Characterization of Inter-Cell Interference in 3D NAND Flash Memory

Suk Kwang Park and Jaekyun Moon, *Fellow, IEEE*

***Abstract*— We characterize inter-cell interference in commercial three-dimensional NAND flash memory. By writing random data into 3D NAND and collecting sample means and sample variances of cell values corresponding to a particular set of input values in fixed relative neighboring cell locations, it is shown that the interference coming from any target cell locations can be measured. We observe that four neighboring cells, two along the same pipe and two along the same bit line, are responsible for most of the interference exerted on a given victim cell. Contrary to the general belief, the total amount of interference is found to be fairly significant even in 3D NAND; if compensated properly, the number of errors can be reduced significantly.**

***Index Terms*— Equalizer, interference, 3D NAND, flash, compensation, characterization.**

I. Introduction

THE rapid advent of information technology results in an explosive increase in the amount of data generated globally. Enterprise data centers and personal storage devices are expected to quickly store and read large amounts of data with reduced power while safeguarding the data from external impacts. To comply with these demands, solid state drives (SSDs) with 2-dimensional NAND flash memory were introduced. However, the cost per bit of SSDs is higher than that of hard disc drives (HDDs) [1]. To increase NAND chip capacity and density in SSDs, researchers and engineers have scaled down the cell size close to 10 nm as of the writing of this article; however, narrowing physical distance between the cells greatly increases inter-cell interference due to voltage coupling between the cells [2]. The charge change of a neighboring cell by its programming operation induces extra charges at the victim cell, and this causes a drift in the threshold voltage of the victim cell. Because of this limitation, it is increasingly difficult to improve cell density in 2D NAND.

More recently, 3D NAND flash memory has been introduced to replace 2D NAND [3]. By adopting vertical cell stacking, the distance between the cells in 3D NAND is increased compared to 2D NAND. Additionally, the charge-trapped flash (CTF) materials used as floating gates (F/Gs) in 3D NAND [4] help reduce inter-cell interference significantly compared to 2D NAND [5]. Compared to 2D NAND, 3D NAND stacks the cell in a vertical way to increase the cell density per area. Note that vertical stacking is also possible at the chip level, as done for hybrid 3D chips containing NAND, DRAM, SSD, controller and ReRAM in [6].

We briefly explain the conceptual structure and operation of a 3D NAND with multi-level cells (MLCs) which stores 2 bits in a physical cell. A vertical cross-section of one pipe in Fig. 1(a) shows how the insides of a cylindrical pipe and each metal layer are organized. The innermost layer is poly-Si, which works as a sensing current path. The outer layer with CTF stores electrons. Between these two, there exists a tunneling oxide layer. All these layers form a cylindrical pipe with a doughnut shape. In read operation, the select gate, which can be seen in Fig. 1 [7], turns on/off the pipes to read the cells contained in the word line. The capacity of a chip is determined based on the number of pipes and the number of metal layers acting as the word line (WL), as shown in Fig. 1(b).

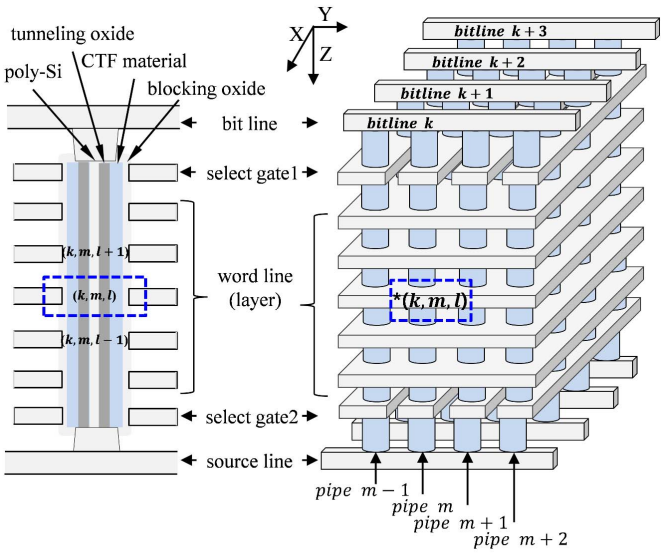

Fig. 1. The structure of cells in 3D NAND.

Manuscript received April 8, 2020; revised July 8, 2020 and October 4, 2020; accepted October 24, 2020. Date of publication January 8, 2021; date of current version February 23, 2021. This work was supported in part by the National Research Foundation of Korea under Grant 2019R1I1A2A02061135 and in part by SK hynix. This article was recommended by Associate Editor P. K. Meher. *(Corresponding author: Jaekyun Moon.)*

The authors are with the School of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon 34141, South Korea (e-mail: sukkwangpark@kaist.ac.kr; jmoon@kaist.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSI.2020.3047484.

Digital Object Identifier 10.1109/TCSI.2020.3047484

1549-8328 © 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Korea Advanced Inst of Science & Tech - KAIST. Downloaded on April 09,2021 at 01:42:54 UTC from IEEE Xplore. Restrictions apply.

The poly-Si channel is connected to both the bit line and the source line to sense the cell data. When a programming voltage is applied to a given target cell at the k-th bit line, the m-th pipe and the l-th layer, electrons are stored in the doughnut-shaped region of the cell, and a group of cells as a logical page in the l-th layer are programmed at the same time [8]. The sensing electrical current flows from the bit line to the source line sequentially, to verify that the electrons are inserted properly to one of the four predetermined states. However, as the number of stacking layers continuously increases to accommodate more cells on the wafer [9], the electrical resistance of the poly-Si in the pipe increases, which results in a reduced data-sensing current as the cells are read. Therefore, even a small variation in the number of electrons according to the programming state greatly affects the sensing current, and this change can cause data errors. In particular, the change is amplified more in triple-level cells (TLCs) where the voltage difference between programming states is small.

To make the pipe process stable, it is advantageous to widen the diameter of the pipe, or to thin the insulating layer interposed between the WL layers to lower the overall height; however, both remedies decrease the cell-to-cell distance vertically as well as horizontally (pipe to pipe), which may result in increased inter-cell interference and cause the bridge failure between cells. Since the number of layers is bound to increase, it is no longer safe to ignore the impact of inter-cell interference even in 3D NAND.

In this article, we indeed show that inter-cell interference in real 3D NAND can be significant; we present a method to characterize 3D NAND inter-cell interference as well as a scheme to compensate it to reduce data errors. This compensation method is expected to mitigate overall data errors in 3D NAND when used in conjunction with an error correcting code (ECC).

We treat the 3D NAND inter-cell interference much like inter-symbol interference in high-speed digital communication systems, and attempt to predict it based on the programmed states (input values) of the cells in the local region of memory.

We first model the soft output of a victim cell as the sum of the input, interference and noise values as in [10]. We characterize the mean and variances of the cell read values conditioned on specific input values in some potentially interfering cell locations, as suggested in [10] for 2D NAND. However, unlike [10], we do not intend to isolate all sources of noise in this article, but rather focus on characterizing interference mean and variance conditioned on certain neighboring 3D cells. As such, we do not rely on the various mask shapes used in [10] when capturing read data; we rely on simpler sample-mean and sample-variance measurements.

Due to the structurally complicated nature of 3D NAND as well as the extra dimension to deal with in the data capture space, it is critical to identify dominant interfering cells first and judiciously balance the estimation accuracy with computational complexity.

The 3D cells to be examined are placed under reliability stresses such as hot-temperature data retention (HTDR) and low-temperature data retention (LTDR), after a number of program/erase (P/E) cycles. The amount of interference generated by the affecting cells is then measured using the conditional average method based on fixing the input values for these cells while experimentally capturing cell output values. Based on the interference values and noise variances estimated, it is shown that the modeled soft outputs fit well with the actual soft outputs. After compensating the interference values, the number of bit errors is reduced by as much as 50%. The ability to compensate interference strongly depends on the chosen coverage of actual interfering cells, which in turn affects the read latency. There is a tradeoff between compensation accuracy and read latency.

