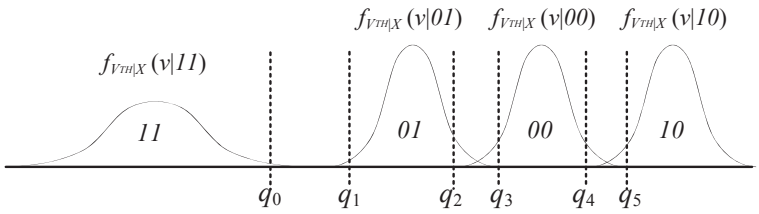


Figure 5.2: Likelihood functions of threshold voltage and quantization boundaries.

5.2 Soft-Decision Error Correction without CCI Cancellation

According to Chapter 2, the threshold voltage distribution can be approximated to a Gaussian mixture. Assume that multiple decision boundaries are employed for discriminating each symbol as illustrated in Fig 5.2. For a memory cell that is sensed at the voltage region of $(q_{i-1}, q_i]$, the quantized LLRs of the least significant bit (LSB) and the most significant bit (MSB) are computed as follows:

$$\Lambda_L(v_q) = \log \frac{\int_{q_{i-1}}^{q_i} f_{V_{TH}|X}(v|11)dv + \int_{q_{i-1}}^{q_i} f_{V_{TH}|X}(v|01)dv}{\int_{q_{i-1}}^{q_i} f_{V_{TH}|X}(v|00)dv + \int_{q_{i-1}}^{q_i} f_{V_{TH}|X}(v|10)dv} \tag{5.1}$$

and

$$\Lambda_M(v_q) = \log \frac{\int_{q_{i-1}}^{q_i} f_{V_{TH}|X}(v|11)dv + \int_{q_{i-1}}^{q_i} f_{V_{TH}|X}(v|10)dv}{\int_{q_{i-1}}^{q_i} f_{V_{TH}|X}(v|01)dv + \int_{q_{i-1}}^{q_i} f_{V_{TH}|X}(v|00)dv}. \tag{5.2}$$

Note that v_q is equal to the representative value of the voltage region $(q_{i-1}, q_i]$. In Eq. (5.1) and (5.2), $f_{V_{TH}|X}(v|k)$ is the likelihood function and can be modeled as a Gaussian distribution as shown in Fig. 5.2. Note that X represents the input symbol and k corresponds to the symbol *11*, *01*, *00*, or *10*. Since the sum of partial CDFs (cumulative density functions) for $f_{V_{TH}|X}(v|11)$ and $f_{V_{TH}|X}(v|01)$ results in the proba-

bility of the symbol *1* in LSB, thus it becomes the numerator of Eq. (5.1). Similarly, the sum of partial CDFs for $f_{V_{TH}|X}(v|10)$ and $f_{V_{TH}|X}(v|00)$ is the probability of the symbol *0* in LSB, thus it comes at the denominator of Eq. (5.1). The computation of quantized LLR for MSB can be conducted in the same way.

Accurate assessment of soft-information is important for soft-decision decoding to obtain the best error correcting performance. Since the likelihood function $f_{V_{TH}|X}(x|k)$ can be obtained by employing the threshold voltage distribution estimation algorithms explained in Chapter 3, the accurate estimation of means and standard deviations is important to obtain reliable LLR values. In order to know the effects of accurate SNR (signal-to-noise ratio) information for the corrected BER, we compare the two approaches. One approach utilizes the estimated means and standard deviations (m^* and σ^*), while the compared one uses the estimated means and the initial (non-updated) standard deviations. We can update the means by employing the conventional moving read technique [29]. Note that the 'non-updated mean' case is not considered because it yields a very poor performance resulting from incorrect decision boundaries.

Figure 5.3 shows the corrected BERs of a (68254, 65536) EG-LDPC code [11] when applying hard-/soft-decision error correction to simulated NAND flash memory. In this figure, the '4-levels', '7-levels', '13-levels', and '16-levels' denote the maximizing mutual information (MMI) quantization schemes that employ only one, two, four, and five memory sensing operations at each symbol boundary, respectively. Note that '4-levels' represents the hard-decision error correction. We can find that increasing the sensing precision improves the error correcting performance. For example, the '7-level' case increases the maximum tolerable retention time more than 10 times when compared to the '4-level' case. We can also find that utilizing more

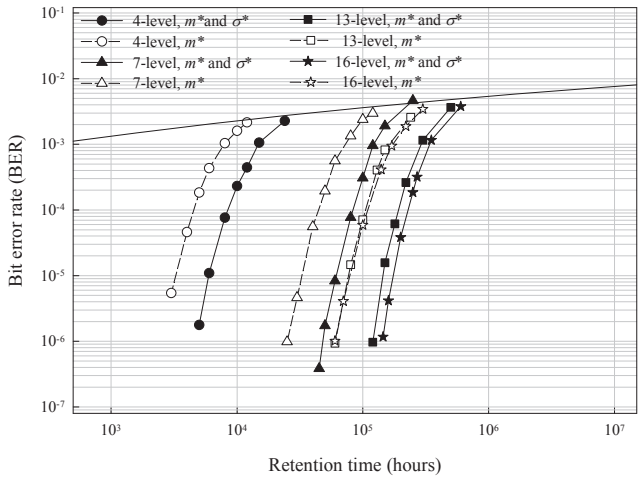


Figure 5.3: Error performance of the (68254,65536) EG-LDPC code for even MSB pages of NAND flash memory.

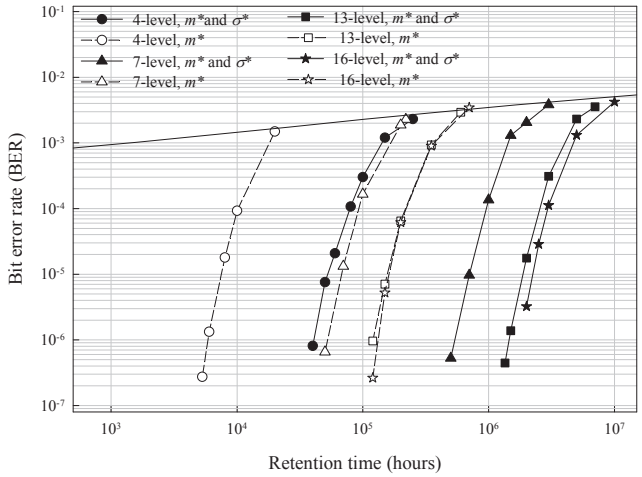


Figure 5.4: Error performance of the (68254,65536) EG-LDPC code for odd MSB pages of NAND flash memory.

accurate statistical information can improve the corrected BER significantly. As a result, we can expect approximately 200 % longer retention time limits by providing accurate standard deviation values to the soft-decision error correction decoder.

These results show that not only the mean but also the standard deviation values are indispensable for soft-decision error correction in NAND flash memory.

Figure 5.4 shows the error performance for odd MSB pages. Similar to Fig. 5.3, we can improve the BER performance significantly by providing accurate standard deviation values to the error correction decoder, which results in almost 10 times longer retention time limits.

