

Statistical Characterization of Noise and Interference in NAND Flash Memory

Jaekyun Moon, *Fellow, IEEE*, Jaehyeong No, *Student Member, IEEE*, Sangchul Lee, Sangsik Kim, Seokhwan Choi, and Yunheub Song

Abstract—Given the limited set of empirical input/output data from flash memory cells, we describe a technique to statistically analyze different sources that cause the mean-shifts and random fluctuations in the read values of the cells. In particular, for a given victim cell, we are able to quantify the amount of interference coming from any arbitrarily chosen set of potentially influencing cells. The effect of noise and interference on the victim cell after repeated program/erase cycles as well as baking is also investigated. The results presented here can be used to construct a channel model with data-dependent noise and interference characteristics, which in turn can be utilized in designing and evaluating advanced coding and signal processing methods for flash memory.

Index Terms—Interference, NAND flash memory, statistical characterization.

I. INTRODUCTION

CHARACTERIZATION of the flash memory cell's input-output behavior is of great interest in the flash memory industry. As flash memory cell density increases, the effect of interference among neighboring cells also increases, causing shifts in as well as variations around the intended written cell values (threshold voltages) [1], [2]. For example, due to floating-gate voltage coupling, the threshold voltage level of an already-programmed cell ("victim" cell) may increase as extra charges are injected into an adjacent interfering cell [1]. Isolating the source of interference and noise would provide valuable insights in the flash memory manufacturing process and in developing a reliable channel model necessary in the design of advanced signal processing schemes. See, for example, the pre- and post-compensation techniques described in [3] based on a cell-to-cell interference model. Also, increasingly strong error correcting codes are being considered for

multi-level-cell (MLC) flash memory to mitigate errors due to interference and noise [4]–[6].

In this paper, we discuss a statistical analysis method applied to a set of experimental read data corresponding to a block of commercial flash memory cells, with a goal in mind to isolate and quantify the sources of the mean-shifts and variations observed on the read values of the memory cells. The materials presented in this paper were discussed in part during our recent conference presentations [7]–[9]. This paper creates a coherent theme by building on isolated results reported at these conferences. The present paper also provides additional analysis and methods that have not been previously presented. Additional sets of data have also been used in this paper to strengthen the validity of the model and conclusions of the earlier conference papers.

Our approach starts with specifying a mask shape that captures a certain number of cells, typically including the victim cell and potentially interfering cells nearby. Assuming a particular set of input (write) values for the cells positioned within the mask, the 2-dimensional (2-D) read data array is scanned over with the mask, and the read (output) value of the victim cell is collected every time the write values of the cells captured under the scanning mask match the assumed input values. This process is repeated for each combination of input values for the cells within the mask. For each such set of cell input values, averaging the collected read values for the victim cell provides the conditional sample mean, which is an unbiased estimate of the victim cell's read value conditioned on the particular input values for the cells within the mask. In this conditional mean, the uncertainties introduced due to random noise as well as the effect of changing input values for the cells outside the assumed mask are averaged out. This allows us to focus on the statistical characteristics of the interference coming from the cells within the mask.

Existing data analysis methods are based on simply fixing the write values of the victim cell and the suspected interfering cells in the neighborhood, scanning over the data collecting the victim cell's read value whenever the local input pattern matches the assumed write values, and examining the histogram plots of the corresponding read values [10]. This method can identify the interfering cells causing the mean-shift in the victim cell's read value, but fails to isolate the sources of the random variations in it. In contrast, the technique discussed in this paper can quantify the effects of different sources on the random variations of the victim cell's read value.

Using the proposed analysis method, we present the results of the analysis on data taken from the state-of-the-art $2 \times$ nm NAND flash memory with MLCs. In particular, input-pattern-

Manuscript received June 07, 2012; revised October 08, 2012; accepted October 28, 2012. Date of publication February 11, 2013; date of current version July 24, 2013. This work was supported in part by SK Hynix Semiconductor and the National Research Foundation of Korea under grant no. 2011-0029854. This work was presented in part at IEEE Globecom 2011, ICNC 2012 and ICEIC 2012. This paper was recommended by Associate Editor C.-C. Tseng.

J. Moon and J. No are with the Department of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon, Korea (e-mail: jmoon@kaist.edu, eee4u@kaist.ac.kr).

S. Lee, S. Kim, and S. Choi are with Department of Flash Solution Development, SK Hynix Semiconductor, Icheon, Korea (e-mail: sangchul.lee@skhynix.com, sangsik1.kim@skhynix.com, seokhwan.choi@skhynix.com).

Y. Song is with the Department of Electronic Engineering, Hanyang University, Seoul, Korea (e-mail: yhsong2008@hanyang.ac.kr).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCSI.2013.2239116

dependent noise variances and mean-interference are quantified. Cell-to-cell coupling constants are extracted. The effects of program/erase (P/E) cycles as well as baking are investigated. Baking simulates the aging process for understanding the retention characteristics while repeated P/E cycles can shed light on the endurance behavior of memory cells. Our results show that the interference characteristics change considerably less with P/E cycles and baking than do the random noise contributions. Accordingly, random noise appears more important than the effect of interference with aging and wear.

II. DATA ANALYSIS METHOD

A. Problem Statement

We discuss how, in particular, changing the input value of a certain cell would affect the output value of the victim cell. This type of cell-to-cell interference arises due to floating-gate voltage coupling [1]. We also investigate how the noise variance of the victim cell gets affected by the input pattern of the local cells. We show that the impact of an arbitrarily chosen cell or group of cells on the victim cell can be quantitatively understood. We are given only a limited set of data, and the statistical analysis needs be planned carefully.

Let x_{jk} and r_{jk} denote the input and soft output values of the jk -th memory cell (i.e., the memory cell on the j -th word line and the k -th bit line). See Fig. 1. Let \mathbf{x} represent the entire set of input values for the two-dimensional (2-D) array of cells. We can write:

$$r_{jk} = x_{jk} + n_{jk}(\mathbf{x}) + f_{jk}(\mathbf{x}) \quad (1)$$

where $n_{jk}(\mathbf{x})$ is the noise associated with the jk -th cell, possibly depending on the input values of all cells, \mathbf{x} , and $f_{jk}(\mathbf{x})$ represents the interference on the jk -th cell that in general depends on the input values for all cells. Given a pair of input and output sets for a finite-size 2-D array of cells, we wish to characterize $n_{jk}(\mathbf{x})$ and $f_{jk}(\mathbf{x})$ conditioned on a specified set of input values on an arbitrary selection of potentially interfering cells. The interference signal $f_{jk}(\mathbf{x})$ can be viewed as the shift in the threshold voltage of the victim cell due to the interfering cells' taking the particular input values \mathbf{x} . The random portion $n_{jk}(\mathbf{x})$, on the other hand, is the noise in r_{jk} due to random system noise (read-side noise) plus the collective contribution of the random deviations around the nominal written charge levels of the jk -th cell (victim cell) and all individual interfering cells.

Let the error signal be defined by $e_{jk} = r_{jk} - x_{jk} = n_{jk}(\mathbf{x}) + f_{jk}(\mathbf{x})$. We are interested in answering the following questions. What is the probability density function (pdf) of e_{jk} conditioned on a specified local pattern of input values? Can we separate the noise from the interference? Where are the interfering cells and can we quantify their impact on the victim cell at position (j, k) ? Within the random noise n_{jk} , can we separate the noise arising from the victim cell itself and the noise due to the random deviations around the nominal write-levels in coupled cells?

A typical data analysis method consists of collecting the output values r_{jk} corresponding to some fixed input value x_{jk} and a specified input value for a suspected interfering cell, say, the cell at position $(j+1, k)$, as the pass is taken over all victim cell positions (j, k) . Fig. 2 shows typical histogram plots for

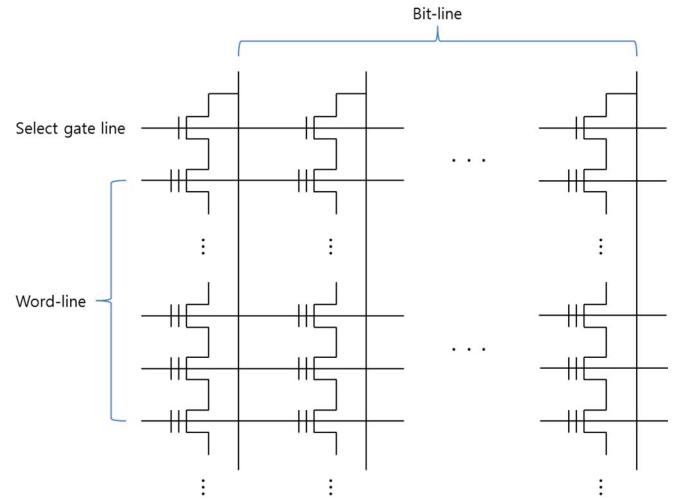


Fig. 1. Illustration of NAND Flash Memory structure.

