Direct and Indirect Measurement of Inter-Cell Capacitance in NAND Flash Memory

Dong-hwan Lee and Wonyong Sung
Department of Electrical and Computer Engineering
Seoul National University
599 Gwanangno, Gwanak-gu, Seoul, 151-744 Korea
Email: ldh@dsp.snu.ac.kr; wysung@snu.ac.kr

***Abstract*—As the density of NAND flash memory grows, the cell-to-cell interference caused by capacitive coupling among neighboring cells becomes a critical source of bit errors. Thus, it is important to precisely measure the value of capacitances to remove the interferences and lower the bit-error rate. Previous approaches have employed the least mean square (LMS) or the least square adaptive filtering approaches to remove the interference and thereby indirectly assess the capacitance values. In this paper, we measure the capacitance values directly by using specific cell programming patterns. It is found that the capacitance values do not change according to the PE (program erase) cycles, thus the measurement can be conducted only once for a fresh chip. We show the comparison results between the direct and the least squares based methods. The direct method not only provides the statistical distribution of the capacitance values but also shows more accurate estimation when the interference is very severe. The indirect methods that employ adaptive filtering have the advantage of using existing data instead of writing specific patterns.**

Keywords*—Cell-to-cell interference, NAND flash memory, least squares, LMS adaptive filter.

I. Introduction

High-density NAND flash memory devices are very important components for many applications from mobile devices to solid state drives (SSDs) in personal computers and servers. 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. During the last 10 years, the capacity of NAND flash memory has increased nearly 1,000 times by aggressive process scaling and multilevel cell (MLC) data coding technology. In MLC technology, two or more bits are stored in a cell to lower the cost per bit, which results in a much increased bit error rate. Thus, error correction techniques are usually employed to NAND flash memory based storage systems. The error sources include cell-to-cell interference (CCI), data retention, and excessive amount of program-erase cycles. Among them, the CCI is the most significant one in many cases [1], [2].

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 not only the target but also the surrounding cells increase [3]. In MLC flash memory, the least significant bit (LSB) programming is followed by the most significant bit (MSB) writing as illustrated in Fig. 1 in order to reduce the amount of CCI. During the MSB programming of the target cell, the incremental step pulse programming (ISPP) technique neutralizes the CCI induced by the previously programmed neighboring cells [4]. 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 located at the next word-line, which is the $(m+1)$-th word-line in Fig. 2, 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 [5]. The amount of interference that the (m,n)-th victim cells receives, where m and n denotes the word- and bit-line indices, respectively, can be represented as the linear combinations of the threshold voltage shifts of neighboring cells. Thus, the CCI for the even victim cells become

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

where $\Delta V[m,n-1]$ is the threshold voltage shift of the left neighboring cell for the victim cell, and so on. The coefficients, C_x, C_y, and C_{xy}, are the coupling ratio, and they

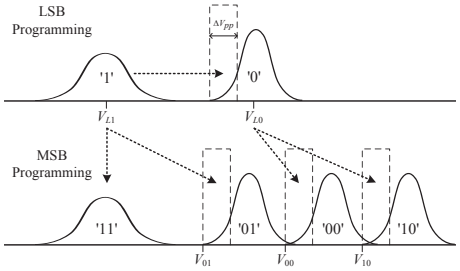

Fig. 1. Multi-page programming scheme.

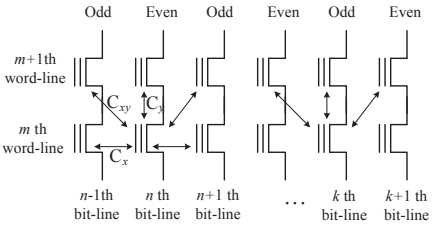

Fig. 2. Cell-to-cell interference model in the even/odd bit-line structure.

are determined by the geometry and the dielectric constant of the material for spacing.

There have been several researches to mitigate the effects of cell-to-cell interferences. One is modifying the memory structure [5] and the programming scheme [6], and the other is using signal processing techniques [7], [8]. The signal processing approach has shown good performances in removing the interference. However, the accuracy of measurement needs to be more precise as the amount interference grows resulting from chip density increase.

In this research, we directly measure the capacitance values among the cells by using specific cell programming patterns. Apparently this approach demands extra operations for cell programming when compared to the signal processing approach that utilizes user data samples. However, this approach can isolate the interference at each measurement and provide more reliable capacitance information.

This paper is organized as follows. Section II explains the NAND flash memory channel model and discusses the least squares based CCI canceller. In Section III, the direct measurement method is explained. The experimental results are shown in Section IV. Finally, concluding remarks are given in Section V.