5.3 Soft-Decision Error Correction with CCI Cancellation

The distribution of the CCI removed signal is quite different from the original one as depicted in Fig. 5.1, thus computing LLR is not straightforward in this case. In this section, we develop two soft-information computation schemes that consider CCI cancellation. In the first approach, we directly use the likelihood function of the CCI removed signal to compute the LLR values. On the other hand, the second approach utilizes the likelihood functions of the original signals that are conditioned on the symbols of the neighboring cells.

5.3.1 Soft-Information Computation using PDF of CCI Removed Signal

A straightforward approach to compute the LLRs is to derive the mathematical formulation for the distribution of the CCI removed signals. According to Chapter 4, the CCI removed signal can be represented as follows:

$$\begin{aligned} V_O &= V_Q - V_{CCI} \\ &= V_Q - \sum_{i=0}^{M-1} C_i \cdot \Delta V[i]. \end{aligned} \tag{5.3}$$

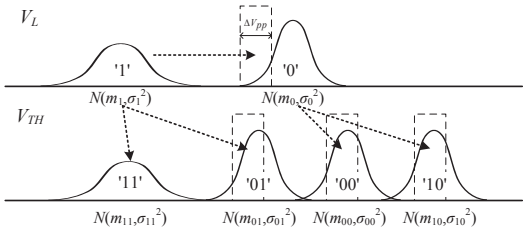


Figure 5.5: Multi-page programming scheme.

In Eq. (5.3), V_Q represents the quantized threshold voltage of the victim cell that is obtained by applying multi-level memory sensing. $\Delta V[i]$ is the threshold voltage shift of the i-th neighboring cell, and C_i is the estimated coupling coefficient. Note that M represents the number of interfering cells and is either 5 (for the even victim cells) or 3 (for the odd victim cells) [2].

Let us denote the likelihood function of V_{CCI} as $f_{V_{CCI}|X}(v|k)$. Since $\Delta V[i]$ is *i.i.d* (independent and identically distributed), we can derive the PDF of V_{CCI} by using that of $\Delta V[i]$. If we apply the multi-page programming scheme shown in Fig. 5.5, $\Delta V[i]$ is equal to $V_{TH}[i] - V_L[i]$. Thus, the PDFs of $\Delta V[i]$ can be approximated to a Gaussian mixture as follows:

$$f_{\Delta V}(v) = \sum_k P_k \cdot N(m_k - m_{k_L}, \sigma_k^2 + \sigma_{k_L}^2), \tag{5.4}$$

where k_L and P_k denote the LSB symbol and the probability of the input symbol k, respectively. Note that $N(m, \sigma^2)$ represents a Gaussian distribution whose mean and variance are m and σ^2, respectively. By using Eq. (5.3), the mean and the variance of V_{CCI} can be computed as follows:

$$E[V_{CCI}] = m_{V_{CCI}} = \sum_{i=0}^{M-1} C_i \cdot E\Big[\Delta V[i]\Big] \tag{5.5}$$

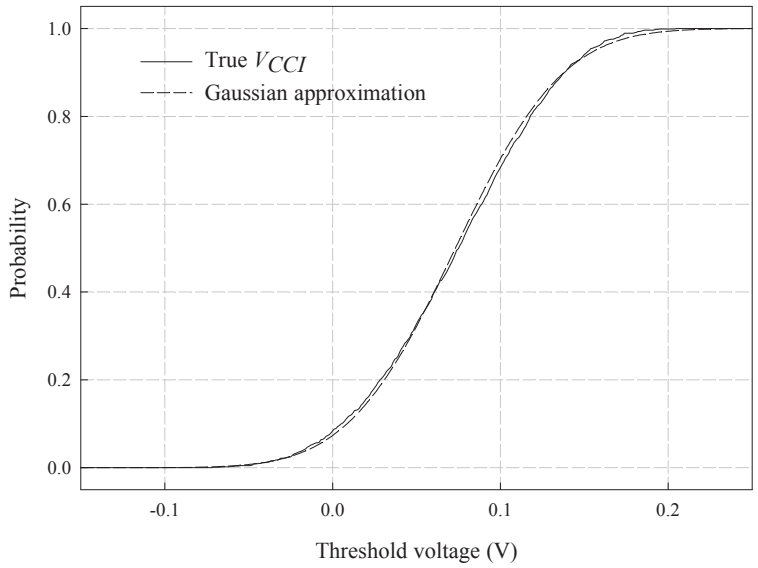


Figure 5.6: Cumulative density functions of V_{CCI} and its Gaussian approximation.

and

$$Var[V_{CCI}] = \sigma^2_{V_{CCI}} = \sum_{i=0}^{M-1} C_i^2 \cdot Var\Big[\Delta V[i]\Big]. \tag{5.6}$$

Since V_{CCI} is independent from the input symbol of the victim cell X, $f_{V_{CCI}|X}(v|k)$ is equal to $N(m_{V_{CCI}}, \sigma^2_{V_{CCI}})$. Figure 5.6 shows the CDFs of V_{CCI} and its Gaussian approximation, where we can find that they are quite similar.

Next, let us derive the likelihood function of V_Q. Recall that V_Q is the output of memory sensing applied to the victim cell. Let us assume that an N_q-level MMI quantizer is used for memory sensing. Then, the likelihood function of V_Q becomes

$$f_{V_Q|X}(v|k) = \sum_{j=0}^{N_q-1} P_{k,j} \cdot \delta(v - r_j), \tag{5.7}$$

where r_j denotes the representative value of the voltage region $(q_{j-1}, q_j]$. Note that

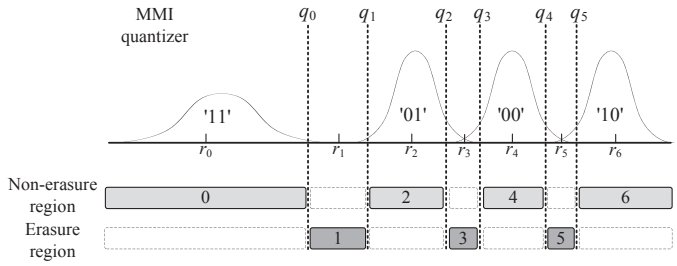


Figure 5.7: 7-level MMI quantization scheme.

$P_{k,j}$ is the probability that the input symbol k falls into the voltage region j and $\delta(x)$ is the delta function.

If V_{CCI} and V_Q are independent each other, the likelihood function of V_O becomes the convolution of $f_{V_Q|X}(v|k)$ and $f_{V_{CCI}|X}(-v|k)$. In reality, however, V_{CCI} and V_Q are not independent. If a cell whose input symbol is *01* has a threshold voltage at the region 3, it is highly likely that the cell has received quite large CCI. On the other hand, if one cell has an input symbol *00* and is observed at the voltage region 3, the amount of CCI is probably small.