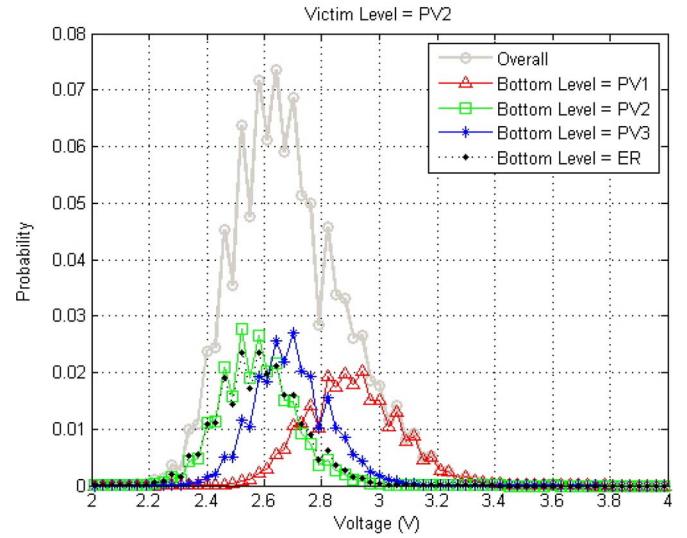


Fig. 2. Probability distributions of victim cell's output values conditioned on different input values for the interfering cell (data with no prior P/E cycles).

the conditional pdfs of r_{jk} obtained using this method. The suspected interfering cell is the one below the victim cell in this case and different conditional pdfs correspond to different input levels assumed for the interfering cell. The cells can take four different input levels in our data: the erasure (*ER*) level and three different program voltage levels, ranging from *PV1* to *PV3*. The victim cell was fixed at *PV2* in the figure. Notice that the conditional pdfs have different means and variances, signifying that the amounts of interference and noise vary depending on the input values for the victim and interfering cells.

While this method predicts with reasonable accuracy the mean shift in the output value of the victim cell due to the interfering cell, it does not allow isolation of the sources for the random variations around the means. For example, the random variations associated with each histogram plot shown in Fig. 2 include effects of both random noise and the change of the input values for influencing cells other than the cell below the victim cell.

As will be shown below, our proposed analysis method overcomes this shortcoming, allowing identification and quantitative assessment of different sources for the random variations in the victim cell's output value. In particular, our method can isolate the amount of variation in the victim cell's output value due to the changing input value of a particular interfering cell or group of cells.

B. Determining the Mask Shape

The proposed method is based on dividing the cells into a group of potentially interfering nearby cells and the rest of cells that are distant and thus are not likely to interfere. We first define a mask that captures the potentially interfering cells. We then attempt to average out the effect of the cells outside of this mask, so a focus can be made only on likely interfering cells. Any noise that does not depend on the written data is also averaged out in this process. The size and the shape of the mask must be chosen judiciously. Let us call the cells captured within the mask internal cells and the ones outside external cells. Since we are dealing with a finite set of data, if the mask size is too big, there would not be enough data samples to average out the effect of the external cells. If the mask size is too small, there would not be enough samples to reliably estimate the random distribution that arises due to internal cells. The mask size requirement will be discussed again shortly, after the presentation of the proposed analysis method.

C. Sample Mean

Once the mask shape is determined and a set of write values are assumed for the internal cells, a pass is taken over the given 2-D data set and the read value is collected whenever the local internal input values match the assumed values. Eventually an average value is to be obtained for each set of assumed write values for the mask. This average represents the read value specific to the given local internal write pattern, averaged over all possible combinations of the external cell write values as well as over all random deviations around the nominal input values of the internal cells. The system noise (e.g., electronics noise during read) is also averaged out in this process.

D. Interference and Noise Model

Let P denote the mask shape defined with respect to the position of the victim cell. We also need to define a submask P' as well as $Q = P \cap [P']^c$, where $[.]^c$ denotes the complementary set. See Table I for some example mask shapes. We shall use lower case letters p , p' and q to denote the specific combination of input values for the cells under the corresponding masks. In this sense, P , P' and Q can be viewed as random variables (RVs) while p , p' and q are corresponding specific realizations. Assuming that all signal and noise processes are stationary (i.e., statistical parameters are position-invariant), we write the output value of the victim cell as

$$r(p) = x(p) + n(p) + f(p) + E(P) \quad (2)$$

where $x(p)$ denotes the victim cell's input value corresponding to p (or the victim cell portion of p), a deterministic array of input values; the noise term $n(p)$ is a zero-mean random variable representing the various noise sources due to random vari-

TABLE I
EXAMPLE MASK PATTERN AND SUBPATTERNS

Pattern	Definition	Example
P	A mask covering the victim cell (v) and selected, potentially interfering neighboring cells (includes all cells shown)	
P'	A subset of P including the victim cell (red+blue)	
$Q = P \cap [P']^c$	A submask covering the cells in P but not those in P' (blue)	

ation induced during the writing of the victim cell and random noise during reading as well as random variations around the nominal write values in all other internal cells in P ; $f(p)$ represents interference caused by the input values of the internal cells of the mask and is non-random once the internal input pattern p is fixed; and another interference term $E(P)$ denotes potential interference coming from all cells outside the chosen mask P and is a random variable as the input values of the external cells are not specified. In writing (2), we have assumed that the victim cell's noise due to random variation around the nominal write level in each external cell is negligible, i.e., $n(P^c) = 0$. We remark that this assumption may not be valid for situations where the external cell group includes one or more interfering cells. As will be seen shortly, this paper includes the analysis of cell-to-cell interference as well as potential interference from cells outside the 3 by 3 mask. In both of these cases, in fact, we use a mask that excludes some interfering cells. Nevertheless, we note that the noise are still removed in these cases after averaging is done over external cell input values and that our analyses on interference effects remain valid.

E. Analysis Based on Conditional Means

We are ready to write the mathematical expression for the conditional mean corresponding to p :

$$\begin{aligned} \bar{r}(p) &= x(p) + \bar{n}(p) + f(p) + \bar{E}(P) \\ &= x(p) + f(p) + \bar{E}(P) \end{aligned} \quad (3)$$

where the over-bar, which denotes the statistical mean, is applied only to random variables. Assuming that the sample size is large enough, we expect $\bar{n}(p) = 0$ for any p .

At this point, we take a close look at the existing pdf extraction method described earlier. We rewrite (2) with p replaced by p' to describe the victim cell's output value conditioned on a specific p' , the set of input values for cells under some submask P' :

$$r(p') = x(p') + n(p') + f(p') + \underbrace{E(P')}_{f(Q)+E(P)} \quad (4)$$

where $f(Q)$ represents interference on the victim cell caused by cells in Q . While the same function symbol f is reused here, the context should make it clear that $f(p')$ and $f(Q)$ are dif-

ferent functions in general. Notice that in (4) it is implied that P' is chosen such that $n(Q) = 0$ (i.e., the variations around the nominal write values in cells in Q do not contribute to noise in the victim cell). Now the average (taken over all possible input values for cells outside the mask P' as well as over random fluctuations around the nominal input values of the cells in P') is:

$$\begin{aligned}\bar{r}(p') &= x(p') + \bar{n}(p') + f(p') + \bar{f}(Q) + \bar{E}(P) \\ &= x(p') + f(p') + \bar{f}(Q) + \bar{E}(P).\end{aligned}\quad (5)$$

Let us compute the variance of $r(p')$:

$$\begin{aligned}\sigma_{r(p')}^2 &= E[\{r(p') - \bar{r}(p')\}^2] \\ &= \sigma_{n(p')}^2 + \sigma_{f(Q)}^2 + \sigma_{E(P)}^2\end{aligned}\quad (6)$$

where $\sigma_{f(Q)}^2$ is the victim cell's output noise contribution due to the input-pattern changes for the cells in Q while $\sigma_{E(P)}^2$ is due to similarly introduced interference from cells outside the mask P . It is now clear that the noise variance computed in this way is a mixture of effects from multiple sources: the pattern-dependent random noise, the input pattern variations in Q and the input pattern variations in the external cells. The existing analysis does not tell us how to separate the effects of these sources.

1) Extracting $\sigma_{f(Q)}^2$, the Noise Variance Due to Local Pattern Changes: We now come back to our approach and show how noise sources can be isolated and analyzed. Let $r(P)$ represent the collection of $\bar{r}(p)$'s corresponding to all distinct p values. Then, $r(P)$ can be interpreted as a random variable representation of $\bar{r}(p)$'s, i.e.,

$$r(P) = x(P) + f(P) + \bar{E}(P)\quad (7)$$

such that $\bar{r}(p)$ is a specific instance of $r(P)$ corresponding to $P = p$. Let us fix p' and collect those $r(P)$ values corresponding to the common p' . The histogram plot of these values reflects the distribution generated by input pattern variations for cells under Q . To see this, define the corresponding random variable as

$$r(P|p') = x(p') + f(p') + f(Q) + \bar{E}(P)\quad (8)$$

where only $f(Q)$ is the random variable causing a distribution around some mean on the right hand side. These distributions are shown in Fig. 6. The mean of $r(P|p')$ is

$$\bar{r}(P|p') = x(p') + f(p') + \bar{f}(Q) + \bar{E}(P).\quad (9)$$

The variance of $r(P|p')$ can be computed using (8) and (9) as

$$\sigma_{r(P|p')}^2 = E[\{r(P|p') - \bar{r}(P|p')\}^2] = \sigma_{f(Q)}^2\quad (10)$$

which is precisely the contribution due to the input pattern variations in Q . Thus, by carefully designing the masks P and P' (and thus Q), we are able to compute the victim's output noise variance due to the input pattern variations in any specific group of cells in Q .

