FlexLevel: a Novel NAND Flash Storage System Design for LDPC Latency Reduction*

Jie Guo[§], Wujie Wen[§], Jingtong Hu[†], Danghui Wang[‡], Hai Li[§], Yiran Chen[§]

[§] Department of Electrical and Computer Engineering, University of Pittsburgh
[†] School of Electrical and Computer Engineering, Oklahoma State University
[‡] School of Computer Science, Northwestern Polytechnical University

[§]{jig26, wuw2, hal66, yic52}@pitt.edu, [†]jthu@okstate.edu, [‡]wangdh@nwpu.edu.cn

ABSTRACT

LDPC code is introduced in NAND flash memory to handle high BER (bit error rate) incurred by technology scaling. Despite strong error correction capability, LDPC decoding induces long NAND flash read latency. In this work, we propose FlexLevel – a robust NAND flash storage system design to improve data reliability and read efficiency affected by the LDPC operations. FlexLevel first reduces BER by enlarging noise margins via V_{th} (threshold voltage) level reduction. It reduces the sensing levels of LDPC but also causes loss of storage capacity. To compensate this capacity loss with minimum impact on read performance, FlexLevel identifies the data with high LDPC overhead and only applies the V_{th} level reduction technique to those data. Experimental results show that compared with state-of-the-art, FlexLevel can achieve up to 33% read speedup with very moderate capacity loss.

Categories and Subject Descriptors

B.8.1 [**Performance and Reliability**]: Reliability, Testing and Fault-Tolerance; D.4.2 [**Operating Systems**]: Storage Management—*Storage hierarchies*

Keywords

NAND flash memories, LDPC, bit error rate

1. INTRODUCTION

NAND flash memory is a prevalently used storage devices, the applications of which range from small mobile devices to high-end servers. In a NAND flash cell, logic bits are represented by different levels of V_{th} (threshold voltage) on a floating gate transistor. The increase in the number of bits stored in a MLC (multi-level cell) is realized by introducing more V_{th} levels. Compared to a SLC (single-level cell) which has only two V_{th} levels, the four-bit MLC is more susceptible to device noise and has higher BER (bit error rate) due to distinction between two neighboring V_{th} levels.

To protect data integrity, ECC (error correction code) is usually deployed in NAND flash storage systems. The selection of ECC is based on the required error correction capability. The qualified ECC should guarantee that UBER (uncorrectable bit error rate) of the storage system is under an acceptable level, e.g., 10^{-14}. For the storage systems of 3Xnm NAND flash memory, hard-decision ECC such as BCH (Bose-Chaudhuri-Hocquenghem) code is usually utilized to reduce UBER.

As technology node scales down to 2Xnm, the BER in NAND flash memory significantly increases and conventional hard-decision ECC is no longer sufficient to meet the system reliability requirements. To enhance error correction capability, LDPC (Low Density Parity Check) code is introduced to NAND flash storage system design. However, LDPC also significantly prolongs read latency: when BER reaches 10^{-2}, LDPC code can induce $7\times$ higher read latency than hard-decision ECC [1, 2].

To address the performance issue of LDPC code, several research works were performed. From the perspective of signal processing, G. Dong et al. proposed entropy coding to reduce data transfer time [3]. However, entropy coding incurs high hardware cost. [1, 4, 2] explored the optimization space to reduce LDPC overhead at system level, but the efficiency of their schemes degrades when BER increases. In this work, we propose FlexLevel – a technique that can effectively reduce LDPC code overhead with a low hardware cost even when BER is high, e.g., 10^{-2}.

FlexLevel reduces BER by enlarging noise margins via V_{th} level reduction. With the reduced BER, no extra sensing level is needed by LDPC decoding, leading to read performance improvement. However, V_{th} level reduction may cause storage capacity loss. In order to overcome this drawback, FlexLevel identifies the data with high LDPC overhead and applies V_{th} level reduction only to these data. As a result, capacity loss can be effectively minimized. The contributions of this work are summarized as follows:

- LevelAdjust technique is proposed at device level to reduce BER. In LevelAdjust, the number of V_{th} levels is reduced to extend noise margins. In order to accommodate the reduced V_{th} levels, novel bit mapping technique, bitline structure, and program method are proposed to achieve substantial BER reduction.
- To further reduce BER under LevelAdjust, NUNMA (non-uniform noise margin adjustment) technique is proposed by allocating larger noise margin to susceptible V_{th} levels.
- AccessEval technique is proposed to accurately identify data with high LDPC overhead. By only applying LevelAdjust and NUNMA to the identified data, we can balance the performance improvement and capacity loss based on application needs.