In order to compensate the correlation between V_{CCI} and V_Q, we modify the likelihood function of V_Q as follows:

$$f_{V_Q|X}(v|k) = \sum_{j=0}^{N_q-1} P_{k,j} \cdot \delta(v - r_{k,j}), \tag{5.8}$$

where

$$r_{k,j} = \begin{cases} G_k(q_{j-1}, q_j) & \text{if } (q_{j-1}, q_j] \text{ is an erasure region,} \\ r_j & \text{otherwise.} \end{cases} \tag{5.9}$$

In Eq. (5.9), $G_k(q_{j-1}, q_j)$ represents the center of mass or the centroid of the region

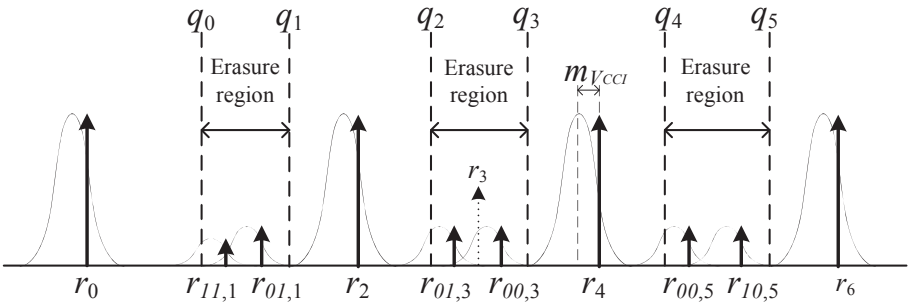


Figure 5.8: Estimated distribution of the CCI removed signals.

$(q_{j-1}, q_j]$ and can be computed as

$$G_k(q_{j-1}, q_j) = \frac{\int_{q_{j-1}}^{q_j} v f_{V_{TH}|X}(v|k)dv}{\int_{q_{j-1}}^{q_j} f_{V_{TH}|X}(v|k)dv}. \tag{5.10}$$

In Eq. (5.8), the representative value of the erasure region is divided into two distinct ones depending on the input symbol. For example, r_3 (dotted arrow in Fig. 5.8) is divided into $r_{01,3}$ and $r_{00,3}$ (solid arrows in Fig. 5.8). By subtracting V_{CCI} from $r_{01,3}$ ($r_{00,3}$) rather than r_3, we can compensate the effect of large (small) CCI on the cells that have input symbols of *01* (*00*) and are observed at the voltage region 3.

By using $f_{V_Q|X}(v|k)$, $f_{V_{CCI}|X}(v|k)$, and Eq. (5.3), we can obtain the likelihood function of V_O as follows:

$$f_{V_O|X}(v|k) = f_{V_Q|X}(v|k) \bigotimes f_{V_{CCI}|X}(-v|k). \tag{5.11}$$

Note that $\bigotimes$ denotes the convolution operation. Figure 5.8 shows the threshold voltage distribution after the convolution.

We can compute soft-information for LSB and MSB pages by using the likeli-

hood functions of the CCI removed signal as follows:

$$\Lambda_L(v) = \begin{cases} \Lambda_{MAX} & \text{if } v < r_2 - m_{V_{CCI}}, \\ \ln \frac{f_{V_O|X}(v|01)}{f_{V_O|X}(v|00)} & \text{if } r_2 - m_{V_{CCI}} \leq v < r_4 - m_{V_{CCI}}, \\ -\Lambda_{MAX} & \text{if } r_4 - m_{V_{CCI}} \leq v, \end{cases} \tag{5.12}$$

and

$$\Lambda_M(v) = \begin{cases} \Lambda_{MAX} & \text{if } v < r_0 - m_{V_{CCI}}, \\ \ln \frac{f_{V_O|X}(v|11)}{f_{V_O|X}(v|01)} & \text{if } r_0 - m_{V_{CCI}} \leq v < r_2 - m_{V_{CCI}}, \\ -\Lambda_{MAX} & \text{if } r_2 - m_{V_{CCI}} \leq v < r_4 - m_{V_{CCI}}, \\ \ln \frac{f_{V_O|X}(v|10)}{f_{V_O|X}(v|00)} & \text{if } r_4 - m_{V_{CCI}} \leq v < r_6 - m_{V_{CCI}}, \\ \Lambda_{MAX} & \text{if } r_6 - m_{V_{CCI}} \leq v, \end{cases} \tag{5.13}$$

where Λ_{MAX} is a large positive constant. In order to reduce the computational overheads and offer more reliable soft-information, the LLR is set to Λ_{MAX} or $-\Lambda_{MAX}$ when the decoding output is quite obvious. Equations (5.12) and (5.13) are for the case when 7-level voltage sensing is applied to the victim cells, and these equations can be modified quite easily when more voltage sensing operations are applied. However, applying the proposed method with 4-level memory sensing is not possible because no erasure regions exist in this case.

5.3.2 Joint CCI Cancellation and Soft-Information Computation

In this section, we also propose a joint CCI cancellation and soft-information computation (JCS) scheme that utilizes the memory sensing output of not only the victim but also the neighboring cells to find more precise threshold voltage distributions. Unlike the previous approach that separately conducts CCI cancellation and soft-decision error correction, the JCS algorithm combines them into one. In the JCS scheme, the likelihood function of V_{TH} for the given sensing output of the neighboring cells, $f_{V_{TH}|\mathbf{X},\mathbf{Z_1},\cdots,\mathbf{Z_M}}(v|k_0,k_1,\cdots,k_M)$, is used when computing the LLR. Note that $\mathbf{X}$ and $\mathbf{Z_i}$ represent the symbols of the victim and the i-th neighboring cells, respectively, and k_i, which is the symbol *11*, *01*, *00*, or *10*, can be found by making hard-decision on the sensed threshold voltage. For an even victim cell, there are 1,024 (=4^5) combinations of $k_1,k_2,\cdots,k_M$, while the number of combinations is reduced to 64 (= 4^3) for an odd victim cell. Similar to Eq. (5.1) and (5.2), the LLRs for the LSB and the MSB pages can be computed as follows:

$$\Lambda_L(v_q) = \log \frac{\int_{q_i}^{q_{i-1}} f(v|11,k_1,\cdots,k_M) + \int_{q_i}^{q_{i-1}} f(v|01,k_1,\cdots,k_M)}{\int_{q_i}^{q_{i-1}} f(v|00,k_1,\cdots,k_M) + \int_{q_i}^{q_{i-1}} f(v|10,k_1,\cdots,k_M)} \tag{5.14}$$

and

$$\Lambda_L(v_q) = \log \frac{\int_{q_i}^{q_{i-1}} f(v|11,k_1,\cdots,k_M) + \int_{q_i}^{q_{i-1}} f(v|10,k_1,\cdots,k_M)}{\int_{q_i}^{q_{i-1}} f(v|01,k_1,\cdots,k_M) + \int_{q_i}^{q_{i-1}} f(v|00,k_1,\cdots,k_M)}. \tag{5.15}$$

Note that $f_{V_{TH}|\mathbf{X},\mathbf{Z_1},\cdots,\mathbf{Z_M}}(v|k_0,k_1,\cdots,k_M)$ is denoted as $f(v|k_0,k_1,k_2,\cdots,k_M)$ for simplicity in Eq. (5.14) and (5.15). When comparing Eq. (5.1) and (5.14), the latter utilizes more specified signal distributions, which reflect the observations on the neighboring cells, to compute the LLR.