2) Extracting $\sigma_{n(p')}^2$, the Pattern-Dependent Random Noise Variance: First write

$$r(p) - \bar{r}(p) = n(p) + [E(P) - \bar{E}(P)]\quad (11)$$

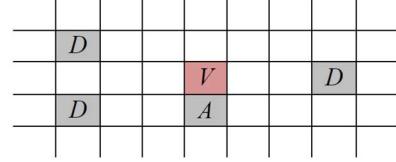


Fig. 3. A mask P containing the victim cell V , the suspected coupled cell A and the distant cells D .

which represents the mean-shifted output conditioned on a specific p . Equation (11) leads to the relation

$$\sigma_{r(p)}^2 = \sigma_{n(p)}^2 + \sigma_{E(P)}^2.\quad (12)$$

Ignoring the effect of the external cells (to be justified later using empirical data), this simply represents the noise process that depends on the local pattern p . Let the corresponding zero-mean pdf be $z(n|p)$. The variance associated with this pdf is $\sigma_{n(p)}^2$. Let us try to extract $z(n|p')$, whose variance is $\sigma_{n(p')}^2$. Clearly,

$$z(n|p') = \sum_q z(n| \underbrace{p', q}_p) Pr(q) = C \sum_q z(n|p).\quad (13)$$

where $Pr(q) = C$, which remains constant over all q . This equation describes how $z(n|p')$ is extracted from $z(n|p)$'s, allowing the computation of $\sigma_{n(p')}^2$. The random noise variance associated only with the victim cell itself, $\sigma_{n(v)}^2$, can also be obtained easily utilizing (13) with p' replaced by v and the summation taken over the input levels for all cells in P other than the victim.

3) Validation of the Noise-Variance Extraction Methods: From (6) and (10), together with the approximation $\sigma_{E(P)}^2 \approx 0$, we have

$$\sigma_{r(p')}^2 = \sigma_{r(P|p')}^2 + \sigma_{n(p')}^2\quad (14)$$

which suggests that the convolution of $z(n|p')$ with the pdf for $r(P|p')$ would match the directly measured pdf for $r(p')$. In the next section, this will indeed be shown to be true, using real data.

If P' is chosen to capture dominating interfering cells in it, then we would expect $\sigma_{n(p)}^2 \approx \sigma_{n(p')}^2$. In this case, (12) and (14) would mean that $\sigma_{r(p)}^2 \approx \sigma_{r(p')}^2 - \sigma_{r(P|p')}^2$. This will also be confirmed using real data in the next section.

4) Characterizing Cell-to-Cell Coupling: As will be justified shortly, the interference coming from cells outside the original 3 by 3 mask is negligible. Based on this, we devise a method to extract the coupling factor for a particular pair of cells focusing on the single-cell-to-single-cell interaction. The mask is now chosen so that it contains the victim cell, a potentially coupled cell whose impact on the victim cell is under investigation, and a number of distant cells whose impact on the victim can be safely ignored but are necessary to provide statistically meaningful sample size for the population of the conditional means. Fig. 3 shows an example mask that contains the victim cell V , the suspected coupled cell A and the distant cells D .

From (2), we write the read value for the victim cell as

$$r(p) = x(p) + n(p) + f(p) + f(P^c)\quad (15)$$

where $f(P^c)$ replaces $E(P)$ and is also a RV since P^c is random (i.e., the cell input values in P^c are not fixed). Realizing P consists of V, A and D , we write

$$r(p) = x(v) + n(p) + f(v, a) + f(d) + f(P^c). \quad (16)$$

Let us now imagine taking the sample mean over the p -specific data $r(p)$. The p -specific mean or the mean conditioned on p can be expressed as

$$\bar{r}(p) = x(v) + f(v, a) + f(d) + \bar{f}(P^c). \quad (17)$$

Note that the noise term has been averaged out and the last term now is the averaged interference from cells outside the mask. For the next step, consider taking a subset of these p -specific means so that each subset corresponds to a common v value. Then the RV representation of this subset can be written as

$$r(A, D) = x(v) + f(v, A) + f(D) + \bar{f}(P^c). \quad (18)$$

Note that A and D are now RVs, as the corresponding cells are no longer associated with specific input values. The mean of the RV $r(A, D)$ is

$$\bar{r}(A, D) = x(v) + \bar{f}(v, A) + \bar{f}(D) + \bar{f}(P^c). \quad (19)$$

and thus the variance of $r(A, D)$ can be written as

$$\begin{aligned} \sigma_{r(A, D)}^2 &= E[\{f(v, A) - \bar{f}(v, A)\}^2] \\ &\quad + E[\{f(D) - \bar{f}(D)\}^2] \\ &\triangleq \sigma_{f(v, A)}^2 + \sigma_{f(D)}^2 \approx \sigma_{f(v, A)}^2. \end{aligned} \quad (20)$$

The approximation in (20) is accurate since the cells in D are well separated from the victim cell. The number of cells in D must be large enough so that the sample size is big enough for evaluating the sample mean and variance of (19) and (20). On the other hand, the sample size for obtaining the conditional sample mean of (17) is inversely proportional to the total number of cells in the mask, given the fixed total data size. Thus, the size of the mask must be chosen judiciously given these conflicting goals. In the present work, capturing three cells in D , as shown in Fig. 3, yielded satisfactory results. In summary, the mask P is designed in such a way that the variance measured from the v -specific subset of p -specific conditional means is equal to the victim cell's read value variation due to the input value changes in cell A .

We note that for the MLCs under investigation, two bits written into each cell originate from two separate logical pages. The first bit is written by either retaining the original ER level of the cell or programming the cell value to a level somewhat below $PV2$. See Fig. 4. At the writing of the second and the final bit, if the first bit corresponded to the ER level, either the same level is retained or the charge level is raised to $PV1$, depending on the second bit value. If the first bit was written as the level somewhat below $PV2$, labelled $PV2'$, then the second bit would drive the new charge level to either $PV2$ or $PV3$.

Let S_A denote the threshold-voltage shift of cell A that occurs after the victim cell is programmed. Let us assume that the

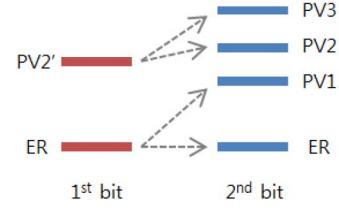


Fig. 4. Writing of a multi-level cell.

amount of threshold-voltage in the victim cell due to the coupling effect is linearly related to S_A [3], i.e.,

$$f(v, A) = c(A, v)S_A \quad (21)$$

where $c(A, v)$ is the coupling factor, which in general is assumed to be dependent on both the location of the interfering cell A and its value v . Note that since $c(A, v)$ is associated with a specific value v , it is not a RV. The variance of $f(v, A)$ is then

$$\sigma_{f(v, A)}^2 = c^2(A, v)\sigma_{S_A}^2 \quad (22)$$

where $\sigma_{S_A}^2 = E[\{S_A - \bar{S}_A\}^2]$. The RV S_A can, in general, take four different values associated with the four distinct input levels in each MLC cell. Each level in S_A occurs with probability 1/4. The final threshold-voltage levels are, assuming no interference and noise, ER , $PV1$, $PV2$ and $PV3$. Now the S_A values can be seen as $\{0, PV2 - PV2', PV3 - PV2', PV1 - ER\}$. Assuming these values occur with equal probabilities, $\sigma_{S_A}^2$ can be obtained easily, and we write

$$c(A, v) = \sigma_{f(v, A)} / \sigma_{S_A}. \quad (23)$$

Numerical results obtained for real data will be presented in the next section.

F. Mask Size Requirements

As already suggested above, the mask size must be chosen so that large enough sample sizes are attained for both the initial averaging that removes local-pattern-independent noise plus interference and the characterization of interferences due to suspected internal cells. Let us assume a given data set of size N , i.e., the data set consists of N read samples corresponding to N distinct flash memory cells. The cell input values are from a known pseudo-random sequence. Let L be the number of input levels and M_P denote the size of mask P .

Then, there exist L^{M_P} distinct p patterns and the sample size used in obtaining each p -specific conditional mean is given by

$$S(p) = N / L^{M_P}. \quad (24)$$

A subset of these L^{M_P} conditional sample means is used to examine victim cell's read distributions due to the input-level variations of a specific group of internal cells captured in submask Q . Thus the sample size associated with this second p' -specific sample mean is equal to

$$S(p') = L^{M_P} / L^{M_{P'}} = L^{M_Q} \quad (25)$$

where $L^{M_{P'}}$ and L^{M_Q} are sizes of the corresponding submasks P' and Q .