As for related past works, most of the existing countermeasures to overcome cell interference issues are based on improving the electrical and physical cell characteristics in the manufacturing process [11], although some efforts exist towards the development of robust ECCs in NAND flash application controllers [12], [13]. In [14], the authors characterized and analyzed threshold voltage distributions in 2D NAND. Some others studied compensating interference calculated by the actual soft output and parasitic capacitance between the victim and neighbor cells in 2D NAND [15], [16]. Also, the authors of [17] proposed a method to reduce the ripple of programming bias, which resulted in a reduced variation of the threshold voltage of programmed cells. We emphasize that the materials filling the space between adjacent cells in 2D NAND are metallic materials, not dielectric materials as in 3D NAND. The interference mechanism of 3D NAND is more complicated compared to capacitive interference in 2D NAND. To our best knowledge, interference characterization and compensation for 3D NAND is largely in an uncharted territory.

A. Read Signal Model

In this study, we treat the inter-cell interference in 3D NAND as inter-symbol interference in a communication system that hampers accurate setting of detection threshold. We characterize the signal-dependent nature of the 3D NAND inter-cell interference based on statistical modeling and by making use of real data captured from writing random input symbols into an actual 3D NAND device. We then describe how the inter-cell interference portion of the output data can be compensated and evaluate by simulation the reduced number of errors based on this method.

Suppose a particular cell positioned in a specific 3D location has been programmed using an input value s. Let us focus on the read (output) value y off this cell. This cell will also be referred to as the victim cell, in relation to the neighboring cells that distort its read value y. Let u be the set of input values (programmed states) for some group of nearby cells that can potentially affect the victim cell. Note that this definition of u also implies a particular set of cell locations. Let u^c be the input values for all remaining cells. Then the read voltage value of the victim cell can be written as

$$y(s|u,u^c) = s + \mathfrak{W}(s) + \mathfrak{f}(s|u,u^c) + \mathfrak{F}_\delta(s|u,u^c) + N \quad (1)$$

where $\mathfrak{W}(s)$ is the random write process noise that depends in general on the intended programming level s as well as

Authorized licensed use limited to: Korea Advanced Inst of Science & Tech - KAIST. Downloaded on April 09,2021 at 01:42:54 UTC from IEEE Xplore. Restrictions apply.

other system parameters like the step size in the incremental step pulse programming (ISPP) procedure, and $\mathfrak{f}(s|u,u^c)$ is the inter-cell interference on the victim cell conditioned on its input value s as well as the nearby cell input values u and, in general, the remaining cell input values u^c. Here, when u and u^c are meant to represent particular sets of input values, $\mathfrak{f}(s|u,u^c)$ is presumed deterministic. However, given that the actual programmed values of the interfering cells themselves will have certain random variations with respect to the intended values u and u^c, the interference itself will be random even with u and u^c fixed. The random variable $\mathfrak{F}_\delta(s|u,u^c)$ reflects the effect of this possible variation. $\mathfrak{F}_\delta(s|u,u^c)$ may also include read voltage flucations due to potential local variations in the physical and material properties of the media. Further, N includes system noise as well as random telegraph noise (RTN) [18], and is assumed zero-mean overall.

The MLC type of NAND flash under investigation needs four states to represent two bits in a physical cell. The programming state s can be the erase state or one of the three programming verifying (PV) states, denoted as as $\{pv_0,\ pv_1,\ pv_2,\ pv_3\}$, respectively.

When the input values u and u^c are not fixed but considered as random variables, even $\mathfrak{f}(s|u,u^c)$ is necessarily viewed random. We shall use the upper case letters U and U^c when the input values are random. Note that when U or its specific realization u is spoken of, a particular *fixed* set of corresponding cell positions is also implied.

The random variable $\mathfrak{f}(s|U,U^c)$ is not zero-mean in general, and neither is the residual interference $\mathfrak{F}_\delta(s|U,U^c)$. This creates difficulty in characterizing them in the sense that their sample variances cannot be easily measured when their means are not known or cannot be measured easily. To circumvent this issue, we first define the mean of $y(s|u,u^c)$ after replacing $u,\ u^c$ with $U,\ U^c$:

$$\begin{aligned}\overline{y}(s|U,U^c) &= s+\overline{\mathfrak{W}}(s)+\overline{\mathfrak{f}}(s|U,U^c)+\overline{\mathfrak{F}}_\delta(s|U,U^c)+\overline{N}\\ &= s+\overline{\mathfrak{W}}(s)+\overline{\mathfrak{f}}(s|U,U^c)+\overline{\mathfrak{F}}_\delta(s|U,U^c) \qquad (2)\end{aligned}$$

where

$$\overline{\mathfrak{f}}(s|U,U^c)=\sum_{u,u^c}\mathfrak{f}(s|u,u^c)P(u,u^c)$$

with $P(u,u^c)$ being the probability of particular input values (u,u^c). From (2), we get $s=\overline{y}(s)-\overline{\mathfrak{W}}(s)-\overline{\mathfrak{f}}(s|U,U^c)-\overline{\mathfrak{F}}_\delta(s|U,U^c)$, and substituting this into (1) with u,u^c changed to their random counterparts U,U^c, we can write

$$\begin{aligned}y(s|U,U^c) &= \overline{y}(s|U,U^c)+[\mathfrak{W}(s)-\overline{\mathfrak{W}}(s)]\\ &\quad+[\mathfrak{f}(s|U,U^c)-\overline{\mathfrak{f}}(s|U,U^c)]\\ &\quad+[\mathfrak{F}_\delta(s|U)-\overline{\mathfrak{F}}_\delta(s|U)]+N. \qquad (3)\end{aligned}$$

Introducing the mean-adjusted random variables $W(s)=\mathfrak{W}(s)-\overline{\mathfrak{W}}(s)$, $f(s|U,U^c)=\mathfrak{f}(s|U,U^c)-\overline{\mathfrak{f}}(s|U,U^c)$, and $F_\delta(s|U,U^c)=\mathfrak{F}_\delta(s|U,U^c)-\overline{\mathfrak{F}}_\delta(s|U,U^c)$, we further write

$$\begin{aligned}y(s|U,U^c) &= \overline{y}(s|U,U^c)+W(s)\\ &\quad+f(s|U,U^c)+F_\delta(s|U,U^c)+N \qquad (4)\end{aligned}$$

where all terms on the right-hand side other than the constant $\overline{y}(s|U,U^c)$ are now zero-mean by definition. Going forward, in the interest of reducing notational burden without the risk of losing clarity, we will use the shorthand notations $y(s)$ and $\overline{y}(s)$ to mean $y(s|U,U^c)$ and $\overline{y}(s|U,U^c)$, respectively.

B. Interference Characterization

The conditional mean $\overline{y}(s|u)$, another shorthand notation we will use for $\overline{y}(s|u,U^c)$, for a specific u can be measured experimentally by writing random data into the 3D NAND flash and then averaging the read values of those cells with the specific program state s and the particular program state values u in given relative neighboring cell positions. Accordingly, (4) gives

$$\overline{y}(s|u)=\overline{y}(s)+\overline{f}(s|u,U^c)+\overline{F}_\delta(s|u,U^c) \qquad (5)$$

where all zero-mean random variables have been averaged out and

$$\begin{aligned}\overline{f}(s|u,U^c) &= \sum_{u^c} f(s|u,u^c)P(u^c)\\ \overline{F}_\delta(s|u,U^c) &= \sum_{u^c} F_\delta(s|u,u^c)P(u^c).\end{aligned}$$

Note that $F_\delta(s|u,U^c)$ for a specific u is not necessarily zero-mean although $F_\delta(s|U,U^c)$ is zero-mean by definition.