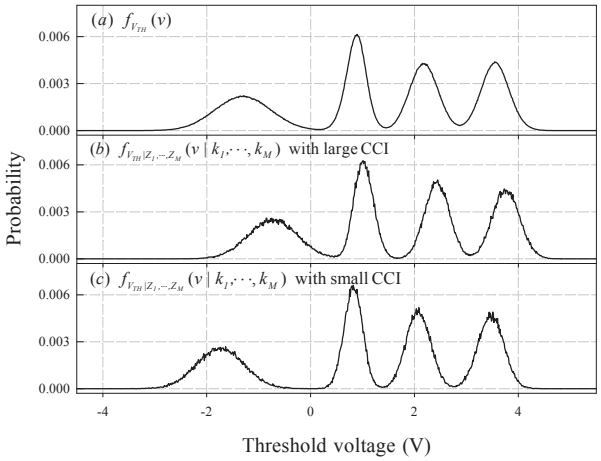


Figure 5.9: Examples of threshold voltage distributions depending on neighboring cells.

In Fig. 5.9, examples of $f_{V_{TH}|\mathbf{X}}(v|k)$ and $f(v|k_0,k_1,k_2,\cdots,k_M)$ are plotted. Note that Fig. 5.9-(b) is the case when the amount of CCI is much larger than that of Fig. 5.9-(c). By considering the memory sensing output of the neighboring cells, we can estimate more precise likelihood functions of V_{TH}.

Let us derive the mathematical formulation for $f(v|k_0,k_1,k_2,\cdots,k_M)$. Since the likelihood function of V_{TH} can be modeled as a Gaussian distribution, we need to find the conditional means and variances of V_{TH}. According to Eq. (2.7) in Chapter 2, V_{TH} is the sum of V_M, V_R, and V_{CCI}, which are the output of the MSB programming, the CCI, and the data retention noise, respectively. Thus, the conditional expectation of V_{TH} becomes

$$\begin{aligned} &E[V_{TH}|\mathbf{X}=k_0,\mathbf{Z_1}=k_1,\cdots,\mathbf{Z_M}=k_M] \\ &\quad = E[V_M|\mathbf{X}=k_0,\mathbf{Z_1}=k_1,\cdots,\mathbf{Z_M}=k_M]+E[V_R|\mathbf{X}=k_0,\mathbf{Z_1}=k_1,\cdots,\mathbf{Z_M}=k_M] \\ &\quad + E[V_{CCI}|\mathbf{X}=k_0,\mathbf{Z_1}=k_1,\cdots,\mathbf{Z_M}=k_M]. \end{aligned} \tag{5.16}$$

In Eq. (5.16), the amount of CCI, V_{CCI}, is uncorrelated with the input symbol of the victim cell $\mathbf{X}$. Also, V_M and V_R are independent with the neighboring symbols $\mathbf{Z_i}$. The data retention induced noise can be expressed via power law functions of the number of charges inside the floating gate [47, 23, 24], thus V_R is correlated with X. As a result, Eq. (5.16) can be simplified as follows:

$$\begin{aligned} &E[V_{TH}|\mathbf{X}=k_0,\mathbf{Z_1}=k_1,\cdots,\mathbf{Z_M}=k_M] \\ &\quad = E[V_M|\mathbf{X}=k_0]+E[V_R|\mathbf{X}=k_0]+E[V_{CCI}|\mathbf{Z_1}=k_1,\cdots,\mathbf{Z_M}=k_M]. \end{aligned} \tag{5.17}$$

According to the threshold voltage signal model in Chapter 2, the expectation of V_M is equal to

$$E[V_M|X=k_0] = V_{k_0}+\frac{1}{2}\Delta V_{pp}, \tag{5.18}$$

for $k_0 = 11$, 01, 00, and 10.

The conditional expectation of V_{CCI} can be obtained by using Eq. (5.5). Thus, we have

$$E[V_{CCI}|\mathbf{Z_1}=k_1,\cdots,\mathbf{Z_M}=k_M] = \sum_{i=1}^{M} C_{i-1}\cdot E\Big[\Delta V[i]|\mathbf{Z_i}=k_i\Big]. \tag{5.19}$$

In Eq. (5.19), we utilize the fact that the threshold voltage shift of the i-th cell is independent from those of the other neighboring cells.

Let us denote the sum of V_M and V_{CCI} as V_I. Note that V_I represents the initial threshold voltage. In order to find $E[V_R|\mathbf{X}=k_0]$, we use the relationship between V_{TH} and V_I, $V_R = V_{TH}-V_I$. Then, $E[V_R|\mathbf{X}=k_0]$ becomes

$$E[V_R|\mathbf{X}=k_0] = E[V_{TH}|\mathbf{X}=k_0]-E[V_I|\mathbf{X}=k_0]. \tag{5.20}$$

Note that $E[V_{TH}|\mathbf{X}=k_0]$ can be obtained by applying the threshold voltage distribution estimation algorithms and $E[V_I|\mathbf{X}=k_0]$ is known in advance. By using Eq. (5.18), (5.19), and (5.20), the conditional expectation of V_{TH} can be computed as follows:

$$\begin{aligned} &E[V_{TH}|\mathbf{X}=k_0,\mathbf{Z_1}=k_1,\cdots,\mathbf{Z_M}=k_M] \\ &\quad = V_{k_0}+\frac{1}{2}\Delta V_{pp}+\sum_{i=1}^{M} C_{i-1}\cdot E[\Delta V[i]|\mathbf{Z_i}=k_i]+E[V_{TH}|\mathbf{X}=k_0]-E[V_I|\mathbf{X}=k_0]. \end{aligned} \tag{5.21}$$

Let us find the conditional variance of V_{TH}. Similar to Eq. (5.17), the conditional variance can be decomposed into three terms as follws:

$$\begin{aligned} &Var[V_{TH}|\mathbf{X}=k_0,\mathbf{Z_1}=k_1,\cdots,\mathbf{Z_M}=k_M] \\ &\quad = Var[V_M|\mathbf{X}=k_0]+Var[V_R|\mathbf{X}=k_0]+Var[V_{CCI}|\mathbf{Z_1}=k_1,\cdots,\mathbf{Z_M}=k_M]. \end{aligned} \tag{5.22}$$

The likelihood function of V_M is a uniform distribution with the width of ΔV_{pp}, thus, its variance becomes

$$Var[V_M|\mathbf{X}=k_0]=\frac{\Delta V_{pp}^2}{12}. \tag{5.23}$$

Since $\Delta V[i]$ is *i.i.d*, the conditional variance of V_{CCI} can be found by using Eq. (5.6) as follows:

$$Var[V_{CCI}|\mathbf{Z_1}=k_1,\cdots,\mathbf{Z_M}=k_M]=\sum_{i=1}^{M} C_{i-1}^2\cdot Var\Big[\Delta V[i]|\mathbf{Z_i}=k_i\Big]. \tag{5.24}$$

The conditional variance of the data retention noise also can be found by using

the statistical information of V_I. Thus, the conditional variance of V_R becomes

$$\begin{aligned} Var[V_R|\mathbf{X}=k_0] &= E[V_R^2|\mathbf{X}=k_0] - E[V_R|\mathbf{X}=k_0]^2 \qquad (5.25)\\ &= E[V_{TH}^2|\mathbf{X}=k_0] - 2E[V_{TH}V_I|\mathbf{X}=k_0] + E[V_I^2|\mathbf{X}=k_0] - E[V_R|\mathbf{X}=k_0]^2 \\ &= Var[V_{TH}|\mathbf{X}=k_0] - Var[V_I|\mathbf{X}=k_0] + E[V_{TH}|\mathbf{X}=k_0]^2 - E[V_I|\mathbf{X}=k_0]^2 \\ &\quad - 2E[V_{TH}V_I|\mathbf{X}=k_0] - E[V_R|\mathbf{X}=k_0]^2. \end{aligned}$$