II. Least Squares Method based Coupling Coefficient Estimation

In this section, we briefly explain the least squares (LS) method based coupling coefficient estimation algorithm [9]. This algorithm utilizes ordinary data patterns for indirect estimation of the coupling coefficients. In the LS approach, the coupling coefficients of the x, y, and xy directions can be obtained by measuring the threshold voltages of cells in the current and neighboring pages only one time, which reduces the memory sensing overheads.

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

$$V_{CCI}[n] = \sum_{i=0}^{M-1} C_i \cdot \Delta V[i], \tag{2}$$

where the number of interfering cells M is either five or three for the even or odd page. In Eq. (2), $\Delta V[i]$ denotes the threshold voltage shift of the i-th neighboring cell during the most significant bit (MSB) programming and is equal to $V_{TH}[i] - V_L[i]$ as shown in Fig. 1. 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. (2) becomes

$$\hat{V}_{CCI}[n] = \sum_{i=0}^{M-1} C_i \cdot \Big\{ V_{TH}[i] - E\big[V_L[i]|X_L[i]\big] \Big\}, \tag{3}$$

where $X_L[i]$ represents the pre-determined least significant bit (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. 1. We assume that V_{L1} and V_{L0} are known values. 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:

$$V_{CCI}[n] = V_{TH}[n] - V_M[n], \tag{4}$$

where $V_M[n]$ is the threshold voltage of the n-th victim cell before being affected by the CCI. Since it is not straightforward to directly measure $V_{CCI}[n]$, the LS algorithm uses $E[V_M|X[n]]$ instead of $V_M[n]$ as follows:

$$\bar{V}_{CCI}[n] = V_{TH}[n] - E[V_M|X[n]], \tag{5}$$

where $X[n]$ denotes the pre-determined symbol of the n-th victim cell. Computing $E[V_M|X[n]]$ is simple because the 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 $\hat{V}_{CCI}[n]$ in Eq. (3) and $\bar{V}_{CCI}[n]$ in Eq. (5) are estimators of $V_{CCI}[n]$, their difference can be used to define a cost function:

$$J = \sum_{n=0}^{N_s-1} \Big(\hat{V}_{CCI}[n] - \bar{V}_{CCI}[n] \Big)^2. \tag{6}$$

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 LS approach. Since we can use whole pages of data to indirectly measure the coupling ratios, the number of data to use, N_s, can be nearly unlimited.

In order to find the solution for Eq. (6), let us define a vector as

$$\mathbf{a}[n] = \Big[\Delta V_0[n], \Delta V_1[n], \cdots, \Delta V_{M-1}[n] \Big]^T, \tag{7}$$

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. (3) as

$$\hat{V}_{CCI}[n] = \mathbf{a}[n]^T \cdot \mathbf{x}, \tag{8}$$

where

$$\mathbf{x} = \Big[C_0,\ C_1,\ \cdots,\ C_{M-1} \Big]^T. \tag{9}$$

By using the above definition, Eq. (6) can be transformed into a matrix-vector form:

$$J_m(\mathbf{x}) = (\mathbf{A} \cdot \mathbf{x} - \mathbf{b})^T \cdot (\mathbf{A} \cdot \mathbf{x} - \mathbf{b}), \tag{10}$$

where

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}[0]^T \\ \mathbf{a}[1]^T \\ \vdots \\ \mathbf{a}[N_s-1]^T \end{bmatrix} \text{ and } \mathbf{b}_m = \begin{bmatrix} \bar{V}_{CCI}[0] \\ \bar{V}_{CCI}[1] \\ \vdots \\ \bar{V}_{CCI}[N_s-1] \end{bmatrix}. \tag{11}$$

Note that $\mathbf{A}$ is an N_s by M matrix and $\mathbf{b}$ is an N_s dimensional vector. Equation (10) is known as the least squares problem, and many algorithms have been developed to find $\mathbf{x}$ that

