

FlashDefibrillator: A Data Recovery Technique for Retention Failures in NAND Flash Memory

Jaeyong Jeong, Youngsun Song, and Jihong Kim

Department of Computer Science and Engineering, Seoul National University, Korea

Email: {jyjeong, yunsong, jihong}@davinci.snu.ac.kr

Abstract—Although NAND flash memory is known as a non-volatile memory device, the non-volatility of the data stored in the NAND flash memory is guaranteed only for a specified retention time. Since the NAND retention time assumes specific operation conditions, when the NAND flash memory is exposed to an abnormal environment beyond the specified operation conditions, stored data cannot be reliably retrieved due to retention failures. In this paper, we propose a novel data recovery technique, called FlashDefibrillator (FD), for recovering retention failures in recent NAND flash memory. By reversely exploiting charge-transient behavior observed in recent 20-nm node (or below) NAND flash memory, FD can identify retention-failed cells in a progressive fashion using a novel selective error-correction procedure. FD repeatedly applies the selective error-correction procedure until retention failures are fully recovered. Our measurement results with recent 20-nm node NAND chips show that FD outperforms the existing recovery technique in both the data recovery speed and the data recovery capability. FD can recover retention failures up to 23 times faster over the existing data recovery technique. Furthermore, FD can successfully recover severely retention-failed data (such as ones experienced eight times longer retention times than the retention-time specification) which were not recoverable with the existing technique.

I. INTRODUCTION

NAND flash memory is a non-volatile memory device which can retain stored data even when power is turned off. Since NAND flash memory stores data as quantities of charges held on floating gates (that are electrically isolated by insulating layers), in theory, NAND flash memory can permanently store its data without a power source if the insulating layers work perfectly. However, actual NAND cells are limited in their data retention capability because various defects in the insulating layers occur during program/erase (P/E) operations. These defects in the NAND cells make charges in the floating gate loosened, thus guaranteeing the integrity of stored data only up to a finite retention time [1]. Since the probability of charge loss due to defects has an exponential dependence on temperature [2], the NAND retention time is specified under a specific operating temperature. For example, NAND flash memory for client-class applications is often required to retain its stored data for at least 1 year at 25 °C [3].

If NAND flash memory is used beyond the specified retention time, the data stored in the NAND flash memory may not be correctly retrieved because of excessive retention errors. For example, when NAND flash memory is left for more than two times longer than the specified retention time, retention failures may occur, losing the stored data. Moreover, since the NAND retention time decreases exponentially as temperature rises [4], an increase in temperature can significantly degrade the NAND retention capability. For example, when temperature rises to

70 °C, the specified NAND retention time of 1 year (at 25 °C) may be reduced to only 32 hours¹. Furthermore, the retention-failure problem can be a more serious technical issue when more aggressive flash-optimization techniques (e.g., [5][6]) are widely employed. Since these flash optimization techniques aggressively reduce the NAND retention capability during run time for higher NAND performance [5] or longer NAND endurance [6], retention errors are likely to increase. Thus, there is a strong demand for *efficient on-line* data recovery techniques for retention failures in NAND flash memory.

In order to deal with the NAND retention-failure problem, several data recovery techniques such as the retention failure recovery (**RFR**) technique [7] and the data retention-error recovery pulse (**DRRP**) technique [8] have been proposed. However, since **RFR** requires to heat NAND chips, it can be used only as an *off-line* recovery solution. Although **DRRP** can be implemented as an on-line recovery solution, it is quite limited because its recovering process is very slow and its recovery capability is rather restricted for recent 20-nm node (or below) NAND flash memory.

In this paper, we propose an efficient *on-line* data recovery technique, called *FlashDefibrillator* (**FD**), which can be effective for recent NAND flash memory. The proposed **FD** technique is motivated by our observations on the characteristics of retention-failed NAND cells in recent 20-nm node NAND flash memory. The key finding is that when read operations are repeated, highly-damaged cells (that probably contributed to retention failures) are more likely to experience abnormal charge-transient behavior (e.g., random charge fluctuation [9] or charge detrapping [10]). Since the abnormal charge-transient behavior of NAND cells (under repeated reads) were rarely observed in 3x-nm NAND flash memory, the existing technique such as **DRRP** (which was developed for 3x-nm NAND flash memory) cannot adequately handle retention errors from this new charge movement phenomenon. The proposed **FD** takes this behavior (as well as retention loss) into account of recovering retention-failed cells, thus resulting in a more efficient on-line data recovery solution.

The proposed **FD** technique consists of two main steps, a diagnostic step and a post-processing step. In the diagnostic step of **FD**, as done in **DRRP** [8], a sequence of diagnostic pulses (i.e., effectively read operations) is applied to NAND cells. The main goal of the diagnostic step is to recharge retention-loss cells so that these cells can be read at the correct state. Since diagnostic pulses add extra charges to NAND cells, a threshold voltage (V_{th}) distribution tends to shift to the right after the diagnostic step, thus making some of retention-failed cells be recovered.

¹This estimation is based on the Arrhenius equation used to calculate thermal acceleration factors for NAND devices [4].