The average $\overline{y}(s)$ can be measured by simply reading all cells with input state s and averaging the read values. The difference between two measured averages gives

$$\overline{y}(s|u)-\overline{y}(s)=\overline{f}(s|u,U^c)+\overline{F}_\delta(s|u,U^c), \qquad (6)$$

providing an estimate of the conditional mean interference $\overline{f}(s|u,U^c)$ combined with the mean residual interference $\overline{F}_\delta(s|u,U^c)$.

With u fixed, (4) becomes

$$\begin{aligned}y(s|u,U^c) &= \overline{y}(s)+W(s)+f(s|u,U^c)\\ &\quad+F_\delta(s|u,U^c)+N. \qquad (7)\end{aligned}$$

Now, write $F_\delta(s|u,U^c)=\overline{F}_\delta(s|u,U^c)+F_{\delta 0}(s|u,U^c)$, where $F_{\delta 0}(s|u,U^c)$ is a zero-mean random variable. Similarly, write $f(s|u,U^c)=\overline{f}(s|u,U^c)+f_0(s|u,U^c)$, where $f_0(s|u,U^c)$ is a zero-mean random variable. Then, (7) can be rewritten as

$$\begin{aligned}y(s|u,U^c) &= \overline{y}(s)+\overline{f}(s|u,U^c)+\overline{F}_\delta(s|u,U^c)\\ &\quad+f_0(s|u,U^c)+F_{\delta 0}(s|u,U^c)\\ &\quad+W(s)+N. \qquad (8)\end{aligned}$$

It is clear that for a given u, any amount of variance observed in the measured samples of $y(s|u,U^c)$ would arise due to $f_0(s|u,U^c)$, $F_{\delta 0}(s|u)$, $W(s)$ and N. With the cell positions for u assumed fixed, think of two different sets of input values for these cells. If there were any difference between the variances in $y(s|u,U^c)$ for the two distinct input value sets u, this must have been caused by $F_{\delta 0}(s|u,U^c)$, the only random variable in (8) that depends on the input values of u. Notice that there is also randomness in $f_0(s|u,U^c)$, but this randomness depends only on the values of u^c. Thus, it would

Authorized licensed use limited to: Korea Advanced Inst of Science & Tech - KAIST. Downloaded on April 09,2021 at 01:42:54 UTC from IEEE Xplore. Restrictions apply.

depend on the cell positions of u but not on the cell input values of u.

Also, notice that $\overline{F}_{\delta}(s|u, U^c)$ always appears together with $\overline{f}(s|u, U^c)$, and there is no motivation to separate them; in other words, $\overline{F}_{\delta}(s|u, U^c)$, if any, can be thought of as an intrinsic part of $\overline{f}(s|u, U^c)$. Accordingly, we shall simply replace every appearance of $\overline{f}(s|u, U^c) + \overline{F}_{\delta}(s|u, U^c)$ with $\overline{f}(s|u, U^c)$. Further, in an effort to reduce notational burden, we shall define

$$f(s|u) := \overline{f}(s|u, U^c)$$

to mean the overall mean-interference (averaged over all input values of u^c) exerted on a victim cell with program state s by neighboring cells at the particular locations and program states specified by u. Thus, we simply rewrite (6) as

$$\overline{y}(s|u) - \overline{y}(s) = f(s|u). \tag{9}$$

Equation (8) is also rewritten as

$$y(s|u, U^c) = \overline{y}(s) + f(s|u) + f_0(s|u, U^c) + F_{\delta 0}(s|u, U^c) + W(s) + N. \tag{10}$$

C. Interference Compensation

We assume that the interference value $f(s|u)$, due to a specific input pattern in a particular group of neighboring cell positions, can be learned and prestored in the controller in the form of a table for different input values s and different cell positions/input values u, prior to the time of read compensation. This process can be accomplished via accumulation of the sample means and conditional sample means for all possible values of s and u, as instructed by (9). Once $f(s|u)$ is available, the read value of a given cell with known neighboring cell values u can be compensated according to

$$\hat{y}(s) = y(s|u) - f(s|u) \tag{11}$$

as suggested by (9), before being subject to the thresholding operation in the data detection process. Here $\hat{y}(s)$ is taken as the read value of the victim cell corresponding to a programmed value of s, with all possible interference effects averaged out over random inputs, as seen in (2).

II. Experimental Results

The 3D cells to be examined are placed under reliability stresses such as HTDR and LTDR, after 3000 program/erase (P/E) cycles, which corresponds to the reliability requirement of consumer solid state drives (CSSDs) and mobile applications. We followed the environment described in Joint Electron Device Engineering Council (JEDEC) for the reliability stresses. The details are as follows. For HTDR stress, P/E cycling was applied to cells at hot temperature over 85 °C for 3 weeks. Then in the same temperature, we programmed all cells with random data. To add retention stress corresponding to 1 year at 55 °C, we applied stress at over 85 °C for a few hours. For LTDR stress, P/E cycling was applied to cells at room temperature (25 °C) for 3 weeks. We applied stress at room temperature for 1,008 hours. Regarding the sample size, from 10 chips, we selected 100 blocks (10 blocks from each chip). We write random input values (states) to 100 blocks or 800 million cells (each block contains about 8 million cells).

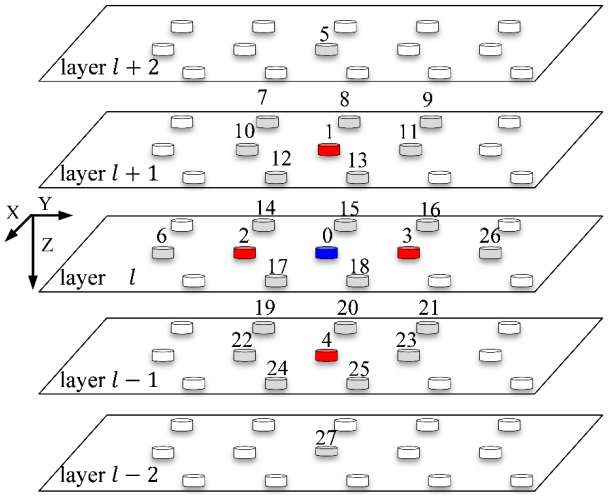

Fig. 2. Neighboring cells around the victim cell at position 0.

For each averaging in measuring conditional means, the number of samples used is typically huge. We write random input values (states) to all 800 million cells. Let us suppose we wish to measure the read value of a victim cell (at relative position 0) with a particular input state, say, pv_1, given a particular set of input state values on the four most-influencing neighboring cells, say, pv_0, pv_2, pv_1, pv_1 for relative cell positions 1, 2, 3, 4, respectively. So the question is: as we scan the written states of all 3D cells, how many times do we encounter the set of 5 cells with the particular written states of pv_1, pv_0, pv_2, pv_1, pv_1 at relative positions 0,1,2,3,4, respectively. The answer is about $(100 \times 8 \times 10^6)/10^3 = 800,000$, since the number of different combinations for the input values for the 5-cell group is 4^5, which is approximately 10^3. So the sample size is more than sufficient for statistical averaging. We do indeed observe that as averaging is computed with an increasing number of samples, the average value settles well before all 800,000 samples are collected.

With reference to the physical cell structure of a given 3D NAND, we first identify dominant interfering cells. Fig. 2 shows a collection of cells indexed from $j = 1$ to $j = 27$ around a victim cell positioned at $j = 0$.

A. Determining Dominant Interfering Cells

Fig. 3 shows the cell-to-cell interference obtained from the real data according to (9), normalized by $d_{min}/2$, where d_{min} is the minimum of the differences between mean voltage values of adjacent programming states, i.e., $\{pv_1 - pv_0, pv_2 - pv_1, pv_3 - pv_2\}$.

The victim cell's programmed level is set at $s = pv_0$ while the programmed state of the interfering cell is varied. The cell with index $j = 4$ at the $l - 1$ layer caused the greatest

Authorized licensed use limited to: Korea Advanced Inst of Science & Tech - KAIST. Downloaded on April 09,2021 at 01:42:54 UTC from IEEE Xplore. Restrictions apply.