According to our experimental results, FlexLevel can achieve up to 33% read speedup with only 6% capacity loss.

2. BACKGROUND

*This work was supported in part by NSF CCF-1217947, NSF CNS-1116171, NSF CNS-1342566, NSFC 61472322, NSFC 61272122, and Fundamental Research Funds for Central Universities of China 3102014JSJ0001.

DAC '15, June 07 - 11, 2015, San Francisco, CA, USA

http://dx.doi.org/10.1145/2744769.2744843

In this section, we first present the basics of MLC NAND flash, including its working mechanism and noise sources. We then discuss the issues of LDPC application in NAND flash storage systems.

2.1 MLC NAND Flash Basics

A NAND flash cell is a floating gate transistor whose V_{th} can be adjusted by changing the number of electrons on the floating gate. The stored values are denoted by different V_{th} levels. In a MLC NAND flash cell, four V_{th} levels are used to store two bits. Gray code is normally utilized to map these bit values, such as mapping 11, 10, 00, and 01 to V_{th} levels 0, 1, 2, and 3, respectively.

A MLC NAND flash chip normally consists of many blocks, each of which contains a number of pages. Figure 1(a) illustrates the schematic of one block where each wordline stores two page groups: even vs. odd. An even/odd bitline structure is adopted to select different page groups by the corresponding bitlines [5]. Each page group contains two pages: a lower page and an upper page. The LSB (least significant bit) of the two bits in the MLC NAND flash cell belongs to the lower page while the MSB (most significant bit) belongs to the upper page.

MLC NAND flash supports three operations: program (write), read and erase. Program operation realizes bit storage by iteratively applying program pulses to a cell until the programmed V_{th} reaches a verify voltage. To program the MLC, a two-step program method is usually performed [6]. The 1st and the 2nd program operations store data to LSB and MSB, respectively. The logic bits stored on the floating gate are read out by comparing the cell's V_{th} with three read reference voltages [5]. Erase operation removes electrons from the floating gate to restore the NAND flash cell to a ready state (V_{th} level 0) for upcoming programming operations.

2.2 NAND Flash Noise and ECCs

As aforementioned, noise in NAND flash memory causes bit errors. Cell-to-cell interference and retention time limit are two major contributors to bit errors in NAND flash memory [7, 8].

Cell-to-cell interference results from a parasitic capacitance coupling effect: programming one floating gate transistor can raise the V_{th} of neighboring cells. The impact of cell-to-cell interference on V_{th} is shown in the top chart of Figure 1(b). From the figure, we can see that V_{th} shifts to the right because of the interference. Each V_{th} level region is confined by lower and upper read reference voltages V_{rd-ref}. If the V_{th} exceeds its upper read reference voltage, an error will occur. Accordingly, the cell-to-cell interference noise margin is defined as the voltage difference between the initial V_{th} before the interference occurs and the upper read reference voltage.

In contrast to cell-to-cell interference noise, retention time noise causes V_{th} to decrease due to electron detrapping and stress induced leakage current [7]. Retention time noise affects the stored data after it is programmed. The impact of retention time noise on V_{th} is shown in the bottom chart of Figure 1(b). From the figure, we can see that V_{th} gradually decreases after being stored. If V_{th} drops below its lower read reference voltage, an error will also occur. Therefore, the retention time noise margin is defined as the voltage difference between the lower read reference voltage and the V_{th} right after programming.

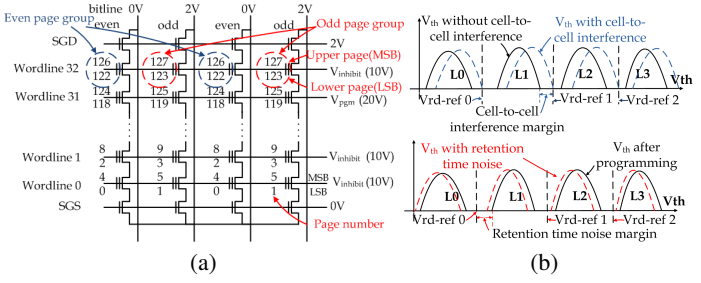


Figure 1: (a) MLC NAND flash even/odd bit-line structure. (b) Illustration of cell-to-cell interference and retention time noise.