In the following post-processing step, **FD** identifies retention-failed cells as ones whose V_{th} states were shifted to the right. This heuristic, as used in **DRRP** [8], reversely exploits the retention-loss mechanism in that highly retention-loss cells are more likely to be recharged with a low voltage. Furthermore, in order to avoid the negative effect of the abnormal charge movements on the **FD**'s recovery capability, **FD** identifies retention-failed data in a progressive fashion using a selective error-correction procedure. The selective error-correction procedure, which identifies retention-recovered cells as early as possible, is based on a simple but effective heuristic: *If a NAND cell c is shifted to a higher V_{th} state after the diagnostic step, the cell c is identified as a retention-failed cell and its V_{th} state is corrected to the higher V_{th} state.* As soon as the cell c is corrected by our heuristic, it is no longer considered in the remaining steps of **FD**. Although our heuristic seems to be very simple, it is quite effective in handling abnormal charge movements (after the diagnostic step) observed in recent NAND flash memory, thus significantly improving **FD**'s data recovery capability over **DRRP**. (See Section V.) The result of the post-processing step is stored to an internal buffer. If bit errors of the buffered data can be fully corrected by ECC, **FD** completes its recovery procedure and the fully recovered data are rewritten to a free page. Otherwise, two **FD** steps are repeated to the buffered data. After a pre-set maximum iteration count is reached, **FD** stops the recovery procedure.

Our experimental results with recent 20-nm node NAND chips show that **FD** outperforms **DRRP** in both the data recovery speed and data recovery capability. **FD** can recover retention failures up to 23x faster over **DRRP**. Our experimental results also show that **FD** can reduce the worst-case raw bit error rate (RBER) by up to 55% over **DRRP**. Based on the RBER measurement results over varying retention times, we can conclude that **FD** can recover retention-failed data which have experienced up to 8x longer retention times than the NAND retention-time specification.

The rest of the paper is organized as follows. In Section II, we explain the existing retention-error management policy for NAND flash memory. Section III presents the motivation of our work. In Section IV, we describe the implementation detail of our proposed **FD** technique based on the charge movement model. Experimental results are given in Section V. We conclude with summaries and future work in Section VI.

II. LIMITATIONS OF THE EXISTING RETENTION-ERROR MANAGEMENT POLICY

Since NAND flash memory stores data by electrically changing quantities of charges in the floating gate, NAND cells have different V_{th} 's depending on bit information. The stored data are read by sensing the V_{th} positions of each NAND cell. Fig. 1 illustrates an example of V_{th} distributions for MLC NAND flash memory which stores two bits per cell by using four distinct V_{th} states, i.e., E , $P1$, $P2$ and $P3$. Four V_{th} states are distinguished by three read reference voltages, i.e., R_{P1} , R_{P2} and R_{P3} . The width W_{P_i} of a V_{th} distribution and the V_{th} gap M_{P_i} between two adjacent states are designed to guarantee the performance and retention requirements.

When NAND flash memory is programmed and left for a long time, retention errors may occur due to retention loss. Fig. 2(a) illustrates an example of V_{th} -distribution changes after 3K P/E cycling and a 1-year retention time. Since the

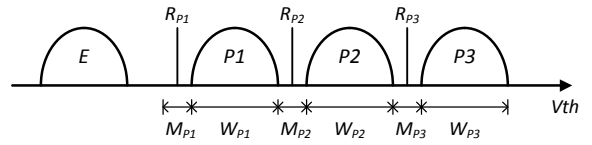
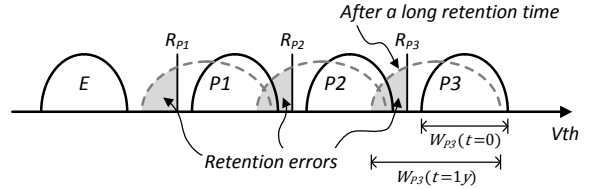
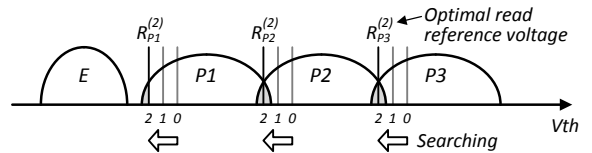


Fig. 1. An example of V_{th} distributions for MLC NAND flash memory and primary design parameters.



(a) An example of V_{th} -distribution changes due to retention loss.



(b) An example of a read retry procedure to find the optimal read reference voltage.

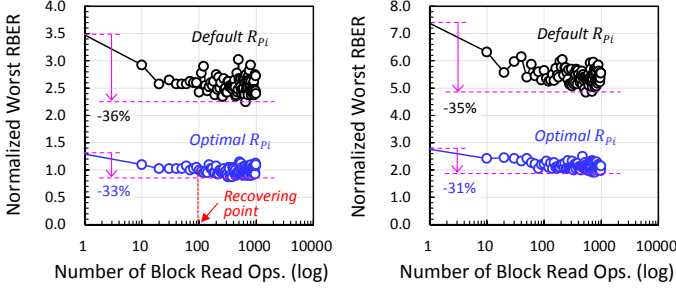
Fig. 2. Examples of a NAND retention-error management policy.

overall V_{th} distributions shift down after a long retention time, a lot of bit errors may occur when the initial read reference voltages (R_{P_i} 's) are used in a read operation. If the number of bit errors exceeds the error-correction capability (e.g., 40 bits per 1 KB for an MLC device [11]) of ECC, a read-retry procedure is invoked to manage retention errors [12]. Read retry is a searching algorithm for the optimal read reference voltage, which iteratively performs read-and-check routines with different read reference voltages until all the bit errors are corrected. For example, as shown in Fig. 2(b), read retry was performed two times to find the optimal read reference voltages ($R_{P_i}^{(2)}$'s).