In Eq. (5.25), the likelihood functions of V_{TH} and V_I are known, thus $E[V_{TH}^2|\mathbf{X}=k_0]$ and $E[V_I^2|\mathbf{X}=k_0]$ can be obtained simply. Also, $E[V_R|\mathbf{X}=k_0]^2$ can be computed by using Eq. (5.20). The remained correlation term can be simplified as follows:

$$\begin{aligned} &E[V_{TH}V_I|\mathbf{X}=k_0] \\ &\quad= \int_{-\infty}^{\infty} E[V_{TH}V_I|\mathbf{X}=k_0, V_I=v] f_{V_I|\mathbf{X}}(v|k_0)dv \\ &\quad= \int_{-\infty}^{\infty} \left(E[V_I^2|\mathbf{X}=k_0, V_I=v] + E[V_IV_R|\mathbf{X}=k_0, V_I=v]\right) f_{V_I|\mathbf{X}}(v|k_0)dv \\ &\quad= \int_{-\infty}^{\infty} \left(v^2 + vE[V_R|\mathbf{X}=k_0, V_I=v]\right) f_{V_I|\mathbf{X}}(v|k_0)dv. \qquad (5.26) \end{aligned}$$

In order to further simplify (5.26), we replace $E[V_R|\mathbf{X}=k_0, V_I=v]$ with $E[V_R|\mathbf{X}=k_0]$. Recall that $V_I(=V_M+V_{CCI})$ is a function of $\mathbf{X}$, thus $E[V_R|\mathbf{X}=k_0, V_I=v]$ and $E[V_R|\mathbf{X}=k_0]$ have similar values. By applying this approximation, we have

$$\begin{aligned} E[V_{TH}V_I|\mathbf{X}=k_0] &= \int_{-\infty}^{\infty} \left(v^2 + vE[V_R|\mathbf{X}=k_0]\right) f_{V_I|\mathbf{X}}(v|k_0)dv \\ &= E[V_I^2|\mathbf{X}=k_0] + E[V_R|\mathbf{X}=k_0] \cdot E[V_I|\mathbf{X}=k_0]. \qquad (5.27) \end{aligned}$$

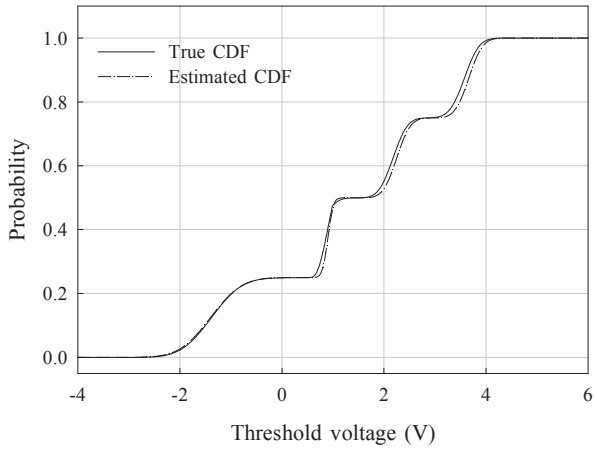


Figure 5.10: Cumulative density functions of $f_{V_{TH}|\mathbf{Z_1},\cdots,\mathbf{Z_M}}(v|k_1,\cdots,k_M)$ and its Gaussian approximation.

Thus, the conditional variance of V_R becomes

$$Var[V_R|\mathbf{X}=k_0] = E[V_{TH}^2|\mathbf{X}=k_0] - E[V_I^2|\mathbf{X}=k_0] \\ - 2E[V_R|\mathbf{X}=k_0]\cdot E[V_I|\mathbf{X}=k_0] - E[V_R|\mathbf{X}=k_0]^2. \tag{5.28}$$

The conditional PDF of V_{TH} can be modeled as a Gaussian function, and the mean and the variance can be computed from Eq. (5.17) and (5.22). Figure 5.10 shows the conditional CDFs of V_{TH} and its Gaussian approximation, where we can find that they are quite similar.

5.3.3 Experimental Results

We conducted experiments to evaluate the post-FEC BER performance of the proposed soft-information computation algorithms. During the experiments, the data samples obtained from the simulated two-bit MLC NAND flash memory model de-

scribed in Chapter 2 were used while changing the retention time. The (68254, 65536) EG-LDPC code with the min-sum decoding algorithm was applied for error correction. The 4-, 7-, and 10-level MMI quantizers are employed for the victim cells, while the 4-level uniform quantizer is used for the surrounding cells. Note that the 4-, 7-, and 10-level MMI quantizers demand 3, 6, and 9 memory sensing operations, respectively. To remove the CCI for the even (odd) victim cells, the five (three) neighboring cells need to be read additionally, which demands 9 (6) memory sensing operations. For the comparison purpose, the error performances of hard- and soft-decision decoding without applying CCI cancellation are also presented, which are denoted as 'LDPC' in Fig. 5.11 and 5.12.

Figure 5.11 shows the error performance of the even MSB pages. Since the BER performance of LSB pages is quite similar to that of MSB pages except that all the curves are shifted to the right (longer retention time), we do not present it here. From Fig. 5.11, we can find that the BER performance is substantially improved by using the proposed soft-information computation schemes, especially for the JCS one. Even for the 'JCS, 7-level' case, which demands 15 memory sensing operations, its tolerable retention time is more than five times longer than that of the 'LDPC, 16-level' case that requires the same number of voltage sensing operations. It is also noteworthy that the 'JCS, 4-level' case shows improved BER performance when compared to the 'LDPC, 4-level' case. For the sequential read requests, in which memory sensing overheads for the neighboring cells can be hidden, the soft-decision error correction with the 'JCS, 4-level' quantization scheme can replace hard-decision error correction. When comparing the JCS scheme and the approach that separately conducts CCI cancellation and soft-decision error correction, which is denoted as 'CCIC, LDPC' in Fig. 5.11 and 5.12, the JCR algorithm shows much improved BER performance. This

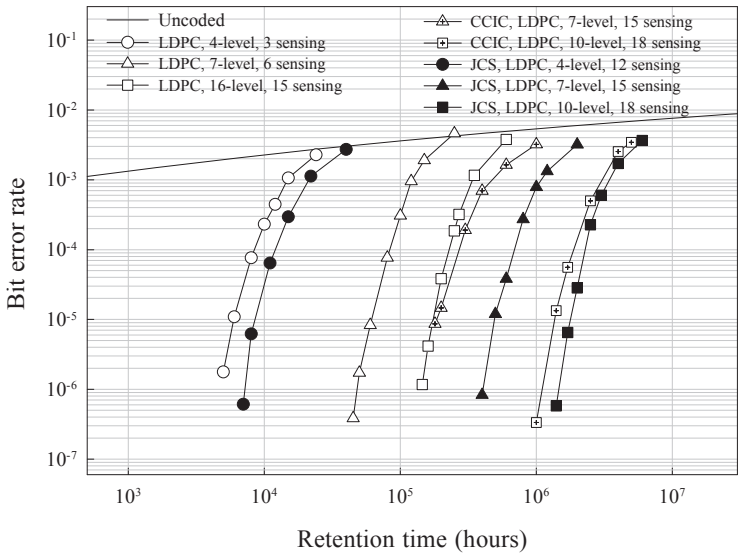


Figure 5.11: Error performance of a (68254,65536) EG-LDPC code for the even MSB pages.

means that the soft-information computed by the JCS scheme is more accurate than that of the counterpart. As the precision of memory sensing output increases, both approaches can yield reliable LLR values, thus the performance gap between the two approaches decreases.