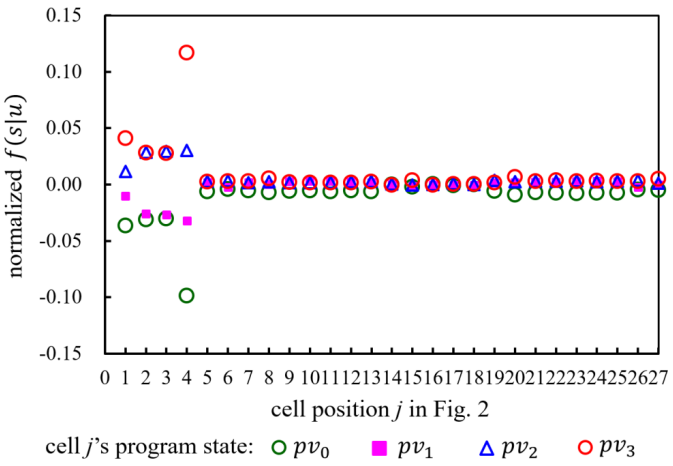

Fig. 3. Interference $f(s = pv_1|u)$ with u being the programmed state of a single neighboring cell at position j.

TABLE I
RELATIVE POSITIONS OF DOMINANT INTERFERING CELLS AROUND THE VICTIM CELL AT (k, m, l)

(x, y, z) coordinates	cell position index j	cell location
(k, m, l)	0	victim cell
$(k, m, l+1)$	1	same pipe as victime
$(k, m-1, l)$	2	same page as victime
$(k, m+1, l)$	3	same page as victime
$(k, m, l-1)$	4	same pipe as victime

interference to the victim cell. The cell with $j = 1$ is at the $l + 1$ layer above the victim cell. The victim cell ($j = 0$) and the cells with $j = 1, 4$ share the same pipe vertically. The cells with $j = 2$ and $j = 3$ are the closest adjacent cells to the victim cell in the Y direction, and they share the same logical page along with the victim cell as shown in Fig. 2. Since the interfering cells with $j = 2, 3$ are in the same page with the victim cell, no additional read latency is required to evaluate the programming states of these cells, which makes our method applicable to real-time scenarios. However, there is not a dominant interfering cell in the X direction that causes significant interference, as illustrated in Fig. 3. This is because for the particular 3D NAND used in this experiment, the cell structure was intentionally designed so that the distance between cells in the X direction is large compared to that along the Y direction. For example, in Fig. 2, the distance between the victim cell 0 and the neighboring cell 15 in the X direction is larger than the distance between cell 0 and cell 2 or 3 in the Y direction. Table I specifies the positions of four dominant interfering cells.

B. Comparing Interference Variances Estimated From Sample Means vs. Sample Variances

From (10), it is clear that the conditional variance associated with the samples $y(s|u, U^c)$ is

$$\sigma^2_{y(s|u,U^c)} = \sigma^2_{f(s|u,U^c)} + \sigma^2_{F_{\delta 0}(s|u,U^c)} + \sigma^2_{W(s)} + \sigma^2_N \quad (12)$$

where $\sigma^2_{f(s|u,U^c)}$ is the variance of interference due to varying input values of u^c, $\sigma^2_{F_{\delta 0}(s|u,U^c)}$ is due to the potential offsets in the given values of u as well as in all possible input values of U^c, while $\sigma^2_{W(s)}$ and σ^2_N are the variances of $W(s)$ and N, respectively. On the other hand, it is also clear that the variance of the read samples $y(s|U, U^c)$, or simply $y(s)$, reflecting the randomness in input values of all cells u and u^c is

$$\sigma^2_{y(s)} = \sigma^2_{f(s|U,U^c)} + \sigma^2_{F_{\delta 0}(s|U,U^c)} + \sigma^2_{W(s)} + \sigma^2_N \quad (13)$$

where $\sigma^2_{f(s|U,U^c)}$ is the interference variance due to input variations in both u and u^c, whereas $\sigma^2_{F_{\delta 0}(s|U,U^c)}$ is due to offsets in the input values of u and u^c. Now, think of taking the average of $\sigma^2_{y(s|u,U^c)}$ across all possible realizations of u values for a given set of interfering cell positions:

$$\overline{\sigma^2}_{y(s|u,U^c)} = \sum_u \sigma^2_{y(s|u,U^c)} P(u).$$

For example, with the four dominant interefering cells discussed above, there will be a total of $4^4 = 256$ different realizations for u. From (12) then, we can write:

$$\overline{\sigma^2}_{y(s|u,U^c)} = \sigma^2_{f(s|u,U^c)} + \overline{\sigma^2}_{F_{\delta 0}(s|u,U^c)} + \sigma^2_{W(s)} + \sigma^2_N \quad (14)$$

where $\sigma^2_{f(s|u,U^c)}$, $\sigma^2_{W(s)}$ and σ^2_N do not depend on the cell input values of u, and thus do not require averaging. $\overline{\sigma^2}_{F_{\delta 0}(s|u,U^c)}$ is the similarly defined average of $\sigma^2_{F_{\delta 0}(s|u,U^c)}$. It is also straightforward to show that

$$\overline{\sigma^2}_{F_{\delta 0}(s|u,U^c)} = \sigma^2_{F_{\delta 0}(s|U,U^c)}. \quad (15)$$

From (13), (14) and (15), we have

$$\sigma^2_{y(s)} - \overline{\sigma^2}_{y(s|u,U^c)} = \sigma^2_{f(s|U,U^c)} - \sigma^2_{f(s|u,U^c)}. \quad (16)$$

Given that $\sigma^2_{f(s|U,U^c)}$ arises due to input variations in both u and u^c while $\sigma^2_{f(s|u,U^c)}$ is due to varying inputs of just u^c (provided that the cell positions for u are fixed), it is not difficult to see that the difference $\sigma^2_{y(s)} - \overline{\sigma^2}_{y(s|u,U^c)}$ can serve as an estimate for the variance due only to u, i.e., the variance of $f(s|u)$ in (9) due to randomness in the input values of u only. Thus, we have

$$\sigma^2_{f(s|U)} \approx \sigma^2_{y(s)} - \overline{\sigma^2}_{y(s|u,U^c)}. \quad (17)$$

Note that the variance of $f(s|u)$ can also be obtained directly from the sample means as instructed by (9), i.e.,

$$\begin{aligned}\sigma^2_{f(s|U)} &= \sum_u f^2(s|u)P(u) \\ &= \frac{1}{|u|}\sum_u [\bar{y}(s) - \bar{y}(s|u, U^c)]^2 \\ &= \overline{[\bar{y}(s) - \bar{y}(s|u)]^2} \end{aligned} \quad (18)$$

where a uniform distribution is naturally assumed for the input values of u and $|u|$ is the number of possible combinations for them. We note that $f(s|U)$ is zero-mean.

Notice that of the conditioning variables, U^c is common and it can often be dropped to reduce notational complexity without compromising clarity. Thus, we will introduce additional shorthand notations: $y(s|u)$ for $y(s|u, U^c)$, $\sigma^2_{f(s|u)}$ for $\sigma^2_{f(s|u,U^c)}$, and $\sigma^2_{F_{\delta 0}(s|u)}$ for $\sigma^2_{F_{\delta 0}(s|u,U^c)}$.

Authorized licensed use limited to: Korea Advanced Inst of Science & Tech - KAIST. Downloaded on April 09,2021 at 01:42:54 UTC from IEEE Xplore. Restrictions apply.