However, if NAND flash memory is left beyond the specified retention time, the stored data cannot be retrieved even with read retry. This is because read retry cannot *actively* reduce bit errors, but *just find* the optimal read reference voltages where the number of bit errors can be minimized for given V_{th} distributions. Since retention loss tends to cause shifting the overall V_{th} distributions as well as widening W_{P_i} 's, after a long retention time, two adjacent V_{th} distributions may overlap each other. For example, as shown in Fig. 2(a), since $W_{P3}(t=1y)$ after 1-year retention time gets wider than $W_{P3}(t=0)$, the $P3$ state is overlapped with the $P2$ state. As a result, remaining bit errors cannot be further reduced by read retry as shown in Fig. 2(b). If there are more bit errors than the error-correction capability at the optimal read reference voltages, there is no way of retrieving the stored data with the existing error management policy.

III. MOTIVATION

Before we describe the proposed **FD** technique in detail, we present our evaluation results of an existing data recovery technique for recovering retention failed cells in recent 20-nm node NAND flash memory. For our evaluation, we used the data retention-error recovery pulse (**DRRP**) technique [8], which was considered as one of the most effective data recovery techniques for 3x-nm NAND chips. As will be discussed



(a) W-RBER variations in the 2-year retention time condition. (b) W-RBER variations in the 8-year retention time condition.

Fig. 3. Normalized worst-case RBER (W-RBER) variations over varying numbers of read operations under **DRRP**.

below, our evaluation results strongly suggest a need for better data recovery techniques for recent 20-nm node (or below) NAND flash memory, which was the main motivation for developing our proposed **FD** technique.

In order to recover retention-failed cells, **DRRP** repeatedly applies weak-stress pulses (e.g., 3 V [8]) to retention-failed cells so that the V_{th} 's of retention-failed cells can be recovered to their original V_{th} state. Measurement results with 3x-nm node NAND chips showed that **DRRP** could reduce the RBER of severely retention-failed cells (who experienced 3K P/E cycling and 3 days' baking at 85 °C) by 56%, on average, after applying 500 weak-stress pulses [8].

However, our measurement results show that the effectiveness of **DRRP** as an *on-line* recovery solution is quite limited because its data recovery process is very slow for recent 20-nm node (more advanced technology over 3x-nm node by about two generations) NAND flash memory. Since applying a weak-stress pulse is not allowed in our test environment, we used *read operations* (which can apply the read voltage of about 6 V) instead of the weak-stress pulse. Fig. 3(a) shows worst-case RBER (i.e., the RBER of a 1-KB sector which has the highest number of bit errors) variations over different numbers of read operations after 3K P/E cycling and 2-year retention times. The measured RBERs were normalized over the maximum error-correction capability of ECC. We denote the normalized worst-case RBER by W-RBER. When the default R_{Pi} 's were used, **DRRP** could reduce W-RBER by 36%. However, it could not lower W-RBER below 1.00. On the other hand, when the optimal R_{Pi} 's were used, **DRRP** could reduce W-RBER below 1.00. However, this reduction was reached after 100 read operations. If the average page read time is 100 μs , for example, it takes about 10 *ms* for each NAND page to be recovered, which is too slow to be employed as an on-line run-time technique.

Moreover, the data recovery capability of **DRRP** is quite restricted in recent NAND flash memory. Fig. 3(b) shows W-RBER changes with varying numbers of read operations after 8-year retention times (8x longer than the specified retention time). When the optimal R_{Pi} 's were used, **DRRP** could reduce W-RBER by up to 31%, however, W-RBER was not reduced below 1.00 until 1000 read operations. In the 4-year retention case, **DRRP** still could not reduce W-RBER below 1.00. These measurement results show that **DRRP** can recover retention-failed data which experienced up to 2x longer retention time than the specified retention time.

Our evaluation results show that **DRRP** is less effective with

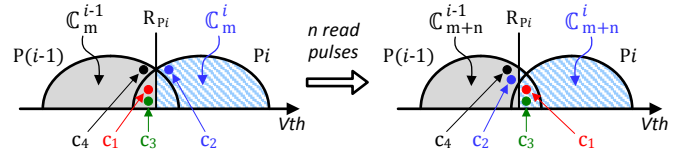


Fig. 4. An example of charge movement after n read pulse applications.

recent 20-nm NAND flash memory in recovering retention-failed data and its recovering speed is very slow. Our main goal was to improve **DRRP** so that it can be as effective with 20-nm NAND chips as with 3x-nm NAND chips while its recovering speed is fast enough so that it can be used as an on-line run-time solution.