In this paper, we have $N = 8 \times 10^6$ and $L = 4$. For a 3 by 3 square-pattern mask P , for example, these parameters lead to $S(p) \approx 32$. During our data analysis, we evaluated and observed p -specific conditional means as functions of the sample size, and we could confirm that the conditional sample mean values converge safely by the time the sample size increases to 32. This indicates that the sample means that we generate from the initial averaging process are close to the true means. As for the sample size $S(p')$, with the 3 by 3 mask P and P' consisting of the victim cell and the cell below it as an example, we get $S(p') = 4^{(9-2)} = 2^{14}$. One can quickly get an idea on the accuracy in estimating the variance of a RV using this type of sample size: it is well-known that with Gaussian variables, the variance associated with the sample variance (which is a RV itself) is given by $2\sigma^4/(S-1)$, where σ^2 is the true variance and S is the sample size [12]. The corresponding root-mean-squared-error associated with the sample variance, expressed as a fraction of the true variance, is $\sqrt{2/(S-1)}$. It can be seen that with the sample size $S = 2^{14}$, the accuracy of the sample variance is comfortably high.

III. NUMERICAL RESULTS AND INTERPRETATION

Empirical data written into and read from a block of MLCs were used in the experiment and analysis. To get the results summarized in this paper, the victim cells lying on the odd-bit lines and even-bit lines were analyzed separately. The flash memory considered in this paper has the alternate odd/even bit line structure where the cells belonging to odd (even) bits lines are programmed together. In this structure, for each odd-bit-lined cell, three cells below it are programmed afterwards, thus creating potential program disturbance for the odd-bit-lined victim cell [3]. For each even-bit-lined cell, two cells on both sides as well as three bottom cells can interfere. The interference coming from the single lower cell tend to dominate for the victim cells on the odd-bit lines, whereas there are some contributions from the side cells for the even-lined victim cells. In this paper, with the exception of the discussions involving cell-to-cell coupling factors, we focus on the results associated with the victim cells that reside on the odd-bit lines. In the sequel, unless noted otherwise, we only present data focusing on the interference effect of the bottom cell. As such the submask P' contains only the victim cell and the cell below it.

A. Justifying the Assumption on $\sigma_{E(P)}^2 \approx 0$

The analysis based on sample means using special mask shapes that capture cells outside the original 3 by 3 mask reveals that the interference coming from these distance cells is negligible, i.e., $\sigma_{E(P)}^2 \approx 0$ if P is as shown in Table I. The special mask shapes used for this purpose are shown in Fig. 5. These two masks consist of five cells chosen for the purpose of estimating accurate sample means with a larger data set. Table II shows the conditional mean and variance values of captured sample mean values. Only the victim cell value is fixed in getting the conditional mean and variance. Of the overall variance $\sigma_{r(p'=v)}^2$, the sum of the conditional variances $\sigma_{r(P_a|p'=v)}^2 + \sigma_{r(P_b|p'=v)}^2$ accounts for only 0.52%, 0.87% and 0.29% for the victim level of $PV1$, $PV2$ and $PV3$,

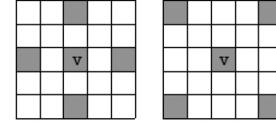


Fig. 5. New masks P_a (left) and P_b (right).

TABLE II
VARIANCE OF CELLS OUTSIDE OF THE ORIGINAL 3 BY 3 MASK

Victim	$\bar{r}(P_a v)$	$\sigma_{r(P_a v)}^2$	$\bar{r}(P_b v)$	$\sigma_{r(P_b v)}^2$
$PV1$	1.11	1.61e-04	1.11	1.04e-05
$PV2$	2.69	3.11e-04	2.69	1.04e-05
$PV3$	4.37	1.86e-04	4.37	2.17e-05

respectively. Thus, we safely conclude $\sigma_{E(P)}^2 \approx 0$ when P is set as in Table I.

B. Results for Read Data With no Prior P/E Cycles

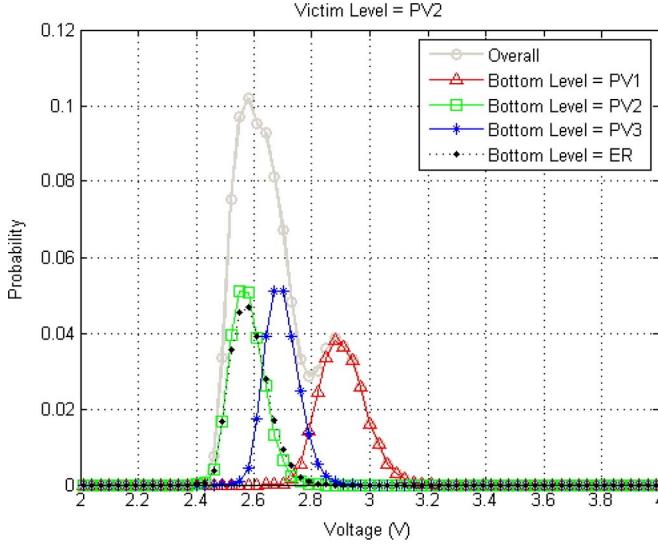
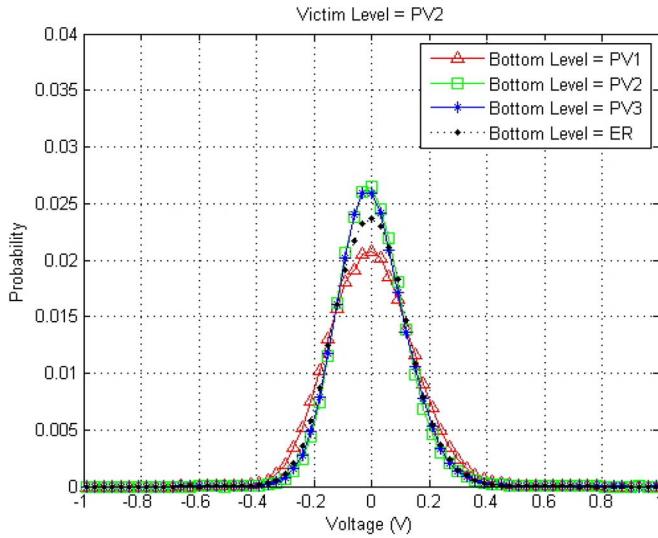
Fig. 6 shows the distribution of $r(P|p')$, the conditional sample means for different p' , where again P' includes just the victim cell and the cell right below it. Figs. 2 and 6 consistently show that the victim cell's read value contains the most bias to the right side when the input level of the bottom cell is $PV1$. This is due to the fact that the change in the applied charge level is the largest when the written cell level transitions from ER to $PV1$, inducing the worst program disturb effect in the victim cell. The second biggest transition is made when the target write level is $PV3$ (as seen in Fig. 4), as reflected in the second largest bias when the interfering cell's write level is $PV3$. Raising the charge level at each programming time involves the well-known incremental step pulse programming (ISPP) method based on repeated program-verify steps with incremental increase in the programming voltage level [11]. ISPP is used to prevent cells being programmed to a charge level higher than intended, in which case the entire block must be erased again. In ISPP, the random variation around the nominal written value depends on the incremental step size [3].

Fig. 7 shows the distribution of mean-shifted $r(p)$ for different p' values, i.e., $z(n|p')$ of (13). With $\sigma_{E(P)}^2 \approx 0$, (12) indicates this distribution arises mainly due to the randomness in the nominal read value of the victim cell as well as the random deviation around the nominal charge levels of neighboring cells, rather than the varying input patterns of the interfering cells.

Fig. 8 shows the convolution of the conditional mean distribution and the mean-shifted $r(p)$ distribution, displaying mean and variance characteristics very similar to (with the overall pdf plots much smoother than) those of the directly obtained distributions in Fig. 2, as the analysis of the previous section predicted.

Table III shows the mean-shift values in the victim cell output value caused by its bottom neighbor for different combinations of the victim cell and the bottom cell input values. For this, the captured conditional mean values are utilized. Conditioning on a specific victim cell level v , we get the average as

$$\bar{r}(P|v) = x(v) + \bar{f}(B) + \bar{f}(Q) + \bar{E}(P). \quad (26)$$

Fig. 6. Distribution of mean values $r(P|p')$.Fig. 7. Distribution of mean-shifted $r(p)$, $z(n|p')$.

Subtracting (26) from (9), we obtain

$$f'(p') \triangleq \bar{r}(P|p') - \bar{r}(P|v) = f(p') - \bar{f}(B) \quad (27)$$

which, for the given victim level, represents the mean conditioned on a specific interfering cell level relative to the unconditioned mean (averaged over all interfering cell levels). Note that $\bar{f}(B)$ is the mean interference from the bottom cell averaged over its all possible input values. The relative mean-interference values $f'(p')$ are given in Table III with P' as specified in Table I. Notice that the data corresponding to the victim level of *ER* is missing (here and throughout this section). This is due to the limitation in our experimental setting that does not allow measuring the *ER* level with sufficient accuracy.