TABLE II

INTERFERENCE VARIANCES ESTIMATED BASED ON SAMPLE MEANS VS. SAMPLE VARIANCES; AS A REFERENCE, WE HAVE $d_{min}^2/4 = 0.522$

cell indices for u	$\overline{[\bar{y}(s) - \bar{y}(s\|u)]^2}$	$\sigma^2_{y(s)} - \overline{\sigma^2}_{y(s\|u)}$
4	3.582E-03	3.680E-03
4,1	4.041E-03	4.192E-03
4,1,2,3	4.571E-03	4.717E-03

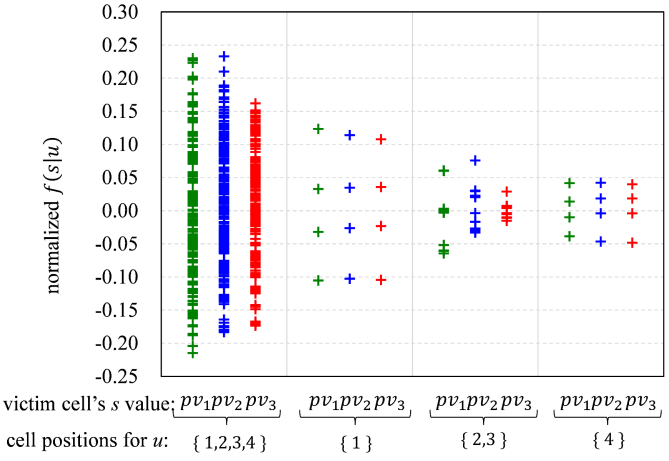

Fig. 4. Interference $f(s|u)$ for each victim cell input value s characterized by interfering cells u (cell positions and their input values) under HTDR degradation.

Table II shows $\sigma^2_{f(s|U)}$ estimated from the sample variances as in (17) versus that obtained from the sample means as in (18) for three different sets of cell positions for u: {4}, {4, 1} and {4, 1, 2, 3}. The results are very consistent between the two different routes of estimating the interference variance, which provides a cross-validation for the two methods. This consistency also gives a level of comfort for the proposed method.

C. Characterization of Interference in HTDR

The interference and overall noise variances of cells degraded by HTDR are characterized for different cell locations and cell input values for u, as shown in Figs. 4 and 5. HTDR is a kind of reliability stress where program/erase cycling stress and retention stress are performed sequentially in high temperature environments. The sum of these stresses are modeled to be exactly the same as the life cycle of the cell. The Y-axis scale of Fig. 4 is normalized by $d_{min}/2$. The mean-interference $f(s|u)$ is plotted for 4 different collections of cells for u, as well as different s values. Each point represents $f(s|u)$ corresponding to a victim cell's program state s and a specific interfering cell position/value (or set of cell positions and cell input values) u.

Consistent with Fig. 3, even for different victim cell input values s, we observe that for a single interfering cell, one at position $j = 4$ give rise to the greatest fluctuation in $f(s|u)$

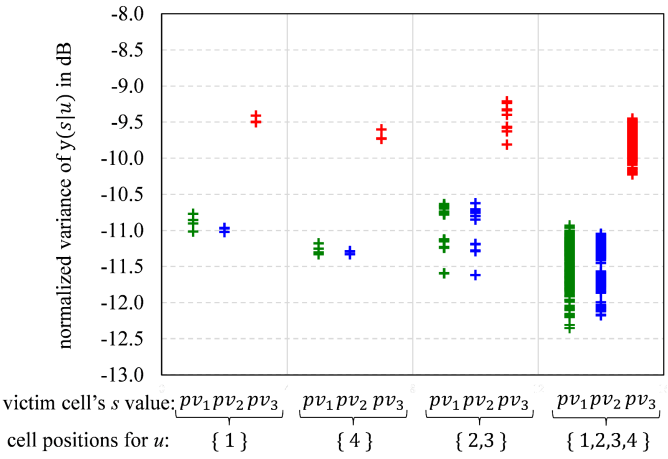

Fig. 5. Normalized $\sigma^2_{y(s|u)}$ values characterized by different u and s, when cells are degraded by HTDR.

as the cell input value for u changes (vertical variation within a given color). Different colors represent different program states s for the victim cell. The cell at $j = 4$ is programmed after the victim cell and is located right below the layer containing the victim cell, sharing the same pipe. The effect of the single interfering cell at $j = 1$ is smaller than that at $j = 4$. The cell at $j = 1$ is programmed ahead of the victim cell because it exists in the upper layer relative to the victim cell. However, cell $j = 1$ still affects the victim cell's programmed state because the F/Gs of cells sharing the same pipe are not isolated [19]. In contrast, we note that all cells in a 2D NAND are isolated from one another electrically. When considering the combined effect of cells at $j = 2, 3$, a smaller interference is induced compared to cell at $j = 4$ alone. We remark that the space between cell $j = 2$ (or cell $j = 3$) and the victim cell is filled with metallic materials instead of dielectric materials. Analyzing the electrical mechanism of these phenomena is beyond the scope of this article. When all four cells at $j = 4, 1, 2, 3$ change their values, the fluctuation in $f(s|u)$ is fairly large regardless of the victim cell state s. For a given color, the vertically represented fluctuation of the $f(s|u)$ value is a reflection of the mean-interference due to $4^4 = 256$ different combinations of the input values for the four interfering cells. It can be concluded that when it comes to interference, the programmed states and locations of the interfering cells make large difference whereas the victim cell's program value does not appear to be a significant factor.

Fig. 5 illustrates the variances of samples $y(s|u)$ characterized by different u and s. Again, a given color corresponds to a particular s value, the victim cell input. For a given color for a given set of interfering cell positions u, the vertical range represents a range of read voltage variances corresponding to different combinations of interfering cell input values. We remind the reader that the variance $\sigma^2_{y(s|u)}$ arises due to the s-dependent write noise $W(s)$, the system/RTN noise N and the residual interference coming from input value variations for cells in u^c as well as possible random offsets in actually written values of all cells relative to the intended programmed

Authorized licensed use limited to: Korea Advanced Inst of Science & Tech - KAIST. Downloaded on April 09,2021 at 01:42:54 UTC from IEEE Xplore. Restrictions apply.

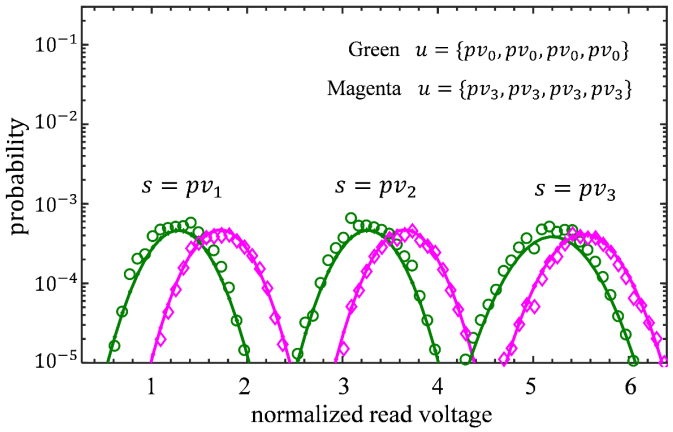

Fig. 6. For each value of s, green and magenta probability distributions for $y(s|u)$ correspond to two different sets of input values u, under four interfering cells. The markers are actual measured data points, whereas the solid lines are Gaussian models based on measured conditional means and variances.

states. The Y-axis represents $\sigma^2_{y(s|u)}$ normalized by $[d_{min}/2]^2$ and is plotted as $20\log[\sigma_{y(s|u)}/(d_{min}/2)]$ dB. In contrast to the interference plot of Fig. 4, the sample variance here depends highly on s, the state of the victim cell. In particular, the variance is considerably larger with $s = pv_3$ than with the other s values. For the case where all 4 dominant interfering cells are considered, the vertical range for a given s value is quite noticeable. The variance terms for $W(s)$ and N terms in (14) obviously do not contribute to this u-value dependence for the read voltage variance. The variance term $\sigma^2_{f(s|u)}$ cannot be a factor either in creating this vertical range of values, as it depends only U^c and thus on the chosen set of cell positions in u, not on the input values of u. Therefore, once the cell locations in u are fixed, the only term responsible for creating a range of different variance values in the read voltage with changing input values of u is the remaining term in (14), namely, the variance $\sigma^2_{F_{\delta 0}(s|u,U^c)}$ or $\sigma^2_{F_{\delta 0}(s|u)}$ in short. Given that all four dominant cells are captured in u, it is reasonable to further conjecture that the effect of the cell input offsets in u^c would be negligible. The only plausible conclusion here then is that the deviations of the actually written values of cells in u around the intended programming values are not entirely ignorable and that they depend on the intended states. These random deviations in turn have caused fluctuations in the amount of interference the dominant cells exert on the victim cell.