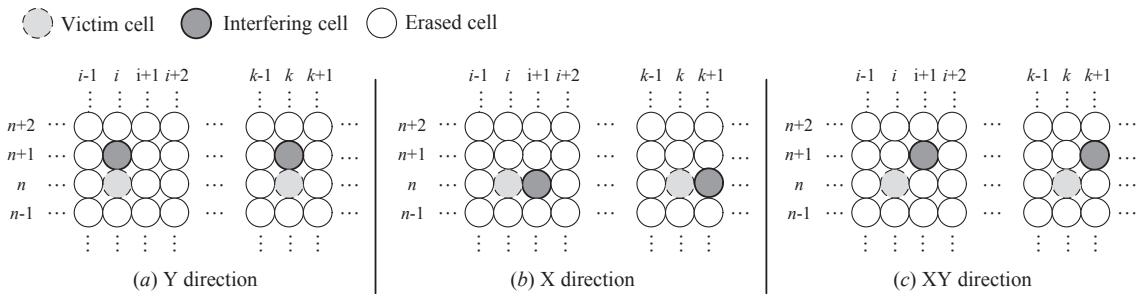

Fig. 3. Programming patterns to measure the coupling coefficients.

minimizes the cost function. Since Eq. (10) is a linear system, the analytic solution can be derived as

$$\mathbf{x}^* = (\mathbf{A}^T \cdot \mathbf{A})^{-1} \cdot \mathbf{A}^T \cdot \mathbf{b}. \tag{12}$$

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 program-erase (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 $O(N_s M^2)$.

III. Direct Measurement of Coupling Coefficients

In the direct approach, the coupling coefficient of each direction is measured by using one of the programming patterns shown in Fig. 3. 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. 3, 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. 3-(a), only the upper neighboring cell of a victim cell is programmed. Thus, $\Delta V[m+1, n]$ in Eq. (1) has a non-zero value, from which we can directly compute C_y. Similarly, the programming patterns shown in Fig. 3-(b) and 3-(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. 3 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. 3 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 [10] 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,

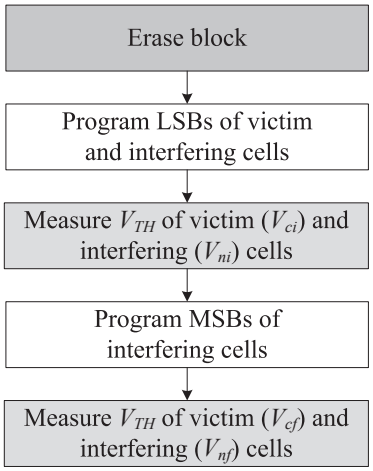

Fig. 4. Procedure to measure the coupling coefficients.

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 shows the procedure to measure the coupling coefficients from raw 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 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 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

$$\Delta V_c = V_{cf} - V_{ci} \tag{13}$$

and

$$\Delta V_n = V_{nf} - V_{ni}, \tag{14}$$

respectively. Note that ΔV_c and ΔV_n correspond to V_{CCI} and $\Delta V[i, j]$ in Eq. (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

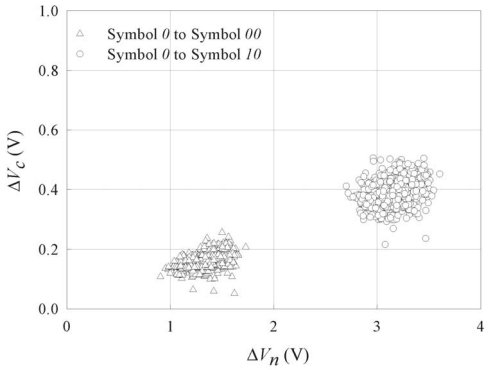

Fig. 5. Threshold voltage shifts of the victim (y-axis) and the interfering (x-axis) cells when using the programming patterns that are designed to measure C_y.

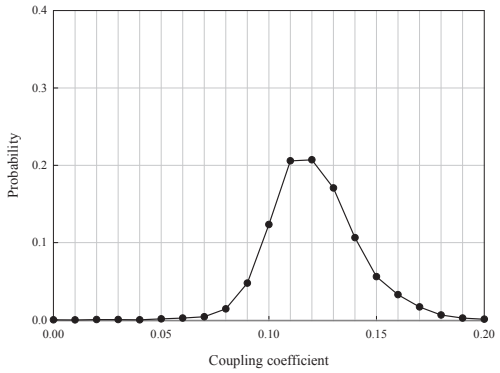

Fig. 6. Measured probability density function of C_y.

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 simply be computed as follows:

$$C = \frac{\Delta V_c}{\Delta V_n}. \tag{15}$$

IV. Experimental Results

A. Direct Measurement of Coupling Coefficients

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 5 shows the threshold voltage shifts of the victim and the interfering cells when the programming patterns shown in Fig. 3-(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. 5 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], [3], [11].