IV. DESIGN AND IMPLEMENTATION OF FLASHDEFIBRILLATOR

In this section, we describe a charge movement model which can capture abnormal charge-transient behavior observed in recent 20-nm node (or below) NAND flash memory. Based on the charge movement model, we present a selective error-correction procedure and the implementation of our proposed **FD** technique in detail.

A. Charge Movement Model

Since applying multiple read pulses can partially recharge retention-loss cells, V_{th} 's of these cells can shift to the right [8]. On the other hand, it is reported that as a side-effect of recent advanced NAND technologies, another type of charge loss may occur due to multiple read pulses [10] so that V_{th} 's of some highly-damaged cells can shift to the left. If these abnormal charge-loss components are not negligible, the effectiveness of the recharging process can be substantially cancelled. Furthermore, it is also reported that in recent advanced NAND cells, V_{th} 's of some weak cells may randomly fluctuate because several charges are periodically trapped and detrapped due to the random telegraph noise (RTN) effect [9]. These randomly-fluctuated components may negatively affect the recharging process.

In order to understand how read pulses affect NAND cells, we built a simple charge movement (*CM*) model. We denote \mathbb{C}_m^i as a set of cells that are read as the i^{th} state after the m^{th} read pulse. For example, in Fig. 4, $\mathbb{C}_m^{i-1} = \{c_1, c_3, c_4, \dots\}$ while $\mathbb{C}_m^i = \{c_2, \dots\}$. After the n read pulses are applied, if the read value of a cell c_1 changes from $P(i-1)$ to P_i (for example, because of recharging), we say the cell c_1 belongs to the set $\mathbb{C}^{(i-1) \rightarrow i}$. That is,

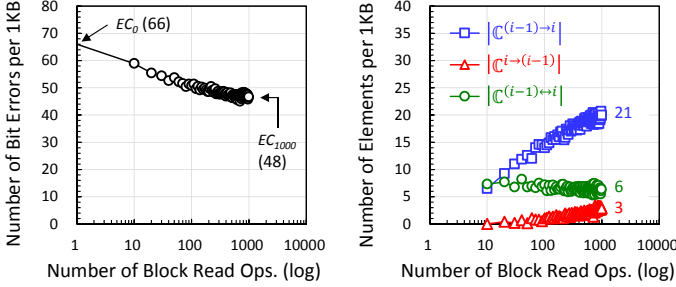
$$c_1 \in \mathbb{C}_m^{i-1} \cap \mathbb{C}_{m+n}^i = \mathbb{C}^{(i-1) \rightarrow i}. \quad (1)$$

On the other hand, if the read value of a cell c_2 changes from P_i to $P(i-1)$ (for example, because of charge detrapping), we say the cell c_2 belongs to the set $\mathbb{C}^{i \rightarrow (i-1)}$. That is,

$$c_2 \in \mathbb{C}_m^i \cap \mathbb{C}_{m+n}^{i-1} = \mathbb{C}^{i \rightarrow (i-1)}. \quad (2)$$

Finally, if the read value of a cell c_3 periodically changes between $P(i-1)$ and P_i (for example, because of random charge fluctuation), we say the cell c_3 belongs to the set $\mathbb{C}^{(i-1) \leftrightarrow i}$. That is,

$$\begin{aligned} c_3 &\in \mathbb{C}_m^{i-1} \cap \mathbb{C}_{m+n}^i \cap \mathbb{C}_{m+2n}^{i-1} \cap \dots \\ &= \left[\bigcap_{k=0} \mathbb{C}_{m+2k \cdot n}^{i-1} \right] \cap \left[\bigcap_{k=0} \mathbb{C}_{m+(2k+1) \cdot n}^i \right] = \mathbb{C}^{(i-1) \leftrightarrow i}. \quad (3) \end{aligned}$$



(a) Variations of the total bit-error count over varying the numbers of read operations.

(b) Variations of the numbers of each CM element over varying the numbers of read operations.

Fig. 5. Measurement results for tracing the CM -component changes in response to multiple read pulses.

After applying the m^{th} read pulse, since the number EC_m^{DRRP} of bit errors under **DRRP** decreases by the number of recharged cells while it increases by the number of additionally detrapped cells, EC_m^{DRRP} can be expressed as follows:

$$EC_m^{DRRP} = EC_0 - |\mathbb{C}^{(i-1) \rightarrow i}| + |\mathbb{C}^{i \rightarrow (i-1)}|, \quad (4)$$

where EC_0 is the initial number of bit errors before applying read pulses. This estimation is based on the assumption that the V_{th} states of the upper-tail cells (e.g, a cell c_4 in Fig. 4) in a widened V_{th} distribution (due to retention loss) rarely change from $P(i-1)$ to P_i after read pulse applications. When retention loss occurs, a V_{th} distribution gets wider as shown in Fig. 2(a), which reflects that the lower-tail cells are more likely to lose charges over the upper-tail cells. As a result, the upper-tail cells have a much lower probability to be recharged over the lower-tail cells so that their effect on the error-correction process can be ignored. Moreover, it is not necessary to include the number $|\mathbb{C}^{(i-1) \leftrightarrow i}|$ of randomly-fluctuated cells in EC_m^{DRRP} because **DRRP** does not distinguish the set $\mathbb{C}^{(i-1) \leftrightarrow i}$ from the set $\mathbb{C}^{(i-1) \rightarrow i}$ or the set $\mathbb{C}^{i \rightarrow (i-1)}$.