D. Read Value Distributions

Fig. 6 shows the measured sample probability distributions of $y(s|u)$ for the four significant interfering cells at positions 4, 1, 2 and 3. Here, the normalized voltage is the read voltage value normalized by $d_{min}/2$, where d_{min} is again the minimum difference between voltage values corresponding to adjacent programming states. In this experiment, all four interfering cells are programmed uniformly to either all pv_0 or all pv_3 for each of the three victim cell program values of pv_1 through pv_3. The individual marker points are actual observations while the solid curves are obtained based on

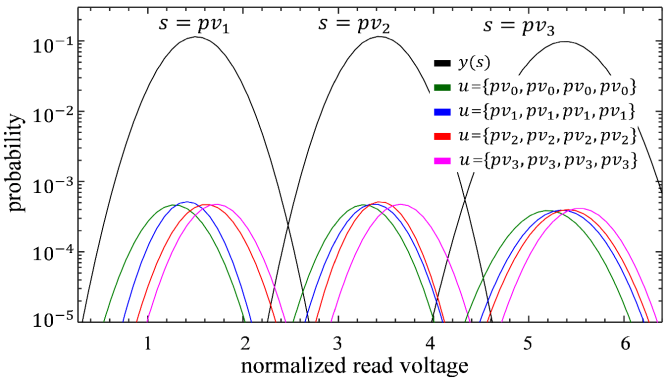

Fig. 7. Probability distributions like in Fig. 6 with different interfering cell input values. The overall distribution for $y(s)$ is shown for each of the three s values.

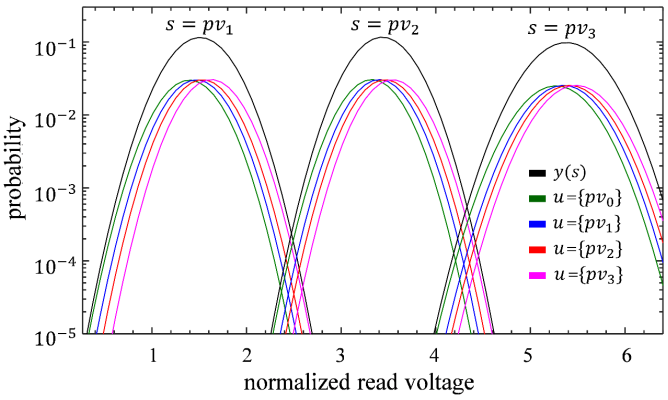

Fig. 8. Probability distributions of $y(s|u)$ with u corresponding to a single cell at position $j = 4$.

imposing Gaussian distributions withe the mean $f(s|u)$ obtained via (9) and the measured sample variance $\sigma^2_{y(s|u)}$. It can been seen that the Gaussian distribution gives a fairly accurate fit, down to the tails.

Fig. 7 provides similar probability distributions for different neighboring cell input values, this time omitting the actual observation data points for better legibility. Fig. 8 shows distributions under a single interfering cell at position 4.

Additionally, in Fig. 9 we observe the effect of step size variation in the ISPP writing process, without any reliability stress. The read voltage distributions for $y(s) = y(s|U, U^c)$ are shown for each s value, with the wider blue distribution corresponding to an increased ISPP step size by the factor 2.86. The solid curves represent Gaussian modeling based on mean and variance measurements, whereas the markers are directly measured data points. Notice that the probability distribution for $y(s)$ is a sum of the probability distributions for $y(s|u)$, which were observed to follow the Gaussian distribution closely. Consequently, the probability distribution for $y(s)$ cannot be Gaussian strictly. Nevertheless, it can be seen that the actual measured data points agree well with the Gaussian curves based on the overall means and variances for the original smaller step size. The agreement is not as good for the larger step size. This is due to the fact that with an increased step size the amount of uncertainty in

Authorized licensed use limited to: Korea Advanced Inst of Science & Tech - KAIST. Downloaded on April 09,2021 at 01:42:54 UTC from IEEE Xplore. Restrictions apply.

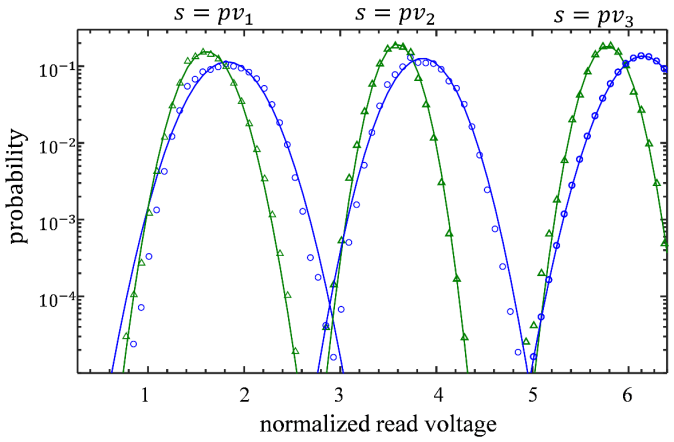

Fig. 9. Distributions of $y(s|U, U^c) = y(s)$ for two different incremental voltage pulse programming steps, without reliability stress, where U consists of four cells at positions {1,2,3,4}. Green and blue are for the smaller and larger steps, respectively. The markers are actual measured data points while the solid lines are Gaussian modeling based on overall means and variances.

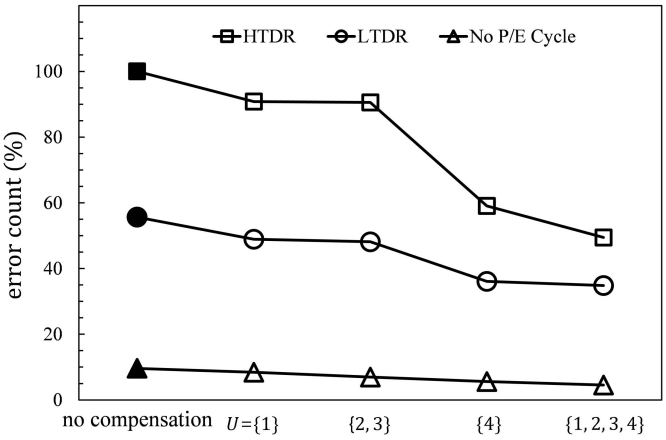

Fig. 11. Percentage reduction of errors with interference compensation under different reliability stress conditions. Compensation is based on an increasing number of known interfering cell input values.