As the technology node scales down to 2Xnm, the noise margin of MLC NAND flash memory is greatly narrowed. As a result, the BER can reach up to 10^{-2} [1]. To provide more powerful error correction capability, LDPC code is introduced in the system design. LDPC can work under either a hard-decision or a soft-decision fashion. When the BER is low, binary error correction information is sufficient to decode data correctly. LDPC is regarded as hard-decision. However, when the BER is high, LLR (log likelihood ratio) information is required for decoding. LDPC is regarded as soft-decision. The performance of soft-decision LDPC heavily depends on the accuracy of LLR information. In NAND flash memory, LLR information can be collected by sensing V_{th} with extra reference voltages or soft sensing levels. Extra memory sensing overhead together with extra data transfer time causes a longer read latency of NAND flash memory. When the BER reaches 10^{-2}, LDPC code will significantly prolong the read latency [1].

3. FLEXLEVEL TECHNIQUE OVERVIEW

In this work, we propose FlexLevel technique to reduce the LDPC code latency. Based on the observation that LDPC overhead increases with BER [2], FlexLevel speeds up read by minimizing the BER of NAND flash memory via V_{th} level reduction. Seong et al. [9] proposed to reduce V_{th} levels to minimize the BER in PCM (phase change memory), which shares a goal similar to that of our work. However, PCM does not adopt even/odd bit-line structure and the error mechanism of PCM is very different from NAND flash. Therefore, Seong's techniques cannot be directly applied to NAND flash memory. To facilitate V_{th} reduction in MLC NAND flash cells, we need to propose new encoding and noise margin adjustment techniques.

Figure 2 shows the overview of FlexLevel design, including two major components: *LevelAdjust* and *AccessEval*. The LevelAdjust technique is proposed to reduce the BER at device level, i.e., adjusting cell noise margins by changing the number of V_{th} levels of floating gate transistors. This technique allows one MLC NAND flash cell to have two states: normal state and reduced state. In normal state, the cell has four V_{th} levels, working as a regular MLC NAND flash cell. In reduced state, the cell has only three V_{th} levels. The BER of the reduced state cell is reduced by allocating a larger noise margin to each V_{th} level. At the early P/E cycling

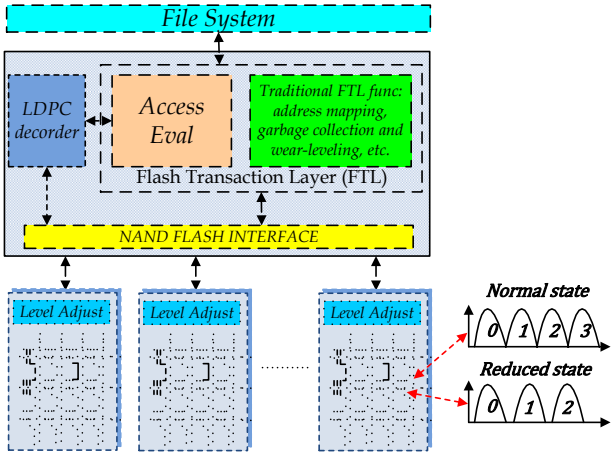


Figure 2: FlexLevel NAND flash storage system overview.

stage, the BER is low and all NAND flash cells are in normal state. Following the increase of P/E cycles and storage time, the cells may switch to reduced state to control the BER below a threshold. Based on the observation that different V_{th} levels associate with different BER, we proposed the NUNMA (non-uniform noise margin adjustment) technique to further reduce BER . The reduction of the BER results in fewer soft sensing levels needed by LDPC code, consequently improving the system read performance.

We note that the V_{th} reduction in the LevelAdjust technique introduces storage capacity loss. To maximize performance improvement and minimize storage capacity loss, we propose AccessEval technique to selectively apply LevelAdjust to NAND flash cells based on demand. AccessEval evaluates the LDPC overhead of the stored data by checking their access patterns. For data with the access patterns associated with high LDPC overhead, AccessEval stores them in the reduced state cells. On the contrary, data with the access patterns that lead to low LDPC overhead will be stored in the normal state cells. Thereby, soft-decision LDPC induced read latency can be effectively reduced with minimum storage loss. The AccessEval module is implemented in FTL (flash translation layer), which is a software layer emulating NAND flash memory as a block device [10].