Table IV compares the variances of mean-shifted $r(p)$ distributions, $\sigma_{r(p)}^2$, with the variances calculated using (14). The two variance values are very similar in each case of p' . The noise associated with *PV3* on the victim cell is the largest while *PV1* on it yields the least noise. If the bottom cell level is *PV1*, the

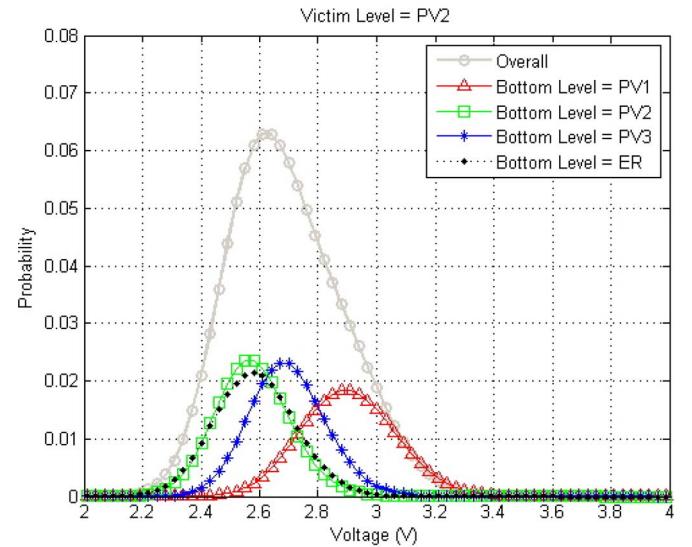
Fig. 8. Estimated $r(p')$ distribution (convolution of Figs. 6 and 7).

TABLE III
AMOUNT OF INTERFERENCE FROM BOTTOM NEIGHBOR: MEAN SHIFT

Victim level	Bottom level	$f'(p')$
<i>PV1</i>	<i>PV1</i>	0.22
	<i>PV2</i>	-0.12
	<i>PV3</i>	-0.00
	<i>ER</i>	-0.10
<i>PV2</i>	<i>PV1</i>	0.22
	<i>PV2</i>	-0.11
	<i>PV3</i>	0.01
	<i>ER</i>	-0.11
<i>PV3</i>	<i>PV1</i>	0.24
	<i>PV2</i>	-0.13
	<i>PV3</i>	-0.01
	<i>ER</i>	-0.10

noise variance has the largest value within the same victim level. To get a feel for the amount of noise contribution due to the effect of varying the input cell values in Q (i.e., the second component in (6)), we also show $\sigma_{f(Q)}^2$ in a separate column. It can be seen that the first component of (6), $\sigma_{n(p')}^2 \approx \sigma_{r(p)}^2$, is considerably larger than the second components. The sample sizes in estimating the variances are large enough to ensure that the any estimation error due to the finite sample size [which should drop as $2\sigma^4/(S-1)$ as discussed earlier] is several orders of magnitude smaller than the estimated variance. Finally, $\sigma_{n(v)}^2$ is also shown in the last column. This represents the noise arising only from the victim cell itself (including any read-side noise).

C. Cell-to-Cell Coupling Factors Extracted

Table V shows the cell-to-cell coupling factors $c(A, v)$ obtained from (23), as A changes from one neighboring cell to another. It turns out that $c(A, v)$ is not a particularly sensitive function of v so results are shown only for $v = PV1$. The results are summarized separately for the case where the victim cells are on odd-bit lines and the case where they are on even-bit lines. Because the programming sequence is such that for a given word

TABLE IV
VARIANCE OF MEAN-SHIFTED $r(p)$ VERSUS $\sigma_{n(p')}^2$ FROM (14)

Victim	Bottom	$\sigma_{r(p)}^2$	$\sigma_{r(p')}^2 - \sigma_{r(P p')}^2$	$\sigma_{f(Q)}^2$	$\sigma_{n(v)}^2$
<i>PV1</i>	<i>PV1</i>	0.017	0.017	0.004	0.013
	<i>PV2</i>	0.010	0.010	0.001	
	<i>PV3</i>	0.011	0.010	0.002	
	<i>ER</i>	0.013	0.013	0.003	
<i>PV2</i>	<i>PV1</i>	0.020	0.020	0.006	0.016
	<i>PV2</i>	0.014	0.014	0.003	
	<i>PV3</i>	0.015	0.014	0.003	
	<i>ER</i>	0.016	0.016	0.004	
<i>PV3</i>	<i>PV1</i>	0.046	0.046	0.014	0.040
	<i>PV2</i>	0.036	0.036	0.011	
	<i>PV3</i>	0.037	0.036	0.011	
	<i>ER</i>	0.040	0.040	0.012	

TABLE V
COUPLING FACTORS FOR DIFFERENT INTERFERING CELLS

V on odd-bit line			V on even-bit line		
0.00	0.00	0.00	0.01	0.00	0.01
0.01	<i>PV1</i>	0.01	0.05	<i>PV1</i>	0.06
0.02	0.11	0.02	0.03	0.11	0.04

line (row) second-bit programming in the odd-bit line cells are completed after the second-bit programming in the even-bit line cells, the program disturbance characteristics are different for the two cases. Also, for our data, for a given bit line upper (upper row or word line) cells are fully programmed before lower cells are.

Based on empirical results, we have $\sigma_{SA}^2 = 1.75$ for the affecting cells on odd-bit lines and 1.65 for those on even-bit lines. In Table V, the center cells represent the victim cell and the values that fill individual cell positions represent the corresponding coupling factors.

It can be seen that for the victim cells on odd-bit lines, the lower cell dominates in terms of the coupling effect. The top three cells have negligible effect; this makes sense since the top cells are programmed before the victim cell. The side cells (which are on even-bit lines) also have little effect as they are fully programmed before the victim cell is.

For the victim cells on even-bit lines, the lower cell again is dominating but the coupling effect for the side cells cannot be ignored. This is expected as the side cells are now on odd-bit lines, for which the second bit programming is done after the programming is completed for the victim cell.

D. Effect of Repeated P/E Cycles and Baking

1) *Pattern-Dependent Interference Effect:* Table VI shows the mean values of the victim cell for different combinations of the input values for the victim cell as well as the interfering cell, which in this case is again the cell right below the victim cell. Five different types of data represent those from cells with no prior P/E cycles, 1.5 K P/E cycles, 1.5 K P/E cycles plus baking, 3 K P/E cycles and 3 K P/E cycles plus baking, respectively. Baking here simulates one-year aging.

In all cases, the mean shift (to the right) is most pronounced when the interfering cell's input level is *PV1*. It is seen that

TABLE VI
MEAN $\bar{r}(P|p')$

Victim	Bottom	P/E 0	1.5K	1.5K +bake	3K	3K +bake
<i>PV1</i>	<i>PV1</i>	1.33	1.34	0.94	1.37	0.76
	<i>PV2</i>	0.99	1.05	0.65	1.10	0.48
	<i>PV3</i>	1.11	1.18	0.77	1.26	0.60
	<i>ER</i>	1.02	1.05	0.66	1.09	0.48
	<i>Overall</i>	1.11	1.16	0.75	1.21	0.58
<i>PV2</i>	<i>PV1</i>	2.91	2.84	2.43	2.85	2.25
	<i>PV2</i>	2.58	2.58	2.18	2.62	2.02
	<i>PV3</i>	2.70	2.70	2.30	2.75	2.14
	<i>ER</i>	2.58	2.57	2.16	2.59	2.00
	<i>Overall</i>	2.69	2.67	2.27	2.70	2.10
<i>PV3</i>	<i>PV1</i>	4.61	4.52	4.02	4.52	3.83
	<i>PV2</i>	4.24	4.23	3.76	4.26	3.60
	<i>PV3</i>	4.36	4.36	3.87	4.39	3.72
	<i>ER</i>	4.28	4.24	3.76	4.26	3.59
	<i>Overall</i>	4.37	4.34	3.86	4.36	3.69

TABLE VII
VARIANCE $\sigma_{r(P|p'=v)}^2 = \sigma_{f(Q)}^2$

Victim	P/E 0	1.5K	1.5K+bake	3K	3K+bake
<i>PV1</i>	0.021	0.017	0.018	0.017	0.019
<i>PV2</i>	0.022	0.016	0.016	0.015	0.017
<i>PV3</i>	0.033	0.023	0.021	0.020	0.020

while P/E cycles do not affect the mean read values significantly, baking reduces the read values noticeably, which is the well-known phenomenon related to the data retention issue [13].