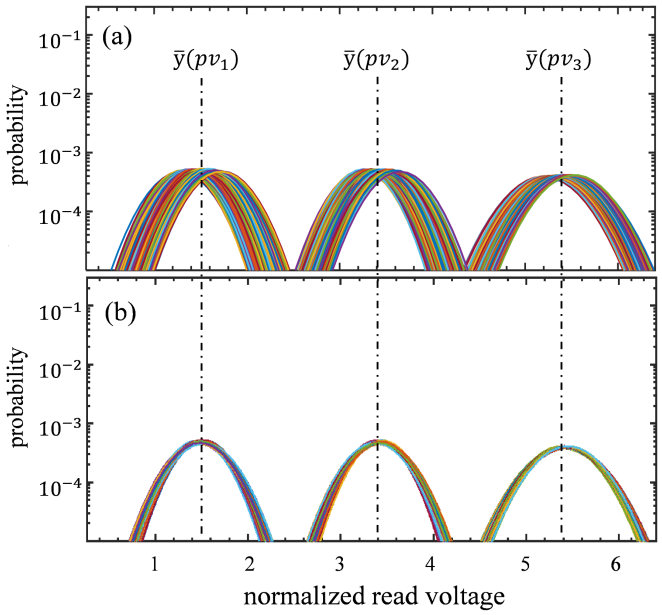

Fig. 10. Distributions of $y(s|u)$ before and after interference compensation under HTDR with interfering cells at {1,2,3,4}. (a) For each s value there are 256 curves corresponding to distinct input values for four interfering cells, (b) Same as (a) after each curve is compensated by interference $f(s|u)$. The overall means for $y(s = pv_i)$ are also shown.

the actual programmed cell input value [i.e., the variance of $W(s)$ in (13)] increases and that the distribution of $W(s)$ is distinctly non-Gaussian by nature [11].

III. Compensation of Inter-Cell Interference

Fig. 10(a) shows, for each value of s, all 256 distributions of $y(s|u)$ corresponding to different input values of u for the four dominant interfering cell positions. The probability distribution of $y(s)$ for a particular victim cell value s would reflect the accumulated effect of all 256 curves. The curves are obtained assuming Gaussian models based on the measured means and variances. A HDTR stress is applied. Recall from (12) that the mean of the distribution of $y(s|u)$ is $\overline{y}(s)+f(s|u)$ while the variance is given by (14). If the interference $f(s|u)$, which can be easily measured by (10), is removed from each distribution $y(s|u)$, then the means would become identical across all 256 realizations of u, for each given s. The variance $\sigma^2_{y(s|u)}$ would be potentially different across 256 distributions, however, due to the presence of the $\sigma^2_{F_{\delta 0}(s|u)}$ term, as argued above in relation to Fig. 5. Recall that $\sigma^2_{f(s|u)}$, another term in (14), depends only on the cell positions of u, not on the cell input values of u. Fig. 10(b) shows the interference-compensated distributions of $y(s|u)$ for each s value. It can be seen that the 256 curves indeed do not exactly fall on top of one another, showing the effect of small variations due to $\sigma^2_{F_{\delta 0}(s|u)}$ across different realizations of u. The difference in overall variances of the accumulated distributions of $y(s)$ before and after the interference compensation represents the reduction of uncertainty in the read value of a cell made possible by observing the cell values of the four interfering cells and successfully estimating the amount of interference caused by these cells.

Assuming that the distributions of $y(s)$ both before and after the interference compensation can be approximated by Gaussian, the reduced variance can be translated into an improved error rate for threshold detection. Fig. 11 shows the percentage reduction in the number of bit errors as interference compensation is done using an increasing number of known neighboring cell values, i.e., by subtracting $f(s|u)$ from the read value y as the range of u is increased from only cell 1 to eventually all four cells {1,2,3,4}. Three cases of HTDR, LTDR and no P/E cycles are covered. The case where all cells are stressed by HDTR with no compensation is set for a 100% error count measure. As u collects more interfering cells, the errors are reduced for all cases. Here, LTDR is the reliability stress with program/erase cycling and retention stresses performed under the room temperature. Compensation based on all 4 cells gives a 50% reduction under the HDTR stress, while using only the cells at $j = 2, 3$ for inter-ference compensation reduces the error count by 10%. The

Authorized licensed use limited to: Korea Advanced Inst of Science & Tech - KAIST. Downloaded on April 09,2021 at 01:42:54 UTC from IEEE Xplore. Restrictions apply.

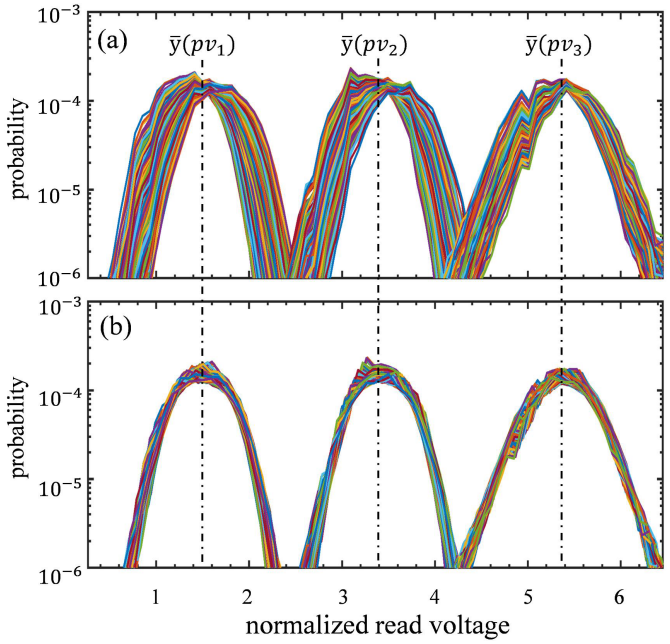

Fig. 12. Distributions of $y(s|u)$ before and after interference compensation, in (a) and (b), respectively, just as Fig. 10 but based on actually measured read voltage distributions.

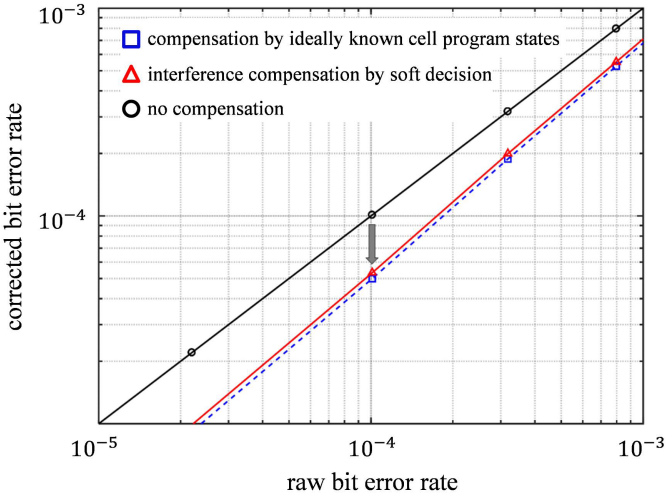

Fig. 13. bit error rates simulated with interference compensation.

latter approach would not require any read latency (and thus real-time operation is possible) as the interfering cells at $j = 2, 3$ would be on the same page as the victim cell. Utilizing the read-level decisions for cells at 1, 4 requires system latency as these cell are in different pages from that containing the victim cell.

Fig. 12 shows the measured distributions (as opposed to Gaussian modeling based on measured means and variances) of the read values before and after interference compensation. Again, variance reduction is evident after interference compensation.

Fig. 13 shows simulation results based on Gaussian noise modeling, allowing potential errors in estimating the program states of all four interfering cells. When all random noise sources are assumed Gaussian and the effect of interference from any other cells are ignored, the conditional read value of (10) can be written as

$$y(s|u) = s + f(s|u) + N$$

where u represents the input states of the four most-interfering cells and N models the overall Gaussian noise. Now, as stated in Section I.C, the interference value $f(s|u)$ can be learned and prestored in the controller in the form of a look-up table for different input values s and u for the particular relative cell positions, prior to the time of read compensation. During simulation, we use hard detected values for s and u to access the stored interference value from the look-up table, and then subtract this interference from $y(s|u)$. The interference-compensated $y(s|u)$ is then compared against the threshold values to make a final hard decision. A given raw bit error rate is reflected in occasional mis-compensation of the interference $f(s|u)$ as the estimated (hard-detected) values of s and u would not be entirely correct. The simulation results, however, indicate that the effect of mis-compensation is negligible in the range of the raw bit error rates considered. In fact, for the raw error rate of 1.0×10^{-4}, interference compensation improves the error rate to around 5×10^{-5}, signifying a 50% error count reduction.