Figure 5.12 shows the error performance for the odd MSB pages. We can find that removing the CCI does not improve the corrected BER performance when the precision of the sensed signal is too low. Actually, hard-decision error correction with the JCS scheme even degrades the BER performance when compared to the 'LDPC, 4-level case'. The odd pages are less severely affected by the CCI than the even ones, thus removing the CCI does not improve the signal quality much, especially when the quantization noises are large. As the precision of memory sensing output increases, however, we can obtain improved BER performances by considering the CCI when

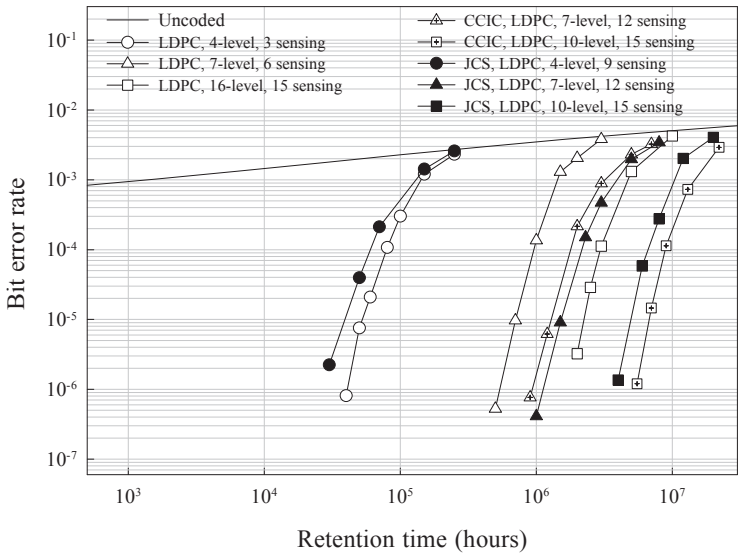


Figure 5.12: Error performance of a (68254,65536) EG-LDPC code for the odd MSB pages.

computing LLRs. For example, the 'CCIC, LDPC, 10-level' case can yield almost 300 % longer lifetime when compared to the 'LDPC, 16-level' case. Note that both cases require the same number of memory sensing operations. The odd pages show much longer lifetime than the even pages, thus improving the BER performance of even pages is more important to increase the overall retention time limit of NAND flash memory.

5.4 Concluding Remarks

We have developed two soft-information computation schemes to combine CCI cancellation and soft-decision error correction in MLC NAND flash memory. The first approach derives the PDF of the CCI removed signal and uses it to compute LLRs. The second one jointly conducts CCI cancellation and soft-input computation. The

developed algorithms are evaluated by using a simulated NAND flash memory model. In the experiments, we can obtain significant improvement on the worst case (even pages) BER performances even when only a limited number of memory sensing operations is used.

Chapter 6

Conclusion

In this dissertation, signal processing algorithms are proposed to improve the signal quality of sub-20 nm MLC (multi-level cell) NAND flash memory. First, we have developed threshold voltage distribution estimation algorithms to offer reliable statistical information, which are the means and the standard deviations, to signal processing and/or error correcting units. The sensing directed estimation (SDE) algorithm shows small estimation errors even when the threshold voltage distribution is shifted significantly, but demands memory sensing overheads. On the other hand, the decision directed estimation (DDE) algorithm requires no additional memory sensing operations, but can be conducted only when error corrected bit patterns are available. The combined approach that employs both the DDE and the SDE algorithms can minimize memory sensing overheads while maintaining the estimation accuracy.

We also provide detailed characterization of the cell-to-cell interference (CCI). Our experimental results reveal that the coupling coefficients vary depending on the physical locations of the victim cells but not on the number of program-erase (PE)

cycles. We also have developed a CCI cancellation algorithm that is similar to the interference cancellers employed in conventional communications systems. The proposed algorithm can remove the deterministic part of the CCI, thus the error correcting unit can focus on the random errors. The experimental results showed that the proposed CCI canceller significantly lowers the bit error rate (BER).

Finally, this dissertation presents soft-information computation schemes that combine the proposed signal processing algorithms and soft-decision error correction, which usually shows much improved error performance than the conventional hard-decision one. The proposed joint CCI cancellation and soft-information computation (JCS) scheme can improve the corrected BER significantly with only a small number of memory sensing operations.

Advanced process technology is no more sufficient for increasing the density of NAND flash memory because scaling also lowers the quality of threshold voltage signal. This study allows to design more reliable high-density NAND flash memory systems by means of signal processing and advanced error correction techniques.

Bibliography

[1] M. Wang, "Technology trends on 3D-NAND flash storage," in *Proceedings of Interational Microsystems, Packaging, Assembly, and Circuit Technology Conference (IMPACT'11)*, 2011.

[2] J.-D. Lee, S.-H. Hur, and J.-D. Choi, "Effects of floating-gate interference on NAND flash memory cell operation," *IEEE Electron Device Letter*, vol. 23, no. 5, pp. 264–266, May 2002.

[3] K. Prall, "Scaling non-volatile memory below 30nm," in *Proceedings of Non-volatile Semiconductor Memory Workshop (NVSMW'07)*, Aug. 2007, pp. 5–10.

[4] S. Gregori, A. Cabrini, O. Khouri, and G. Torelli, "On-chip error correcting techniques for new-generation flash memories," *Proceedings of IEEE*, vol. 91, no. 4, pp. 602–616, Apr. 2003.

[5] W. Liu, J. Rho, and W. Sung, "Low-power high-throughput BCH error correction VLSI design for multi-level cell NAND flash memories," in *Proceedings of IEEE Workshop on Signal Processing Systems (SiPS'06)*, Oct. 2006, pp. 303–308.

[6] R. Micheloni *et al.*, "A 4Gb 2b/cell NAND flash memory with embedded 5b BCH ECC for 36MB/s system read throughput," in *Proceedings of IEEE International Solid-State Circuit Conference (ISSCC'06)*, Feb. 2006, pp. 497–506.

[7] F. Sun, S. Devarajan, K. Rose, and T. Zhang, "Design of on-chip error correction systems for multilevel NOR and NAND flash memories," *IET Circuits, Devices and Systems*, vol. 1, no. 3, pp. 241–249, June 2007.

[8] B. Chen, X. Zhang, and Z. Wang, "Error correction for multi-level NAND flash memory using Reed-Solomon codes," in *Proceedings of IEEE Workshop on Signal Processing Systems (SiPS'08)*, 2008, pp. 94–99.