4. LEVELADJUST: V_{TH} LEVEL ADJUSTMENT

This section introduces the details of the LevelAdjust technique: the basic LevelAdjust is presented in Section 4.1 first, followed by the introduction of NUNMA technique in Section 4.2. At last, the overhead of LevelAdjust is discussed in Section 4.3.

4.1 Basic LevelAdjust Technique

LevelAdjust minimizes the BER of NAND flash cells through V_{th} level reduction and introduces two states to the NAND flash cells. In normal state, a cell has four V_{th} levels. It adopts the same even/odd bitline structure and program/erase operation in the regular MLC NAND flash devices. Standard gray code is still deployed to map two bits to the four V_{th} levels. In reduced state, a cell has only three V_{th} levels: V_{th} level 0, 1 and 2. Compared with the normal state cell, the reduced state cell has enlarged noise margins at each V_{th} level and hence can bear larger noise magnitude.

In LevelAdjust technique, if gray code is still used to map the bits in reduced state, each cell can only store one bit, leading to loss of a half storage capacity. Hence, ReduceCode technique is proposed to maximize the storage capacity of the reduced state cell. We observed that each reduced state cell has three V_{th} levels and two cells indeed can represent nine V_{th} combinations. Therefore, ReduceCode uses eight out of the nine V_{th} combinations to represent 3 bits. In this way, two cells can represent 3 bits instead of just 2 bits with gray code.

A mapping scheme between a 3-bit value and V_{th} level combinations in the reduced state cell is shown in Table 1. V_{th} I and V_{th} II represent the V_{th} levels of the 1st cell and 2nd cell, respectively. Similar to gray code, ReduceCode can minimize the BER when V_{th} distortion occurs. Take a 3-bit value 101 as an example, it is mapped to V_{th} level 0 in the 1st cell and V_{th} level 2 in the 2nd cell. In case of V_{th} distortion, e.g., the V_{th} level of the 2nd cell changes from level 2 to level 1, the 3-bit value 101 will change to 001, causing only one-bit error. In summary, one level distortion in any of the two cells will cause only one bit error. Thus, bit errors are effectively minimized.

Table 1: Bit value mapping under ReduceCode

3-bit value	V_{th} I	V_{th} II	3-bit value	V_{th} I	V_{th} II
000	0	0	100	2	2
001	0	1	101	0	2
010	1	0	110	2	0
011	1	1	111	2	1

Table 2: V_{th} transition under 2-step programming operation

MSB	two LSBs	targeted V_{th} I	targeted V_{th} II	ΔV_{th} I	ΔV_{th} II	program seq.
-	00	0	0	–	–	1st program
-	01	0	1	–	0→1	1st program
-	10	1	0	0→1	–	1st program
-	11	1	1	0→1	0→1	1st program
1	00	2	2	0→2	0→2	2nd program
1	01	0	2	–	1→2	2nd program
1	10	2	0	1→2	–	2nd program
1	11	2	1	1→2	–	2nd program

To implement the proposed mapping scheme, a dedicated ReduceCode bitline structure is designed, as shown in Figure 3(a). Two neighboring even or odd cells are combined to represent 3 bits and a pair of even cells or odd cells have one MSB and two LSBs totally. Two LSBs from all even cells on one wordline form a "lower page" while two LSBs from all odd cells on the same wordline form a "middle page". Also, the MSBs from all the cells on the same wordline form a "upper page".

Under the dedicated ReduceCode bitline structure, the original 2-step program operation of MLC NAND flash no longer works. Thus, in ReduceCode bitline structure, we propose a two-step program algorithm to program each page: in the 1st step, two LSBs, i.e., the lower or middle page is programmed; in the 2nd step, the MSBs, i.e., the upper page is programmed.

The V_{th} transitions under two program steps are summarized in Table 2. Here, ΔV_{th} I and ΔV_{th} II denote the V_{th} level transition of the 1st and 2nd cells, respectively. Targeted V_{th} I and targeted V_{th} II denote the V_{th} levels that a cell is programmed to. Before programming, erase operation resets the reduced state cell to V_{th} level 0. During the 1st program step, depending on whether the lower or the middle page needs to be programmed, the even or the odd bitlines will be selected accordingly. The V_{th} level either increases to V_{th} level 1 or remains at V_{th} level 0 based on the stored bit value. During the 2nd program step, the upper page is programmed. Since the MSBs of all cells on the same wordline form the upper page, all bitlines will be selected.