In order to trace the overall trend of CM -element changes in response to read pulses, we measured the average number of each CM element (per 1-KB unit) every ten read pulses. Fig. 5(a) shows EC_m^{DRRP} variations over varying numbers (e.g., $m = 0 \sim 1000$) of read operations after 3K P/E cycling and the 8-year (equivalent) retention times. In this example, EC_0 was 66 while EC_{1000} after 1,000 read pulses is reduced to 48. Fig. 5(b) shows measurement results for each CM element, which can explain the cause of retention-error reductions as shown in Fig. 5(a). As read operations are repeated, $|\mathbb{C}^{(i-1) \rightarrow i}|$ grows rapidly in the early stage (i.e., ~ 100 read pulses) but slowly at the end. (Note that the x -axis of Fig. 5(b) is a log scale.) On the other hand, $|\mathbb{C}^{i \rightarrow (i-1)}|$ grows slowly from beginning to the end. Since the differences between $|\mathbb{C}^{(i-1) \rightarrow i}|$ and $|\mathbb{C}^{i \rightarrow (i-1)}|$ are nearly saturated after one thousand read pulses, further read pulses have little effect on reducing EC_m^{DRRP} . Measurement result of the total bit-error count (e.g., $EC_{1000}^{DRRP} = 48$) almost matched the estimation (i.e., $66 - 27 + 9 = 48$) from Equation 4. An interesting observation is that $|\mathbb{C}^{(i-1) \leftrightarrow i}|$ starts with non-zero counts which is comparable with $|\mathbb{C}^{(i-1) \rightarrow i}|$ in the early stage. However, since **DRRP** expects only $\mathbb{C}^{(i-1) \rightarrow i}$ elements after multiple read pulses, $\mathbb{C}^{(i-1) \leftrightarrow i}$ elements (as well as $\mathbb{C}^{i \rightarrow (i-1)}$ elements) are not considered in its error-correcting process.

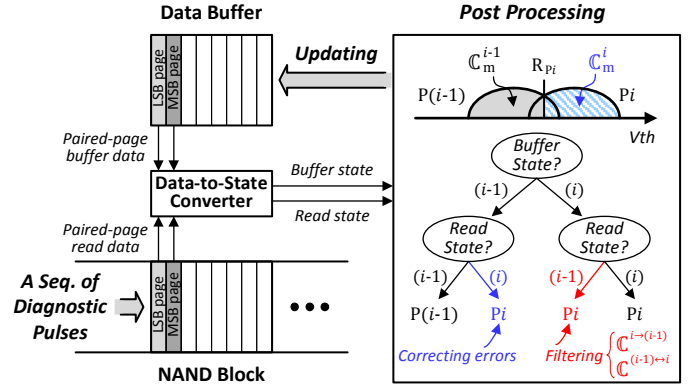


Fig. 6. An overview of the current **FD** implementation with a selective error-correction procedure.

B. A Selective Error-Correction Procedure

By progressively taking CM elements into account of a data recovery process, the proposed **FD** can more efficiently recover retention failures over **DRRP**. Since non-zero $|\mathbb{C}^{i \rightarrow (i-1)}|$ indicates the occurrence of additional charge loss during the recovery process, if those elements can be identified from the read data, the data recovery capability can be enhanced. Moreover, since random charge fluctuation is more active in highly-damaged cells [9] (which probably contributed to retention errors [2]), taking $\mathbb{C}^{(i-1) \leftrightarrow i}$ elements as retention-failed cells can be an effective way of correcting retention errors. Another important advantage of considering $\mathbb{C}^{(i-1) \leftrightarrow i}$ elements is that the data recovery speed can be accelerated. Since $\mathbb{C}^{(i-1) \leftrightarrow i}$ elements are frequently observed even in the early stage of the recovery process as shown in Fig. 5(b), if the error-correction process can consider these elements, the error-correction capability nearly doubles in the early stage. Since each CM element can be separately extracted from the read data as shown in Fig. 5(b), conceptually, the total number EC_m^{FD} of bit errors under **FD** after the m^{th} read pulse can be expressed as follows:

$$EC_m^{FD} = EC_m^{DRRP} - |\mathbb{C}^{i \rightarrow (i-1)}| - |\mathbb{C}^{(i-1) \leftrightarrow i}| \quad (5)$$

For example, if $|\mathbb{C}^{(i-1) \rightarrow i}|$, $|\mathbb{C}^{i \rightarrow (i-1)}|$ and $|\mathbb{C}^{(i-1) \leftrightarrow i}|$ are 21, 3 and 6, respectively, after 1000 read pulses as shown in Fig. 5(b), EC_{1000}^{DRRP} is 48 ($= 66 - 21 + 3$) while EC_{1000}^{FD} is only 39 ($= 66 - 21 - 6$). In this example, **DRRP** reduces retention errors by 27% while **FD** reduces retention errors by 41%.