On the other hand, the mean shift to the left due to baking is significantly more with the cells with 3 K P/E cycles than those with 1.5 K cycles. This can be explained as follows. Repeated P/E cycles result in trapping of electrons in the tunnel oxide. After 3 K P/E cycles, a fewer number of electrons are injected into the floating gate during programming because of an increased number of trapped electrons, relative to the media with 1.5 K cycles. Baking tends to release the trapped electrons in the tunnel oxide. Thus, after baking, there would remain a smaller number of electrons in the floating gate of the cells with 3 K P/E cycles than those with 1.5 K P/E cycles.

Table VII shows victim cell's output variances due to input pattern variations in Q , where in this case Q covers all cells in the 3×3 mask other than the victim cell. It appears that P/E cycles tend to reduce the variance while baking does not change it much (the changes with baking seem mostly in the direction of increasing the variance but the effect appears fairly small). This suggests that P/E cycles reduce the coupling effect whereas baking does not seem to make significant differences.

Table VIII shows the mean shift values due to the interfering bottom cell, $f'(p') \approx f(p') - f(B)$ for each value of p' . While the interference effect $f(p')$ is less visible due to the presence of the second term $f(B)$, we can still make useful interpretations. Choose the case where the interfering cell input is *PV1*, which creates the largest coupling effect. In this case, we know for sure $f(p')$ is greater than $f(B)$, the interference averaged over all possible input values for the interfering cell. In fact, the table

TABLE VIII
RELATIVE MEAN-INTERFERENCE $f'(p')$

Victim	Bottom	P/E 0	1.5K	1.5K +bake	3K	3K +bake
<i>PV1</i>	<i>PV1</i>	0.22	0.19	0.19	0.16	0.18
	<i>PV2</i>	-0.12	-0.11	-0.11	-0.10	-0.10
	<i>PV3</i>	-0.00	0.03	0.01	0.05	0.02
	<i>ER</i>	-0.10	-0.10	-0.10	-0.12	-0.10
<i>PV2</i>	<i>PV1</i>	0.22	0.17	0.17	0.15	0.15
	<i>PV2</i>	-0.11	-0.09	-0.09	-0.09	-0.08
	<i>PV3</i>	0.01	0.03	0.03	0.05	0.04
	<i>ER</i>	-0.11	-0.11	-0.11	-0.11	-0.11
<i>PV3</i>	<i>PV1</i>	0.24	0.18	0.17	0.16	0.15
	<i>PV2</i>	-0.13	-0.11	-0.09	-0.10	-0.08
	<i>PV3</i>	-0.01	0.02	0.02	0.03	0.03
	<i>ER</i>	-0.10	-0.10	-0.09	-0.10	-0.10

shows that $f'(p')$ is positive for all cases where the bottom cell input is *PV1*. Now, going back to Table VI, of all cells in Q , it is reasonable to assume that the bottom cell exert the largest interference on the victim. Thus, we should expect a consistent story between Tables VI and VIII with regards to how the coupling effect changes with P/E cycles and baking. Focusing on the cases where the bottom cell is *PV1* in Table VI and comparing the numbers with those in Table VIII, it can be confirmed that this is indeed true; namely, the mean-interference is reduced with P/E cycles whereas it remains largely comparable with baking.

Recall that interference arises as a cell programmed in advance is disturbed by an adjacent cell that is programmed afterwards. Also recall that an MLC is written via a two-step process where the second bit is written sometime after the first bit has been written. In this two-step process, the interference induced at the writing of the first bit is effectively removed during the writing of the second bit by successive program-verify steps. Thus, we can ignore the interference due to the writing of the first bit and focus only on the interference effect created during the writing of the second and final bit (See Fig. 4). In addition, notice that there is no explicit write process to write an *ER* level. Also, the difference between *PV2'* and *PV2* is small enough that we assume raising a cell value from *PV2'* to *PV2* does not induce any significant interference on its neighboring cells.

In summary, we assume that if the final cell written value is either *ER* or *PV2*, then the cell does not exert any significant interference on its neighbors. Based on this assumption, for each of the last five columns and for each victim cell value in Table VI, we average the mean values corresponding to the two cases where the bottom cell is either *PV2* or *ER*. We take this value as the bias value that can be removed from the mean values of Table VI to obtain the absolute mean-interference values $f(p')$. In other words, for the victim level *PV1* or *PV3*, we have $\bar{r}(P|p') \approx x(p') + f(p') + \bar{f}(Q)$, whereas for the victim level *ER* or *PV2*, this changes to $\bar{r}(P|p') \approx x(p') + 0 + \bar{f}(Q)$. Thus, the difference between the two equations reduces to $f(p')$. Table IX shows the absolute mean-interference values obtained with this method. Again, we confirm that the mean-interference is reduced with P/E cycles on the cases

TABLE IX
MEAN-INTERFERENCE $f(p')$

Victim	Bottom	P/E 0	1.5K	1.5K +bake	3K	3K +bake
<i>PV1</i>	<i>PV1</i>	0.32	0.29	0.29	0.28	0.27
	<i>PV2</i>	0	0	0	0	0
	<i>PV3</i>	0.12	0.14	0.12	0.15	0.13
	<i>ER</i>	0	0	0	0	0
<i>PV2</i>	<i>PV1</i>	0.32	0.28	0.27	0.26	0.26
	<i>PV2</i>	0	0	0	0	0
	<i>PV3</i>	0.12	0.13	0.12	0.14	0.12
	<i>ER</i>	0	0	0	0	0
<i>PV3</i>	<i>PV1</i>	0.34	0.28	0.26	0.26	0.24
	<i>PV2</i>	0	0	0	0	0
	<i>PV3</i>	0.12	0.12	0.11	0.13	0.12
	<i>ER</i>	0	0	0	0	0

where the bottom cell is *PV1*. On the other hand, the mean-interference appears increasing with P/E cycles on cases where the bottom cell is *PV3*.

Recall that the amount of interference on a victim cell depends on the amount of the final charge added to the affecting cell during the writing of the second bit. This means that if the bottom cell's final program level is *PV1*, the amount of interference on the victim cell will be proportional to the difference (*PV1* – *ER*); likewise, if the bottom cell's final write level is *PV3*, then the amount of interference would be proportional to (*PV3* – *PV2'*). The fact that the amount of interference changes with P/E cycles suggests that these differences in the written levels also change as functions of P/E cycle numbers. It is already known that the threshold voltage of an erased cell increases significantly with an increasing number of write/erase cycles, while a programmed cell is typically only slightly affected [14]. This explains why the mean-interference is reduced with P/E cycles when the bottom cell is *PV1*. On the other hand, the observation that interference increases with P/E cycles when the bottom cell level is *PV3* seems to suggest the difference (*PV3* – *PV2'*) increases with P/E cycles. Assuming that the coupling factor does not increase with P/E cycles, this implies that the threshold level associated with *PV3* again rises as the number of P/E cycles increases. We conjecture that this is due to the effect of interface states, which tends to increase in high program levels.

A brief physics-based explanation is given as follows on the phenomena of threshold voltage changes with P/E cycles. In a NAND flash cell, as P/E cycles increase, the trapped charge in the tunnel oxide also increases, creating repulsive electric fields in two opposite directions. The repulsive fields slow down tunneling in both directions. The channel-side field tends to raise the threshold voltage levels in both erased and programmed cells, whereas less tunneling on the other side leads to a higher threshold voltage for an erased cell and a lower threshold voltage for a programmed cell. Thus, the two effects add in the case of an erased cell in pushing the threshold voltage up as P/E cycles increase, while they tend to cancel each other in a programmed cell [14]. Furthermore, a higher program level creates more interface traps, inducing a rise in the threshold

TABLE X
NOISE VARIANCE $\sigma_{n(p')}^2$

Victim	Bottom	P/E 0	1.5K	1.5K +bake	3K	3K +bake
PV1	PV1	0.017	0.025	0.042	0.037	0.062
	PV2	0.010	0.017	0.032	0.032	0.047
	PV3	0.011	0.021	0.035	0.043	0.052
	ER	0.013	0.018	0.036	0.030	0.049
	Overall	0.013	0.020	0.036	0.036	0.052
PV2	PV1	0.020	0.028	0.045	0.036	0.064
	PV2	0.014	0.019	0.037	0.028	0.055
	PV3	0.015	0.022	0.038	0.034	0.056
	ER	0.016	0.021	0.040	0.028	0.058
	Overall	0.016	0.023	0.040	0.031	0.058
PV3	PV1	0.046	0.057	0.074	0.065	0.092
	PV2	0.036	0.044	0.063	0.052	0.078
	PV3	0.037	0.046	0.063	0.054	0.079
	ER	0.040	0.051	0.068	0.059	0.085
	Overall	0.039	0.049	0.067	0.057	0.084

voltage level with increased P/E cycles. It has been observed that the interface traps increase the native threshold voltage without diminishing tunneling [15].