We note that the V_{th} level transition during the 2nd program step depends on the two LSB values mapped in the 1st program step and the MSB value. If MSB is 0, V_{th} level transition stops and V_{th} levels remain the same as that after the 1st program step. If MSB is 1, V_{th} level transition follows Table 2. During read operations of the ReduceCode bitline structure, V_{th} will be compared with the new read reference voltages.

4.2 NUNMA Technique

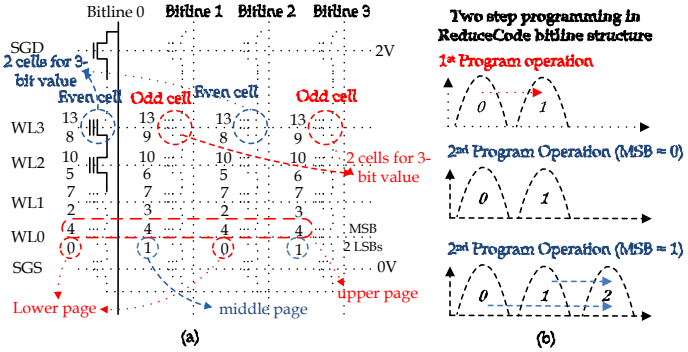


Figure 3: ReduceCode bitline structure.

To further reduce BER, we propose NUNMA (non-uniform noise margin adjustment) technique. Since retention time error dominates the overall BER when P/E cycle is high [6], the NUNMA technique focuses on reducing the retention time error. Retention time BER is V_{th} level dependent. Simulation results show that 78% and 15% bit errors occur at V_{th} level 2 and 1 on average under the basic LevelAdjust technique. It means that the V_{th} decrease speed at the higher level is faster than that at the lower level. Therefore, if we allocate larger noise margins to the higher levels, the retention time errors can be greatly reduced.

A V_{th} level region is confined by its lower and upper read reference voltages. Originally, the program verify voltage is set close to the lower read reference voltage and the V_{th} distribution is placed in the center of its level region, as shown in Figure 4(a). The decrease in V_{th} with an increase of storage time results in retention time errors. In order to improve the retention time noise margins, the programmed V_{th} distribution should be shifted to right by increasing the verify voltage while the read reference voltages remain unchanged. As a result, the programmed V_{th} will be much higher than the reference reference voltage, enabling enhanced noise margin and better tolerance to charge loss, as shown in Figure 4(b). However, increasing the verify voltages may cause V_{th} level 1 to exceed its upper read reference voltage, introducing more cell-to-cell interference errors. Since retention time BER at the low V_{th} levels is lower than that at the high V_{th} levels, it is safe to allocate relatively small retention time noise margins to the low level V_{th}'s and large retention time noise margin to the high level V_{th}'s. A low verify voltage in V_{th} level 1 together with a high verify voltage in V_{th} level 2 can be employed, as shown in Figure 4(c). Both cell-to-cell interference and retention time BER are reduced. NUNMA technique can be easily implemented into NAND flash as the programming verify and read reference voltages are all adjustable [11].

4.3 LevelAdjust Overhead

The application of LevelAdjust is associated with overheads. One overhead is the logic gates that are needed to implement ReduceCode circuit. However, the circuit can be implemented with less than 20 gates, causing negligible hardware and energy costs compared with the entire control logic of a NAND flash chip. Besides the hardware overhead, ReduceCode introduces one clock-cycle encoding and decoding overhead, e.g. 5ns for a 200MHz clock frequency. The induced overhead on data transfer and sensing latency (normally tens of microseconds) is also negligible. The major overhead of LevelAdjust is the capacity loss incurred by V_{th} level reduction. In reduced state, two MLC NAND flash cells are combined to represent 3 bits, leading to 25% storage density reduction compared to two normal state cells. This capacity loss has to be compensated since the storage system capacity must be consistent to the file system. Although the over-provision space [12] may be used to compensate the capacity loss, it may cause performance degradation. In order to minimize such capacity loss, AccessEval technique is introduced in Section 5.