C. FD Implementation

Based on the charge movement model, we have implemented **FD** with the selective error-correction procedure. Fig. 6 shows an overview of the current **FD** implementation which consists of two main steps, a diagnostic step and a post-processing step.

In the diagnostic step, a sequence of diagnostic pulses is applied to retention-failed cells. The main role of the diagnostic step is two-fold. First, it recharges retention-loss cells (same as **DRRP** [8]). Second, it senses the V_{th} changes in response to diagnostic pulses for the following post-processing step. In order to achieve these two functions at the same time, we use a read operation as a diagnostic pulse. Since a read operation senses the data of a selected page while it applies the read voltage (e.g., ~ 6 V) to unselected pages in a NAND block, when read operations are sequentially executed to all of pages in a block, recharging the unselected pages and sensing

the selected page can be executed in a pipelined fashion. Since the effect of just one read pulse on recharging may not be noticeable for causing V_{th} changes, it is more efficient to use a sequence of (consecutive) read pulses as a unit operation of the diagnostic step. For example, ten consecutive read pulses are required to cause V_{th} changes in our measurements. On the other hand, in order to detect randomly-fluctuated cells (i.e., cells in $\mathbb{C}^{(i-1)\leftrightarrow i}$) as early as possible, the post-processing step is invoked after every read operations in the early stage (e.g., less than one hundred read pulses) of **FD**. If the number of consecutive read pulses is conditionally changed (we call this policy the *variable-length sequence policy*), although the data recovery capability may not be improved, the data recovery speed may be substantially enhanced.

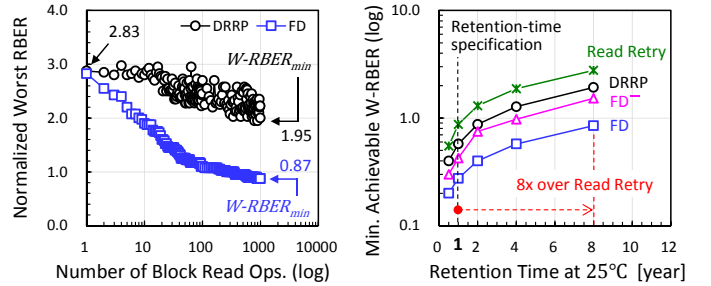
In the following post-processing step, **FD** identifies retention-loss cells by a selective error-correction procedure so that retention errors can be progressively corrected. Since *CM* is based on V_{th} states as presented in Section IV-A, it is necessary to convert raw data to V_{th} states before the post-processing step as shown in Fig. 6. The selective error-correction procedure is based on simple, but effective heuristics: (1) *When a buffer state is $P(i)$, if the corresponding read state is $P(i-1)$ or P_i , then the buffer state is not changed.* On the other hand, (2) *when a buffer state is $P(i-1)$, if the corresponding read state is $P(i)$, then the buffer state is changed to $P(i)$.* The first heuristic can avoid the negative impacts of V_{th} -decreased cells (i.e., cells in $\mathbb{C}^{i\rightarrow(i-1)}$ or $\mathbb{C}^{(i-1)\leftrightarrow i}$) on correcting retention errors. On the other hand, the second heuristic takes V_{th} -increased cells (i.e., cells in $\mathbb{C}^{(i-1)\rightarrow i}$ or $\mathbb{C}^{(i-1)\leftrightarrow i}$) as retention-failed cells so that retention errors can be selectively corrected. In **FD**, once a retention-failed cell is corrected by the second heuristic, then the corrected cell is no longer considered in the remaining post-processing steps by the first heuristic. However, since **DRRP** takes only a cell belongs to a set $\{\mathbb{C}_0^{i-1} \cap \mathbb{C}_m^i\}$ (after the m^{th} read pulse) as a retention-failed cell regardless of the error-correction history, **DRRP** cannot properly handle the negative impacts of V_{th} -decreased cells on its data recovery capability.

The result of the post-processing step is updated to the data buffer as shown in Fig. 6 so that retention errors in the buffer can be progressively corrected. If the buffered data is correctable by ECC, **FD** completes its recovery procedure and rewrites the recovered data to a free page. Otherwise, two **FD** steps are repeated until a pre-set maximum iteration count (e.g., one thousand) is reached.

Our proposed **FD** implementation as shown in Fig. 6 requires a data buffer with a single block size (e.g., 1 MB in an MLC device) and several state registers with a single page size (e.g., 8 KB). Moreover, since the post-processing step and the diagnostic step can be performed independently for each other, **FD** can exploit a pipelined execution between the diagnostic step and the post-processing step so that the total **FD** execution time can be partially reduced.