2) *Noise Effect Due to Random Deviations Around Nominal Cell Values in P' :* Table X shows the noise variance component of the victim cell due to random fluctuations around the nominal cell values for all cells in P' , for different combinations of the cells' input values. The pattern-dependent nature of noise is clearly revealed. In all cell types, the noise variance is the largest when the victim cell's input is PV3 and the bottom cell's input is PV1. For a given victim cell level, the noise is the largest when the interfering cell has PV1 as input. It is also clear that the noise variance increases significantly with P/E cycles and aging; compared to the pattern-dependent interference case, the increase in the noise variance is considerably more pronounced. Table X shows the noise variance due to the randomness associated with the victim cell itself (including the read-side system noise). Again the noise power increases substantially with P/E cycles and aging.

E. Application: Channel Models and State-Transition Modeling

The results obtained above can be used to construct a simple channel model that can be used to design and analyze signal processing strategies. The input/output model of (1) can be simplified as

$$r_{jk} = x_{jk} + n_{jk}[\sigma^2(x_{jk}; x_{j+1,k})] + f(x_{jk}; x_{j+1,k}) \quad (28)$$

when the inputs associated with only the victim cell itself and its bottom neighbor are assumed to affect the channel output. When both the three bottom cells and two side cells are considered together, the input/output model is given as

$$\begin{aligned} r_{jk} = x_{jk} + n_{jk}[\sigma^2(x_{jk}; x_{j,k\pm 1}; x_{j+1,k}; x_{j+1,k\pm 1})] \\ + f(x_{jk}; x_{j,k\pm 1}; x_{j+1,k}; x_{j+1,k\pm 1}). \end{aligned} \quad (29)$$

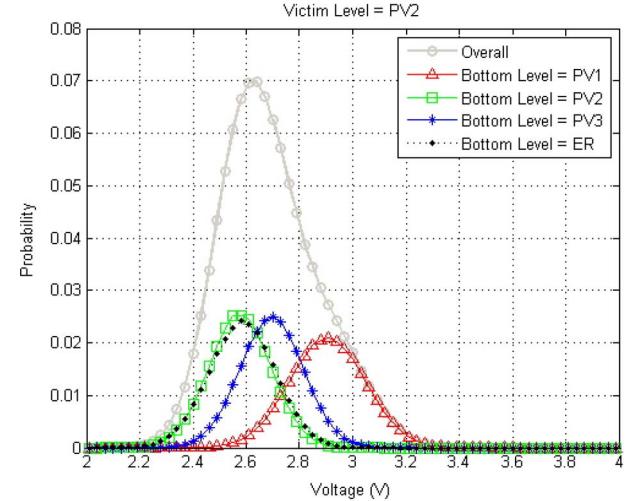


Fig. 9. Simulated output distribution.

In our simple experiment we focused on odd-bit-lined victim cells and generated read samples corresponding to pseudo random input bits according to the model of (28). We assumed that the noise n_{jk} was Gaussian with its pattern-dependent variance obtained from $\sigma_{n(p')}^2$ of Table X and that the pattern-dependent mean interference f could be determined by Table IX. We then proceeded to count the errors using a simple threshold detector with slicing levels placed in the middle of the nominal write levels. For the real data, the error rate was $1.502e - 05$ whereas for the simple pattern-dependent Gaussian noise and mean-interference model the error rate was $1.037e - 05$. Given the simple nature of the model involving only one interfering cell and the Gaussian noise assumption, the error rate results appear fairly close. Fig. 9 shows distribution of simulated output generated by above simple channel model. The statistical distributions are similar between Figs. 8 and 9.

An input/output channel model such as the one given in (28) can also be used to run advanced sequence detectors such as the Viterbi and the BCJR algorithms [16], [17]. First, a proper state-transition diagram corresponding to the input/output model needs be constructed under the assumption that the channel output samples are fed to the detector in some sequential fashion. A trellis can then be constructed so that best candidate input paths can be tracked with processing complexity that remains constant as the channel output samples are fed in sequentially. The state-transition diagram corresponding to the channel model of (28) is shown in Fig. 10. The state variable is defined in such a way that the current state variable plus the new detector input variable completely specifies the noiseless channel output as well as the necessary statistical parameters of the noise. The current state variable and the new detector input together should also fully determine the new state variable. The state variable in this case is simply equal to the current input $U_{jk} = x_{jk}$, and a state-transition occurs to a new state of $U_{j+1,k} = x_{j+1,k}$ with the new input $x_{j+1,k}$. This state-transition diagram gives a rise to a trellis with 4 states with each state possibly transitioning to each of the four states in the next stage. The assumption in this state-transition description is that the channel outputs arrive at the detector input sequentially

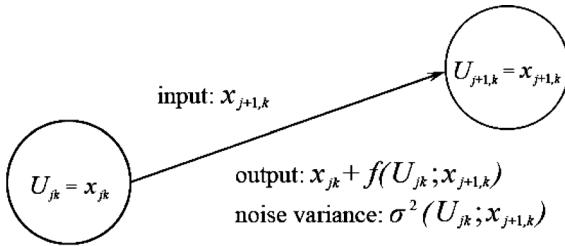


Fig. 10. State transition diagram for the channel model of (28).

along a given bit line. This assumption may not be practical, however, since in current flash memory systems, logical pages that are far apart from one another (in the data stream arriving at the controller from the host side) get written along the same bit line and a large page buffer would be necessary for a sequential processing of read samples taken from logically distant multiple pages. Nevertheless, in principle sequential processing that finds the best path along this trellis yields the optimal detection quality, given the channel model of (28).

When interference exists in more than one direction as in the model of (29), where interferences come from the vertical, horizontal and diagonal directions, it is not possible to construct a trellis that allows signal processing with complexity that remains time-invariant. This is because the global likelihood or *a posteriori* probability function cannot be decomposed into simpler terms for a recursive computation. A practically viable approach in these cases would be to construct a suboptimal trellis. For example, imagine that the detector scans the read samples along a given word line (along a row in the 2-D array of stored read samples) from left to right, taking and processing one read sample at a time. One possible construction of a state-transition model corresponding to the channel model of (29) is based on defining the state variable as a collection of four cell inputs, $U_{jk} = \{x_{j,k-1}; x_{jk}; x_{j+1,k-1}; x_{j+1,k}\}$, and defining the detector input as $W_{jk} = \{x_{j,k+1}; x_{j+1,k+1}\}$. Again, the read samples from multiple pages are assumed to be available to the detector, but here this assumption is more reasonable since logically proximate pages are written along the same word line. The state transition occurs from $U_{jk} = \{x_{j,k-1}; x_{jk}; x_{j+1,k-1}; x_{j+1,k}\}$ to $U_{j,k+1} = \{x_{jk}; x_{j,k+1}; x_{j+1,k}; x_{j+1,k+1}\}$; the transition completely specifies the nominal channel output $x_{jk} + f(x_{jk}; x_{j,k\pm 1}; x_{j+1,k})$ and the variance of the noise $\sigma^2(x_{jk}; x_{j,k\pm 1}; x_{j+1,k})$. Fig. 11 shows the corresponding state-transition diagram. Based on this diagram, a 4^4 -state trellis with each state possibly transitioning to 4^2 new states in the next stage can be constructed. The function $f()$ may change between odd and even bit lines. Any reasonable path-tracking sequence detection algorithm can be designed using this trellis or its reduced version. Again, the interference-generating function $f()$ and the pattern-dependent noise variance function $\sigma^2()$ can be obtained from Tables IX and X. Notice that the detector in this case will be tracking a reasonable path among possible local input pattern (or state) sequences that has likely caused the given read sample sequence. Identifying a likely state path would identify a reasonable input data sequence associated with the observed read sequence.

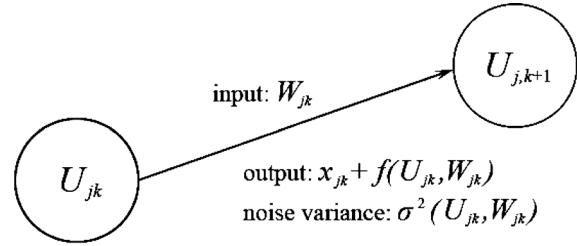


Fig. 11. State transition diagram for the channel model of (29).

It is conceptually easy to expand the state model for a better detection quality at the expense of increased processing requirements. For example, a detector can be designed to scan two rows of read samples simultaneously, taking one column of two samples, $\{r_{jk}; r_{j+1,k}\}$, at a time. In this case, the state variable is given by $U_{jk} = \{x_{j,k-1}; x_{jk}; x_{j+1,k-1}; x_{j+1,k}; x_{j+2,k-1}; x_{j+2,k}\}$ and the detector input by $W_{jk} = \{x_{j,k+1}; x_{j+1,k+1}; x_{j+2,k+1}\}$. Extensions are straightforward to cases where a rows of read samples are scanned while different combinations of b rows of input values are searched through, with $b > a$.