Note that the interfering cell right above the victim cell is programmed before the victim cell itself. Thus, the read voltage shift of the victim cell caused by this cell cannot be explained by capacitive interference. This phenomenon was presumed to be due to (i) the physically un-isolated F/G in the Z-direction (along the pipe), and (ii) the charge concentration difference between this interfering cell and the victim cell [20]. Among all interfering cells, the cell underneath the victim cell induced the greatest amount of interference to the victim cell, causing the least negative interference when it is programmed to pv_0 and the greatest positive interference when programmed to pv_3. Although the interference from two other cells in the same layer is lower than the first two, no additional read latency is required to determine their programmed states because these two cells are located in the same page with the victim cell. As shown above, a fair amount of errors can be corrected if interference coming from all four cells can be compensated. But compensating interference from cells in different pages from one containing the victim cell gives rise to another challenge in terms of causing extra read latency. Finding a system-level solution to handle latency would be a fruitful research direction.

IV. Conclusion

In this study, the dominant interfering cells in 3D NAND were identified and characterized via a convenient measurement method based on collecting sample means and sample variances of cell read values corresponding to random input data. The mean and variance of interference coming from any target set of cells could be estimated accurately based on this method. It has been shown that two upper/lower cells along the same pipe as well as two right/left cells along the same bit line are responsible for most of the interference

Authorized licensed use limited to: Korea Advanced Inst of Science & Tech - KAIST. Downloaded on April 09,2021 at 01:42:54 UTC from IEEE Xplore. Restrictions apply.

a victim cell experiences. The most severe interference that come from the two upper/lower cells cannot be viewed as capacitive interference and is different in nature from that observed in 2D NAND media. Analysis and measurements indicate that a significant number of errors can be corrected if interference coming from all four cells can be compensated.

REFERENCES

[1] G. Wong, "SSD market overview," in *Inside Solid State Drives (SSDs)*. Springer, 2013, pp. 1–17.

[2] S. Lee, "Scaling challenges in NAND flash device toward 10 nm technology," in *Proc. 4th IEEE Int. Memory Workshop*, May 2012, pp. 1–4.

[3] H. Tanaka *et al.*, "Bit cost scalable technology with punch and plug process for ultra high density flash memory," in *Proc. IEEE Symp. VLSI Technol.*, Jun. 2007, pp. 14–15.

[4] E.-S. Choi and S.-K. Park, "Device considerations for high density and highly reliable 3D NAND flash cell in near future," in *IEDM Tech. Dig.*, Dec. 2012, pp. 4–9.

[5] K. Parat and C. Dennison, "A floating gate based 3D NAND technology with CMOS under array," in *IEDM Tech. Dig.*, Dec. 2015, p. 3.

[6] C. Huang, F. Liu, Q. Wang, and Z. Huo, "A small ripple program voltage generator without high-voltage regulator for 3D NAND flash," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 67, no. 6, pp. 1049–1053, Jun. 2020.

[7] R. Katsumata *et al.*, "Pipe-shaped BiCS flash memory with 16 stacked layers and multi-level-cell operation for ultra high density storage devices," in *Proc. Symp. VLSI Technol.*, 2009, pp. 136–137.

[8] J. Jang *et al.*, "Vertical cell array using TCAT (terabit cell array transistor) technology for ultra high density NAND flash memory," in *Proc. Symp. VLSI Technol.*, 2009, pp. 192–193.

[9] A. Goda and K. Parat, "Scaling directions for 2D and 3D NAND cells," in *IEDM Tech. Dig.*, Dec. 2012, pp. 1–2.

[10] J. Moon, J. No, S. Lee, S. Kim, S. Choi, and Y. Song, "Statistical characterization of noise and interference in NAND flash memory," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 60, no. 8, pp. 2153–2164, Aug. 2013.

[11] Y. Cai, O. Mutlu, E. F. Haratsch, and K. Mai, "Program interference in MLC NAND flash memory: Characterization, modeling, and mitigation," in *Proc. IEEE 31st Int. Conf. Comput. Design (ICCD)*, Oct. 2013, pp. 123–130.

[12] C. R. Lanka, "Weight consistency matrix framework for non-binary LDPC code optimization," Ph.D. dissertation, Univ. California, Los Angeles, Los Angeles, CA, USA, 2016.

[13] L. Qiao, H. Wu, D. Wei, and S. Wang, "A joint decoding strategy of non-binary LDPC codes based on retention error characteristics for MLC NAND flash memories," in *Proc. 6th Int. Conf. Instrum. Meas., Comput., Commun. Control (IMCCC)*, Jul. 2016, pp. 183–188.

[14] Y. Cai, E. F. Haratsch, O. Mutlu, and K. Mai, "Threshold voltage distribution in MLC NAND flash memory: Characterization, analysis and modeling," in *Proc. Design, Automat. Test Eur. Conf. Exhib. (DATE)*, 2013, pp. 1285–1290.

[15] T. Kim, G. Kong, X. Weiya, and S. Choi, "Cell-to-cell interference compensation schemes using reduced symbol pattern of interfering cells for MLC NAND flash memory," *IEEE Trans. Magn.*, vol. 49, no. 6, pp. 2569–2573, Jun. 2013.

[16] 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.

[17] C. Sun, K. Miyaji, K. Johguchi, and K. Takeuchi, "A high performance and energy-efficient cold data eviction algorithm for 3D-TSV hybrid ReRAM/MLC NAND SSD," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 2, pp. 382–392, Feb. 2014.

[18] E. Nowak *et al.*, "Intrinsic fluctuations in vertical NAND flash memories," in *Proc. Symp. VLSI Technol. (VLSIT)*, Jun. 2012, pp. 21–22.

[19] C. Kang *et al.*, "Effects of lateral charge spreading on the reliability of TANOS (TaN/AlO/SiN/Oxide/Si) NAND flash memory," in *Proc. 45th Annu. IEEE Int. Rel. Phys. Symp.*, Apr. 2007, pp. 167–170.

[20] H.-J. Kang *et al.*, "Comprehensive analysis of retention characteristics in 3-D NAND flash memory cells with tube-type poly-Si channel structure," in *Proc. Symp. VLSI Technol. (VLSI Technology)*, Jun. 2015, pp. T182–T183.

Suk Kwang Park received the B.S. degree in physics from Dong-A University, Busan, South Korea, in 1999, the M.S. degree in physics from Pusan National University in 2002, and the Ph.D. degree from the School of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2019. Since 2002, he has been with SK hynix. From 2002 to 2005, he analyzed and enhanced the chip characteristics and reliability in DRAM Research and Development Division. He has been with the NAND Flash Development Division since 2005. He participated in efforts to realize the first commercial 2D TLC NAND in 2009, with enhancing the reliability and chip performance. He has focused on the process integration of the first 4D NAND stacked high. He studies the channel characterizations, signal processing, and the coding theory of NAND flash memory. His research interests include concentrate on how to reduce data error rates and how to improve the performance related to SSDs and other storage systems for the data center and the cloud computing.

Jaekyun Moon (Fellow, IEEE) received the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, USA. From 1990 to 2009, he was with the faculty of the Department of Electrical and Computer Engineering, University of Minnesota, Twin Cities. From 2004 to 2007, he consulted as a Chief Scientist for DSPG, Inc. He was also a Chief Technology Officer with Link-A-Media Devices Corporation. He is currently a Professor of electrical engineering with the Korea Advanced Institute of Science and Technology (KAIST). His research interests include channel characterization, signal processing, and coding for data storage and digital communication. He received the McKnight Land-Grant Professorship from the University of Minnesota. He received the IBM Faculty Development Awards and 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 a 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. 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 the IEEE TRANSACTIONS ON MAGNETICS in the area of signal processing and coding from 2001 to 2006.

Authorized licensed use limited to: Korea Advanced Inst of Science & Tech - KAIST. Downloaded on April 09,2021 at 01:42:54 UTC from IEEE Xplore. Restrictions apply.