V. EXPERIMENTAL RESULTS

We evaluated the effectiveness of **FD** for recovering retention failures with ten blocks (pre-cycled for 3K P/E cycles) out of five 20-nm node NAND chips. As a main evaluation metric, we measured RBERS of about 10,000 sectors and computed W-RBER (i.e., the normalized worst-case RBER as defined in Section III) among measured sectors. In order to emulate a



(a) W-RBER variations over varying the numbers of read operations in the 8-year retention time condition. (b) $W-RBER_{min}$ variations over varying retention time condition under different techniques.

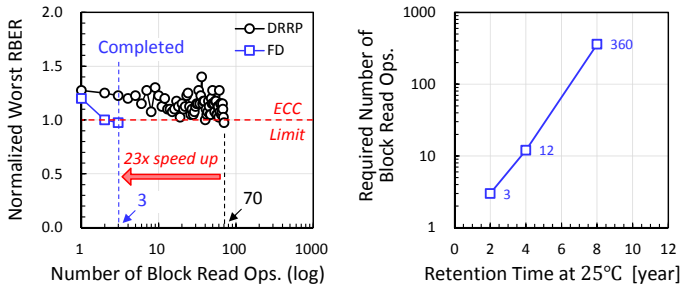
Fig. 7. Comparisons of the data recovery capability under different data recovery techniques.

long retention state such as a 2-year retention time condition, we baked selected chips at 100 °C for a equivalent retention time (e.g., 4 hours) estimated by the Arrhenius equation [4].

In order to compare the data recovery capability of **DRRP** and **FD**, we measured W-RBER while varying the number of read pulses in a very long range (without applying the stopping condition of the error-correction procedures). Fig. 7(a) shows the data recovery capability of both techniques in the 8-year retention case. Since **DRRP** cannot lower its W-RBER even with 1,000 read pulses, it cannot recover retention-failed data under the 8-year retention condition. On the other hand, **FD** can recover retention-failed data under the same retention condition after about 360 read pulses. In order to compare the data recovery capability of various techniques under varying retention time conditions, we computed the minimum achievable W-RBER, denoted as $W-RBER_{min}$, of each technique for a given retention condition. For example, in Fig. 7(a), $W-RBER_{min}$ of **FD** is 0.87 while $W-RBER_{min}$ of **DRRP** is 1.95. Intuitively, $W-RBER_{min}$ indicates the maximum data recovery power of a given technique. Fig. 7(b) shows $W-RBER_{min}$ variations under different retention time conditions for several different techniques. As shown in Fig. 7(b), **FD** can effectively extend the NAND retention capability by up to 8 years (which is eight times longer than the retention-time specification) while **DRRP** can guarantee only 2-year retention times.

The enhanced error-correction capability of **FD** over **DRRP** mainly comes from the selective error-correction procedure which can efficiently identify retention-loss cells by finely distinguishing $\mathbb{C}^{i\rightarrow(i-1)}$ and $\mathbb{C}^{(i-1)\leftrightarrow i}$ elements as explained in Section IV-B. In order to understand the impact of the fine-grained cell classification on the data recovery capability, we disabled the $\mathbb{C}^{(i-1)\leftrightarrow i}$ identification step from **FD**. We denote this modified **FD** technique by **FD**⁻. The only difference between **DRRP** and **FD**⁻ is for **FD**⁻ to filter cells in $\mathbb{C}^{i\rightarrow(i-1)}$. As shown in Fig. 7(b), **DRRP** can extend the NAND retention capability by up to 2 years. On the other hand, **FD**⁻ can extend the NAND retention capability by up to 4 years while **FD** can extend it by up to 8 years. This result indicates that identifying cells in $\mathbb{C}^{(i-1)\leftrightarrow i}$ in the data recovery procedure significantly strengthens the data recovery capability of **FD** over **FD**⁻.

In order to compare the data recovery speed of **DRRP** and **FD**, we tested both techniques under three different retention conditions. Fig. 8(a) shows the data recovery speed of **DRRP** and **FD** in the 2-year retention condition. In **DRRP**, W-RBER slowly decreases as read pulses are repeated, and all the retention errors are corrected (i.e., $W-RBER \leq 1.0$) after



(a) W-RBER variations over varying the number of read operations in the 2-year retention time condition.

(b) Required numbers of read operations to complete **FD** over different retention-time conditions.

Fig. 8. Comparisons of the data recovery speed between **DRRP** and **FD**.

TABLE I. REQUIRED NUMBERS OF READ PULSES TO COMPLETE **FD**.

Retention time	Variable-length sequence policy	Fixed-length sequence policy
2 years	3	10
4 years	12	30
8 years	360	370

applying 70 read pulses. On the other hand, in **FD**, retention errors can be fully corrected only after 3 read pulses. Once all the data are correctable, **FD** is completed. As a result, **FD** can recover retention failures up to 23x faster over **DRRP** for the 2-year retention case. When the average page read time is, for example, 100 μs , it takes about 7 ms for **DRRP** to recover retention failures while only 300 μs is required for **FD**. In order to further compare the data recovery speed in longer retention cases, we performed additional experiments for 4-year and 8-year retention conditions. As shown in Fig. 8(b), in the 4-year and 8-year retention cases, **FD** can successfully recover retention failures after applying 12 and 360 read pulses, respectively. On the other hand, in both cases, **DRRP** could not recover retention failures until 1,000 read pulses. (In our evaluations, the maximum number of read pulses was set to 1,000.)