[9] G. Dong, N. Xie, and T. Zhang, "On the use of soft-decision error-correction codes in NAND flash memory," *IEEE Transactions on Circuits and Systems I, Regular Papers*, vol. 58, no. 2, pp. 429–439, Feb. 2011.

[10] C. Yang, Y. Emre, and C. Chakrabarti, "Product code schemes for error correction in MLC NAND flash memories," *IEEE Transactions on VLSI Systems*, vol. 20, no. 12, pp. 2302–2314, Dec. 2012.

[11] J. Kim, D. Lee, and W. Sung, "Performance of rate 0.96 (68254, 65536) EG-LDPC code for NAND Flash memory error correction," in *Proceedings of IEEE International Conference on Communications (ICC'12), Workshop Emerging Data Storage Technology*, June 2012, pp. 7029–7033.

[12] D. Lee and W. Sung, "Least squares based cell-to-cell interference cancelation technique for multi-level cell NAND flash memory," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'12)*, 2012, pp. 1601–1604.

[13] ——, "Estimation of NAND flash memory threshold voltage distribution for optimum soft-decision error correction," *IEEE Transactions on Signal Processing*, vol. 61, no. 2, pp. 440–449, Jan. 2013.

[14] ——, "Least squares based coupling cancelation for MLC NAND flash memory with a small number of voltage sensing operations," *Journal of Signal Processing Systems*, vol. 71, no. 3, pp. 189–200, 2013.

[15] ——, "Decision directed estimation of threshold voltage distribution in NAND flash memory," *IEEE Transactions on Signal Processing*, accepted for publication.

[16] ——, "Soft-decision decoding with cell-to-cell interference removed signal in NAND flash memory," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'13)*, 2013, pp. 318–323.

[17] D. Lee, J. Kim, and W. Sung, "Signal processing techniques for reliability improvement of sub-20 nm NAND flash memory," in *Proceedings of IEEE Workshop on Signal Processing Systems (SiPS'13)*, 2013, pp. 318–323.

[18] K. Takeuchi, Y. Kameda, S. Fujimura, H. Otake, K. Hosono, H. Shiga, Y. Watanabe, T. Futatsuyama, Y. Shindo, M. Kojima, *et al.*, "A 56-nm CMOS 99-mm^2 8-Gb multi-level NAND flash memory with 10-MB/s program throughput," *IEEE Journals of Solid-State Circuits*, vol. 42, no. 1, pp. 219–232, Jan. 2007.

[19] K. Park, M. Kang, D. Kim, S. Hwang, B. Choi, Y. Lee, C. Kim, and K. Kim, "A zeroing cell-to-cell interference page architecture with temporary LSB storing and parallel MSB program scheme for MLC NAND flash memories," *IEEE Journals of Solid-State Circuits*, vol. 43, no. 4, pp. 919–928, Apr. 2008.

[20] Y. Li, S. Lee, Y. Fong, F. Pan, T. Kuo, J. Park, T. Samaddar, H. Nguyen, M. Mui, K. Htoo, *et al.*, "A 16Gb 3b/cell NAND flash memory in 56nm with 8MB/s write rate," in *Proceedings of IEEE International Solid-State Circuits Conference (ISSCC'08)*, 2008, pp. 506–632.

[21] R. Cernea, L. Pham, F. Moogat, S. Chan, B. Le, Y. Li, S. Tsao, T. Tseng, K. Nguyen, J. Li, *et al.*, "A 34 MB/s MLC write throughput 16 Gb NAND with all bit line architecture on 56 nm technology," *IEEE Journals of Solid-State Circuits*, vol. 44, no. 1, pp. 186–194, Jan. 2009.

[22] K.-D. Suh *et al.*, "A 3.3 V 32 Mb NAND flash memory with incremental step pulse programming scheme," *IEEE Journals of Solid-State Circuits*, vol. 30, no. 11, pp. 1149–1156, Nov. 1995.

[23] J. Lee, J. Choi, D. Park, and K. Kim, "Effects of interface trap generation and annihilation on the data retention characteristics of flash memory cells," *IEEE Transactions on Device and Materials Reliability*, vol. 4, no. 1, pp. 110–117, Jan. 2004.

[24] C. Miccoli, M. Compagnoni, S. Beltrami, A. Spinelli, and A. Visconti, "Threshold-voltage instability due to damage recovery in nanoscale NAND Flash memories," *IEEE Transactions on Electron Devices*, vol. 58, no. 8, pp. 2406–2414, Aug. 2011.

[25] M. Wilk and R. Gnanadesikan, "Probability plotting methods for the analysis of data," *Biometrika*, vol. 55, no. 1, pp. 1–17, Jan. 1968.

[26] K. Takeuchi, T. Tanaka, and H. Nakamura, "A double-level-Vth select gate array architecture for multilevel NAND flash memories," *IEEE Journals of Solid-State Circuits*, vol. 31, no. 4, pp. 602–609, Apr. 1996.

[27] G. Dong, Y. Pan, N. Xie, C. Varanasi, and T. Zhang, "Estimating information-theoretical NAND flash memory storage capacity and its implication to memory system design space exploration," *IEEE Transactions on VLSI Systems*, vol. 20, no. 9, pp. 1705–1714, Sept. 2012.

[28] Y. Cai, G. Yalcin, O. Mutlu, E. Haratsch, A. Cristal, O. Unsal, and K. Mai, "Flash correct-and-refresh: Retention-aware error management for increased flash memory lifetime," in *Proceedings of IEEE International Conference on Computer Design (ICCD'12)*, 2012, pp. 94–101.

[29] C. Lee, S. Lee, S. Ahn, J. Lee, W. Park, Y. Cho, C. Jang, C. Yang, S. Chung, I. Yun, *et al.*, "A 32-Gb MLC NAND Flash memory with Vth endurance enhancing schemes in 32 nm CMOS," *IEEE Journals of Solid-State Circuits*, vol. 46, no. 1, pp. 97–106, Jan. 2011.

[30] J. Wang, T. Courtade, H. Shankar, and R. Wesel, "Soft information for LDPC decoding in flash: Mutual-information optimized quantization," in *Proceedings of IEEE Global Communication Conference (GLOBECOM'11)*, Dec. 2011, pp. 1–6.

[31] *NAND Flash Memory Data Book*, SK Hynix Semiconductor, Icheon, Korea, 2012.

[32] S. Cha, "Taxonomy of nominal type histogram distance measures," in *Proceedings of the American Conference on Applied Mathematics*, 2008, pp. 325–330.

[33] D. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *Journal of the society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431–441, Feb. 1963.

[34] J. More, "The Levenberg-Marquardt algorithm: Implementation and theory," *Numerical analysis*, pp. 105–116, 1978.

[35] B. Flannery, W. Press, S. Teukolsky, and W. Vetterling, *Numerical recipes in C*. Press Syndicate of the University of Cambridge, New York, 1992.

[36] G. Dong, S. Li, and T. Zhang, "Using data postcompensation and predistortion to tolerate cell-to-cell interference in MLC NAND flash memory," *IEEE Transactions on Circuits and System I, Regular Papers*, vol. 57, no. 10, pp. 2718–2728, Oct. 2010.