Finally, it is worth presenting a linearized version of the $f()$ function in (29) utilizing (21) and the results summarized in Table V. Denote the coupling factors in Table V by c_{jk} 's. Also, let $S(x_{jk})$ indicate the incremental charge (applied in writing the second bit) associated the input level x_{jk} , i.e.,

$$S(x_{jk}) = \begin{cases} PV3 - PV2' & \text{for } x_{jk} = PV3 \\ PV2 - PV2' & \text{for } x_{jk} = PV2 \\ PV1 - ER & \text{for } x_{jk} = PV1 \\ 0 & \text{for } x_{jk} = ER \end{cases} \quad (30)$$

Then, the channel model suggested by (29) reduces to:

$$r_{jk} = x_{jk} + n_{jk}[\sigma^2(x_{jk}; x_{j,k\pm 1}; x_{j+1,k}; x_{j+1,k\pm 1})] + \sum_{\substack{k-1 < l < k+1 \\ j \leq l \leq j+1 \\ il \neq jk}} c_{il} S(x_{il}). \quad (31)$$

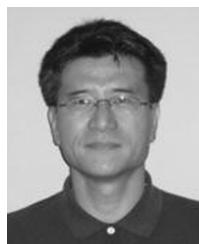
IV. CONCLUSION

A technique based on extracting conditional means with controlled local input patterns was discussed for statistically analyzing read data from flash memory cells. A specified mask pattern has been utilized. Once the conditional means were captured, they were used to quantitatively analyze the impact of the interfering cells within the mask on the victim cell's read value. Using the method discussed in this paper, we have been able to isolate the random noise from the effect of interference from coupled cells. Both interference and noise are clearly pattern-dependent. Cell-to-cell coupling coefficients were extracted as well. Memory cells with different P/E cycles and with/without baking have also been analyzed. The victim cell's noise due to random write-level fluctuations in all cells increases significantly with P/E cycles and with aging, whereas the victim cell's noise component due to the changing local input pattern does not vary as much with repeated P/E steps or baking. This implies that coupling characteristics do not change significantly with P/E cycles and aging. The proposed technique can be used to

aid the flash memory design process as well as the design of advanced coding and signal processing algorithms that require accurate statistical characterization of the noise and interference.

REFERENCES

- [1] J. Lee, S. Hur, and J. Choi, "Effects of floating-gate interference on NAND flash memory cell operation," *IEEE Electron Device Lett.*, vol. 23, pp. 264–266, May 2003.
- [2] K. Kim, "Future memory technology: Challenges and opportunities," in *Proc. Int. Symp. VLSI Technol.*, Apr. 2008, pp. 5–9.
- [3] G. Dong, S. Li, and T. Zhang, "Using data postcompensation and predistortion to tolerate cell-to-cell interference in MLC NAND flash memory," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 10, pp. 2718–2728, Oct. 2010.
- [4] G. Dong, N. Xie, and T. Zhang, "On the use of soft-decision error-correction codes in NAND flash memory," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 58, no. 2, pp. 429–439, Feb. 2011.
- [5] Y.-M. Lin, C.-H. Yang, C.-H. Hsu, H.-C. Chang, and C.-Y. Lee, "A MPCN-based parallel architecture in BCH decoders for NAND flash memory devices," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 58, no. 10, pp. 682–686, Oct. 2011.
- [6] H.-C. Chang, C.-C. Lin, T.-Y. Hsiao, J.-T. Wu, and T.-H. Wang, "Multi-level memory systems using error control codes," in *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS)*, May 2004, pp. II 393–II 396.
- [7] J. Moon, J. No, S. Lee, S. Kim, J. Yang, and S. Chang, "Statistical analysis of flash memory read data," in *Proc. IEEE GLOBECOM*, Dec. 2011, pp. 2925–2930.
- [8] J. Moon, J. No, S. Lee, S. Kim, J. Yang, and S. Chang, "Noise and interference characterization for MLC flash memories," in *Proc. ICNC*, Jan. 2012, pp. 588–592.
- [9] J. No, J. Moon, J. Yang, S. Joo, S. Lee, S. Choi, H. Lee, and T. Zhang, "Characterizing cell-to-cell coupling in flash memory," in *ICEIC*, Feb. 2012.
- [10] T. Himeno, N. Matsukawa, H. Hazama, K. Sakui, M. Oshikiri, K. Masuda, K. Kanda, Y. Itoh, and J. Miyamoto, "A new technique for measuring threshold voltage distribution in flash EEPROM devices," in *Proc. IEEE Int. Conf. Microelectronic Test Structures*, Mar. 1995, pp. 283–287.
- [11] K. Suh, B. Suh, Y. Lim, J. Kim, Y. Choi, Y. Koh, S. Lee, S. Kwon, B. Choi, J. Yum, J. Choi, J. Lim, and H. Lim, "3.3 V 32 Mb NAND flash memory with incremental step pulse programming scheme," *IEEE J. Solid-State Circuits*, vol. 30, no. 11, pp. 1149–1156, Nov. 1995.
- [12] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, 2nd ed. New York: McGraw-Hill, 1984, pp. 178–187.
- [13] J. Lee, J. Choi, D. Park, and K. Kim, "Data retention characteristics of sub-100 nm NAND flash memory cells," *IEEE Electron Device Lett.*, vol. 24, no. 12, pp. 748–750, Dec. 2003.
- [14] S. Aritome, R. Kirisawa, T. Endoh, R. Nakayama, R. Shirota, K. Sakui, K. Ohuchi, and F. Masuoka, "Extended data retention characteristics after more than 1E4 write and erase cycles in EEPROMs," in *Proc. IRPS*, Mar. 1990, pp. 259–264.
- [15] 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 cell," in *Proc. IRPS*, Mar. 2003, pp. 497–501.
- [16] G. D. Forney, Jr., "The Viterbi algorithm," *Proc. IEEE*, vol. 61, no. 3, pp. 268–278, Mar. 1973.
- [17] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. 20, no. 2, pp. 284–287, Mar. 1974.



Jaekyun Moon (F'12) received the Ph.D. degree in electrical and computer engineering at Carnegie Mellon University, Pittsburgh, PA, USA.

He is a Professor of electrical engineering at KAIST. From 1990 through early 2009, he was with the faculty of the Department of Electrical and Computer Engineering at the University of Minnesota, Twin Cities. He consulted as Chief Scientist for DSPG, Inc. from 2004 to 2007. He also worked as Chief Technology Officer at Link-A-Media Devices Corporation. His research interests are in the area of channel characterization, signal processing and coding for data storage and digital communication.

Prof. Moon received the McKnight Land-Grant Professorship from the University of Minnesota. He received the IBM Faculty Development Awards as well as the IBM Partnership Awards. He was awarded the National Storage Industry Consortium (NSIC) Technical Achievement Award for the invention of

the maximum transition run (MTR) code, a widely-used error-control/modulation code in commercial storage systems. He served as Program Chair for the 1997 IEEE Magnetic Recording Conference. He is also Past Chair of the Signal Processing for Storage Technical Committee of the IEEE Communications Society. In 2001, he cofounded Bermai, Inc., a fabless semiconductor start-up, and served as founding President and CTO. He served as a guest editor for the 2001 IEEE JSAC issue on Signal Processing for High Density Recording. He also served as an Editor for IEEE TRANSACTIONS ON MAGNETICS in the area of signal processing and coding for 2001–2006.



Jaehyeong No received the B.S. degree with honor in electrical and electronic engineering from Yonsei University, Seoul, Korea, in 2009 and the M.S. degree in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 2011. He is currently working toward the Ph.D. degree in the Department of Electrical Engineering at KAIST.

Since 2009, he has been a member of the Communication and Storage Laboratory, KAIST. His research interests include coding theory, equalization/detection and signal processing with application to digital communication and data storage systems.



Sangchul Lee received the B.S. and M.S. degrees in electrical engineering from Chong-ju University, Chong-ju, Korea, in 2000 and 2002, respectively.

He has worked in the area of IP design for signal processing applications for 12 years. His recent focus has been on signal processing and error correction codes for NAND flash devices. He is a Senior Engineer of the Flash Development Department of SK Hynix, Inc.



Sangsik Kim received the B.S. degree in electrical engineering from Kyunghee University and the M.S. degree in electrical engineering from Yonsei University, Seoul, Korea, in 2006 and 2009, respectively.

Currently, he is working on NAND flash devices and controller systems for flash memory. He is an Engineer in the Flash Development Department of SK Hynix, Inc.



Seokhwan Choi received the B.S. degrees in physics and electronics engineering from Wonkwang University, Iksan, Korea, in 2000.

His interests are reliability and signal processing issues for Flash memories. He is an Engineer in the Flash Development Department of SK Hynix, Inc.



Yunheub Song received the M.S. degree in electronic engineering from Hanyang University, Seoul, Korea, in 1992, and the Ph.D. degree in intelligent mechanical engineering from Tohoku University, Sendai, Japan, in 1999.

He has researched semiconductor device and circuit design for more than 20 years in Semiconductor R&D Center, Samsung Electronic Co., Korea. He is currently an Associate Professor in Electronic Engineering, Hanyang University, Seoul, Korea. His research interests include device structure and architecture for memory and logic applications, circuit design and algorithm for low power and high speed, and bio-systems based on semiconductor technology.