We also evaluate if the *variable-length sequence policy* (described in Section IV-C) is effective in speeding up the overall data recovery procedure. Under the variable-length sequence policy, until the total number of read pulses reaches 100, a single diagnostic pulse is applied to NAND cells between consecutive post-processing step. Once the total number of read pulses reaches 100, ten consecutive read pulses are applied in a row between consecutive post-processing step. In order to evaluate the effectiveness of the variable-length sequence policy, we compared it with the *fixed-length sequence policy* (which always applies ten consecutive read pulses at a time). As summarized in Table I, in the 2-year and 4-year retention cases, the variable-length sequence policy can reduce the total data recovery time by 70% and 60%, respectively, over the fixed-length sequence policy. This is because, in an early stage of **FD**, frequently reading retention-failed cells can increase the probability of detecting cells in $\mathbb{C}^{(i-1) \leftrightarrow i}$ so that they can be excluded quickly from the remaining data recovery procedure. However, in the 8-year retention case, the variable-length sequence policy has a little benefit over the fixed-length sequence policy. This is because the main advantage of the variable-length sequence policy is to detect cells in $\mathbb{C}^{(i-1) \leftrightarrow i}$ early. For severely retention-failed data such as the 8-year retention case, after most of cells in $\mathbb{C}^{(i-1) \leftrightarrow i}$ are detected early, other components such as

$\mathbb{C}^{(i-1) \rightarrow i}$ (not yet classified) are the dominant source of the retention errors. As a result, the overall recovery time of **FD** is decided by how long it takes to find cells in $\mathbb{C}^{(i-1) \rightarrow i}$ (which is similar under two polices).

VI. CONCLUSIONS AND FUTURE WORK

We have presented a new data recovery technique, called FlashDefibrillator (**FD**), for recent 20-nm (or below) NAND flash memory. Based on the unique characteristics of charge-transient behavior in recent NAND flash memory, **FD** can identify retention-failed cells quickly and accurately so that it can be used as an on-line solution for handling retention failures in NAND flash memory. Our experimental results with 20-nm node NAND chips show that **FD** can recover retention failures up to 23x faster over the existing **DRRP** technique. Furthermore, since **FD** can recover severely retention-failed data, it effectively extends the NAND retention time. Our result indicates that the NAND retention time can be effectively extended by up to 8x over the specified retention time.

The current version of **FD** can be further improved in several ways. For example, most of computations required by **FD** can be more efficiently implemented using simple NAND flash-internal arithmetic and logical units. Since most **FD** computations can be carried out inside NAND flash chips, it is expected that the current **FD** execution time can be significantly reduced.

ACKNOWLEDGMENT

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Ministry of Science, ICT and Future Planning (MSIP) (NRF-2013R1A2A2A01068260). The ICT at Seoul National University and IDEC provided research facilities for this study.

REFERENCES

- [1] J. E. Brewer *et al.*, "Nonvolatile Memory Technologies with Emphasis on Flash," Wiley, pp. 12–13, 2008.
- [2] N. Mielke *et al.*, "Bit Error Rate in NAND Flash Memories," in *Proc. IEEE IRPS*, 2008.
- [3] A. Cox, "JEDEC SSD Specifications Explained," Available: <http://www.jedec.org>.
- [4] JEDEC standard, "Method for Developing Acceleration Models for Electronic Component Failure Mechanisms," *JESD91A*, Aug. 2003.
- [5] R.-S. Liu *et al.*, "Optimizing NAND Flash-Based SSDs via Retention Relaxation," in *Proc. USENIX FAST*, 2012.
- [6] L. Shi *et al.*, "Retention Trimming for Wear Reduction of Flash Memory Storage Systems," in *Proc. DAC*, 2014.
- [7] Y. Cai *et al.*, "Data Retention in MLC NAND Flash Memory: Characterization, Optimization, and Recovery," in *Proc. IEEE Symp. High-Performance Comput. Architecture (HPCA)*, 2015.
- [8] S. Tanakamaru *et al.*, "Error-Prediction LDPC and Error-Recovery Schemes for Highly Reliable Solid-State Drives (SSDs)," *IEEE J. Solid-State Circuits*, vol. 48, no. 11, pp. 2920–2933, Nov. 2013.
- [9] K. Fukuda *et al.*, "Random Telegraph Noise in Flash Memories - Model and Technology Scaling," in *Proc. IEEE IEDM*, 2007.
- [10] B. Tang *et al.*, "Read and Pass Disturbance in the Programmed States of Floating Gate Flash Memory Cells With High- k Interpoly Gate Dielectric Stacks," *IEEE Trans. Electron Devices*, vol. 60, no. 7, pp. 2261–2267, July 2013.
- [11] A. A. Chien *et al.*, "Moore's Law: The First Ending and a New Beginning," *IEEE Computer*, vol. 46, no. 12, pp. 48–53, Dec. 2013.
- [12] J. Yang, "High-Efficiency SSD for Reliable Data Storage Systems," in *Proc. Flash Memory Summit*, 2011.