By using Eq. (15) and the data samples shown in Fig. 5, we can compute C_y and plot its probability density function

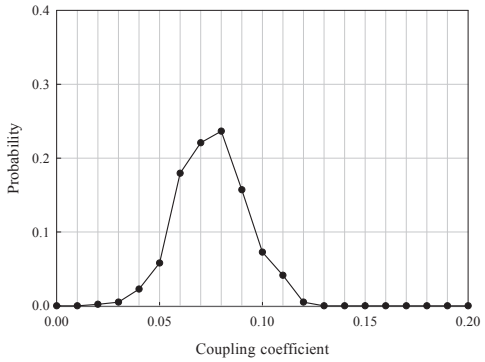

Fig. 7. Measured probability density function of C_x.

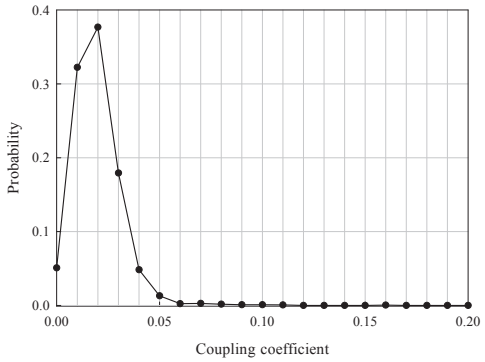

Fig. 8. Measured probability density function of C_{xy}.

(PDF) as depicted in Fig. 6. Similar to the results in [2], [11], C_y varies depending on the physical locations of the victim cells 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 [2]. 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.

As shown in Fig. 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, and they are smaller than those of C_y. Note that similar results were reported in [1], [2].

Figure 8 shows the measured PDF of C_{xy} when using the programming patterns shown in Fig. 3-(c). The experimental results show that C_{xy} is much smaller than C_y or C_x. Note that the mean and the standard 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 PE cycling to coupling coefficients, we measure C_y while increasing the number of PE cycles from 0 K to 5 K. We can observe that the mean of C_y remains almost the same as shown in Fig. 9. This experimental result demonstrates that PE cycling does not alter the coupling coefficients. Figure 10 shows the standard deviation values of C_y when increasing the number of PE cycles from 0 K to 5

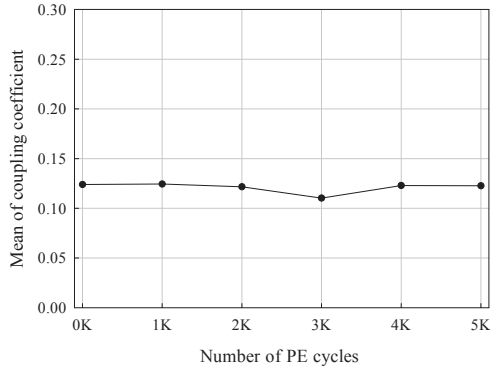

Fig. 9. Mean values of C_y when increasing the number of PE cycles from 0 K to 5 K.

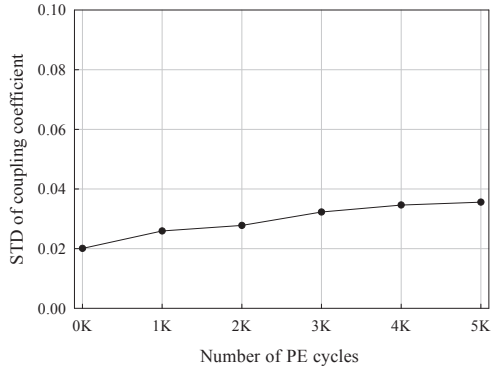

Fig. 10. Standard deviation values of C_y when increasing the number of PE cycles from 0 K to 5 K.

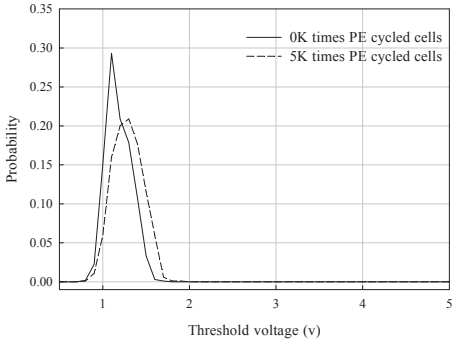

Fig. 11. Threshold voltage distributions when only the LSB of victim cell is programmed to the symbol *0*.