[37] *64Gb, 128Gb, 256Gb, 512Gb Asynchronous/Synchronous NAND Features*, Micron Technology Inc., Boise, ID, 2009. [Online]. Available: http://www.micron.com/products/nand-flash/mlc-nand

[38] S. Tanaka, M. Sawahashi, and F. Adachi, "Pilot symbol-assisted decision-directed coherent adaptive array diversity for DS-CDMA mobile radio reverse link," *IEICE Transactions on Fundamentals of Electronics, Communications, and Computer Science*, vol. 80, no. 12, pp. 2445–2454, Dec. 1997.

[39] K. Shi, E. Serpedin, and P. Ciblat, "Decision-directed fine synchronization in OFDM systems," *IEEE Transactions on Communications*, vol. 53, no. 3, pp. 408–412, Mar. 2005.

[40] J. Kim and W. Sung, "Low-energy error correction of NAND Flash memory through soft-decision decoding," *EURASIP Journal on Advances in Signal Processing*, vol. 2012, no. 1, pp. 1–12, Sept. 2012.

[41] Y. Cai, E. Haratsch, O. Mutlu, and K. Mai, "Threshold voltage distribution in MLC NAND flash memory: Characterization, analysis, and modeling," in *Proceedings of the Conference on Design, Automation and Test in Europe (DATE'13)*, 2013, pp. 1285–1290.

[42] D. Park and J. Lee, "Floating-gate coupling canceller for multi-level cell NAND flash," *IEEE Transactions on Magnetics*, vol. 47, no. 3, pp. 624–628, Mar. 2011.

[43] M. Jeon, K. Kim, B. Shin, and J. Lee, "Interference compensation technique for multilevel flash memory," in *Proceedings of IEEE International Midwest Symposium on Circuits and Systems (MWSCAS'11)*, 2011, pp. 1–4.

[44] M. Rhee, S. Lee, and S. Yoon, "Implementing a NAND controller for ONFI NAND flash memory," in *Proceedings of KIISE Korea Computer Congress (KCC'12)*, 2012.

[45] P. Poliakov, P. Blomme, M. M. Corbalan, J. V. Houdt, and W. Dehaene, "Cross-cell interference variability aware model of fully planar NAND Flash memory including line edge roughness," *Microelectronics Reliability*, vol. 51, no. 5, pp. 919–924, May 2011.

[46] T. M. Cover and J. A. Thomas, *Elements of information theory*. New York: John Wiley, 1991.

[47] J. Lee, J. Choi, D. Park, and K. Kim, “Degradation of tunnel oxide by FN current stress and its effects on data retention characteristics of 90 nm NAND flash memory cells,” in *Proceedings of IEEE International Reliability Physics Symposium (IRPS’03)*, 2003, pp. 497–501.

국문 초록

낸드 플래시 메모리는 빠른 읽기/쓰기 동작과 낮은 소비 전력, 충격에 강한 특성 등의 장점으로 인하여 휴대 전화, USB 메모리, 디지털 카메라 등의 다양한 전자 장치에서 광범위하게 사용되고 있다. 낸드 플래시 메모리의 대용량화는 한 셀에 여러 비트를 저장하는 멀티 레벨 셀(multi-level cell) 방식과 공정 기술의 미세화를 통하여 꾸준하게 이루어져왔다. 그러나 메모리의 집적도가 높아질수록, 셀의 문턱 전압(threshold voltage) 신호는 각종 잡음 및 인접 셀의 간섭(cell-to-cell interference)에 취약해지고 발생되는 비트 에러의 수도 증가한다. 현재까지는 낸드 플래시 메모리의 신뢰도를 향상시키기 위하여 BCH (Bose-Chaudhuri-Hocquenghem) 부호 또는 RS (Reed-Solomon) 부호 기반의 경판정 에러 정정(hard-decision error correction)이 널리 이용되고 있다. 그러나 20 nm 이하 공정의 낸드 플래시 메모리에서는 경판정 에러 정정만으로는 감당할 수 없는 정도로 비트 에러의 수가 급격하게 증가한다. 본 논문에서는 20 nm 이하 공정을 사용하는 고밀도 낸드 플래시 메모리의 신뢰도 향상을 위한 신호처리 알고리즘과 연판정 에러 정정(soft-decision error correction) 방법을 제안한다.

본 논문의 첫 번째 부분에서는 문턱 전압 분포 추정 알고리즘을 제안한다. 낸드 플래시 메모리에서는 시간이 지남에 따라 플로팅 게이트(floating gate)에 저장된 전하가 누설되는 현상이 발생하는데, 이는 비트 에러의 주요한 원인이 된다. 본 논문에서 제안된 문턱 전압 추정 방식은 크게 둘로 구분된다. 먼저, SDE (sensing directed estimation) 방식은 메모리를 비교적 높은 정밀도로 읽고 이 과정에서 측정된 데이터를 이용하여 문턱 전압 분포를 추정하는 방식이다.

이에 반하여, DDE (decision directed estimation) 방식에서는 에러 정정에 성공한 비트 패턴을 이용하여 추가의 메모리 센싱 없이 전압 분포를 추정한다. 본 논문에서는 실제 및 가상의 낸드 플래시 메모리에서 얻은 셀 데이터를 이용하여 제안된 알고리즘이 문턱 전압 분포를 매우 정확하게 추정할 수 있음을 보였다.

본 논문의 두 번째 부분에서는 셀 간 간섭 제거 알고리즘을 제안한다. 본 연구에서는 먼저 실제 낸드 플래시 메모리 칩에서 측정된 셀 간 간섭 계수의 통계적인 특성을 제시하고, 이를 바탕으로 기존의 통신 시스템에서 널리 사용되는 간섭 제거기와 유사한 형태의 셀 간 간섭 제거 알고리즘을 제안하였다. 제안된 알고리즘에서는 셀 간 간섭 계수를 신호처리 방법을 이용하여 추정하고, 비교적 간단한 연산을 통하여 셀 간 간섭을 제거한다. 이에 더하여, 본 논문에서는 셀 간 간섭 제거에 최적화된 양자화 방법에 대한 연구도 추가적으로 수행하였다.

마지막으로 본 논문에서는 연판정 에러 정정을 낸드 플래시 메모리에 적용하기 위한 신호처리 알고리즘에 대한 연구를 수행하였다. 연판정 에러 정정을 위해서는 복호화기의 입력으로 신뢰도 정보가 요구된다. 본 연구에서는 셀 간 간섭을 고려한 신뢰도 정보의 계산 방식을 제안하였다. 첫 번째 방법은 셀 간 간섭 제거 알고리즘과 연판정 에러 정정을 개별적으로 수행하는 것으로, 셀 간 간섭이 제거된 신호의 확률 분포를 추정하여 신뢰도 정보를 계산한다. 두 번째 방식에서는 주변 셀의 정보가 고려된 전압 분포의 수식을 유도하고, 이를 이용하여 셀 간 간섭 제거와 연판정 에러 정정을 동시에 수행한다. 본 연구를 통해 제안된 문턱 전압 분포 추정, 셀 간 간섭 제거, 연판정 에러 정정 방법을 통하여 낸드 플래시 메모리의 신뢰도를 비약적으로 향상시킬 수 있다.

주요어 : 낸드 플래시 메모리, 메모리 신호처리, 문턱 전압 분포 추정, 셀 간 간섭 제거, 연판정 오류 정정

학번 : 2009-20856