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 11 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 random telegraph noise (RTN) 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

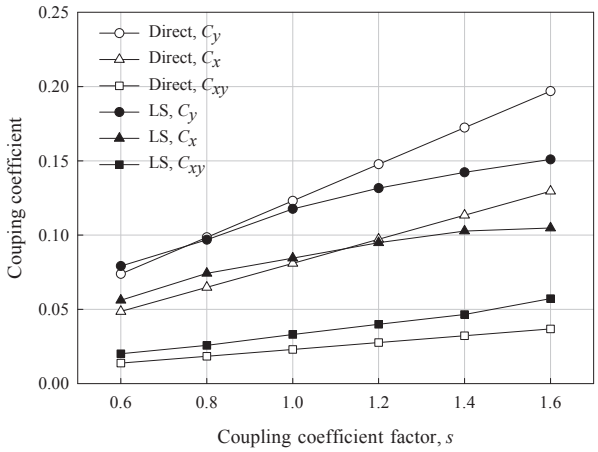

Fig. 12. Mean values of the coupling coefficients that are obtained by employing the direct ('Direct') and the least squares ('LS') based approaches.

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.

B. Comparison of Direct and Indirect Measurements

In order to assess the accuracy of coupling coefficients estimated by the LS approach, we apply both methods to the simulated NAND flash memory model [11] and compare the coupling coefficients as shown in Fig. 12. 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. 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. (3) and (4). When the CCI is extremely severe, those pre-determined symbols become erroneous, which degrades the estimation accuracy much.

C. BER Performance with CCI Cancellation

Figure 13 shows the bit error rate (BER) performances for the even victim cells when applying the CCI cancellation algorithm to the simulated NAND flash memory model. We apply both the direct (denoted as 'Direct' in Fig. 13) and least squares method (denoted as 'LS' in Fig. 13) based approaches to obtain the coupling coefficients. Note that the numbers inside the parenthesis represent quantization levels for the victim

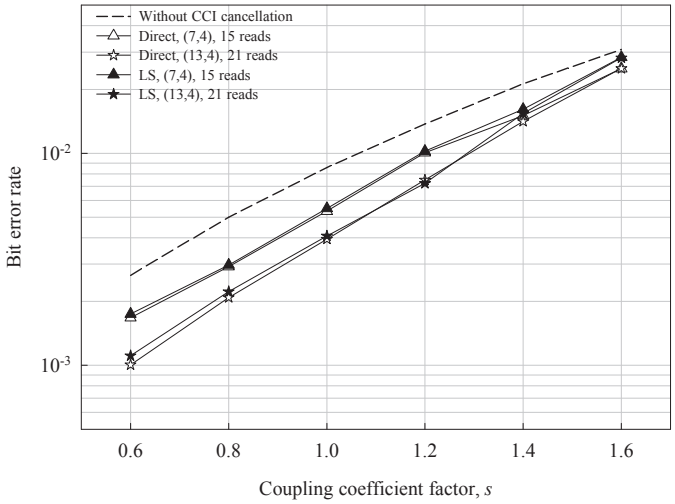

Fig. 13. BERs of the proposed CCI cancellation algorithm when applied to even pages of the simulated memory model.

and the neighboring cells, respectively. For example, '(7,4), 15 reads' represents the case when the 7-level maximizing mutual information (MMI) [12] 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. 13, we can find that the 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 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. 3, 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 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 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 various quantization schemes.

V. Concluding Remarks

In this paper, we propose a direct method for measuring the coupling ratios among the cells in NAND flash memory, and compare the technique with the least squares adaptive filtering approach. The proposed approach shows more accurate results, but demands specific programming patterns. Since the coupling ratios are independent of the program-erase cycles, it is possible to measure the coupling ratios when the memory systems are initialized for the first time. For both direct and indirect methods, the cell-to-cell interference cancelling performance depends on the sensing accuracy. Thus, it is needed to reduce the amount of sensing while keeping the interference cancelling performance.

References

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

[2] 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.

[3] 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.

[4] K.-D. Suh, B.-H. Suh, Y.-H. Um, J.-K. Kim, Y.-J. Choi, Y.-N. Koh, S.-S. Lee, S.-C. Kwon, B.-S. Choi, J.-S. Yum, J.-H. Choi, J.-R. Kim, and H.-K. Lim, "A 3.3 V 32 Mb NAND flash memory with incremental step pulse programming scheme," in *Proceedings of IEEE International Solid-State Circuits Conference (ISSCC'95)*, Feb 1995, pp. 128–129.

[5] 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.

[6] K.-T. Park, M. Kang, D. Kim, S.-W. Hwang, B.-Y. Choi, Y.-T. 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 Journal of Solid-State Circuits*, vol. 43, no. 4, pp. 919–928, April 2008.

[7] D. Lee and W. Sung, "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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.