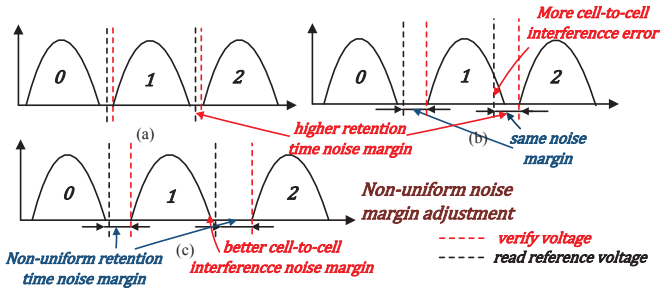


Figure 4: NUNMA technique.

5. ACCESSEVAL: ACCESS PATTERN EVALUATION

In this section, we first present the details of the AccessEval technique. Then the overhead of AccessEval will be discussed.

Because LevelAdjust introduces inevitable storage capacity loss, it will be beneficial to restrict the application of LevelAdjust to a minimum number of NAND flash cells that really need. In reality, not every data contributes equally to the overall LDPC overhead. Therefore, if we can identify the data which contribute to the majority of the LDPC overhead and only apply LevelAdjust to these data, i.e., storing them in the reduced state pages, the impacts of LevelAdjust will be limited to a minimum level. The LDPC overhead can still be reduced while the incurred system storage capacity loss is minimized. Accordingly, we propose AccessEval technique that selectively applies LevelAdjust to the data based on their needs.

The AccessEval module consists of three components: HLO (high-LDPC-overhead) data identifier, ReducedCell pool and AccessEval controller. HLO data identifier determines the proper data, i.e., the HLO data, to be stored in the reduced state pages. ReducedCell pool is a data structure recording the reduced state pages. The size of ReducedCell pool limits the maximum number of the reduced state pages. AccessEval controller manages the data conversion between the reduced state pages and the normal state pages. During read operations, once a data is identified as HLO data, it will be stored in the reduced state pages. If all reduced state pages are used, ReducedCell pool will first convert the least-recently-accessed reduced state pages to normal state pages, and the upcoming HLO data is stored in the newly constructed reduced state pages. In AccessEval, data migration between the reduced state pages and the normal state pages introduces extra program and erase operations. Hence, improving the identification accuracy of HLO data becomes essential to enhance the AccessEval efficiency.

The LDPC overhead contributed by a data depends on the read frequency of this data and the LDPC overhead per read. Intuitively, high read frequency leads to high LDPC overhead. The techniques proposed in [13, 14] can be employed to identify the frequently read data. The LDPC overhead per read is determined by the number of extra sensing levels needed for successful decoding. The higher the sensing level is, the higher LDPC overhead per read is incurred. Based on both read frequency and sensing levels, a LDPC overhead estimation rule can be created as follows: the read frequency of a data is divided into N levels (L_f) while its soft sensing levels are divided into M buckets ($L_{sensing}$). LDPC overhead is measured by $L_f \times L_{sensing}$. If the LDPC overhead of a data exceeds a pre-defined threshold, this data will be stored in the reduced state pages.

According to our experimental results, AccessEval can successfully reduce the capacity loss from 25% down to 6% with only marginal write overheads caused by over-provisioning space reduction. The main overhead of AccessEval is ReducedCell pool which stores the reduced state pages. Assume that each entry in ReducedCell pool is 4 bytes. If 32GB data needs to be stored in reduced state pages and the page size is 16KB, reducedCell pool only occupies 8MB, which is negligible compared to the total memory capacity.

6. EXPERIMENTS

In this section, we first evaluate the effectiveness of LevelAdjust and find out the optimal configuration. Then we show the experimental results of our AccessEval technique.

6.1 LevelAdjust Evaluation

Table 3: Non-uniform LevelAdjust configuration

Scheme	V_{pp}	$V_{verify1}$	$V_{verify2}$	$V_{read-ref1}$	$V_{read-ref2}$
NUNMA 1	0.15	2.71	3.61	2.65	3.55
NUNMA 2	0.15	2.70	3.65	2.65	3.55
NUNMA 3	0.15	2.75	3.70	2.65	3.55

LevelAdjust evaluation is based on the reliability index UBER (uncorrectable bit error rate). Here we assume a rate-n/m ECC is employed. n and m represent information length and total codeword length, respectively. UBER can be estimated by [6, 16]:

$$uber(k) = \frac{1 - \sum_{i=0}^{k} C_m^i p_c^i (1-p_c)^{(m-i)}}{n}. \quad (1)$$

Here, k is the correctable bit number. p_c denotes BER of a single MLC NAND flash cell. The targeted UBER is set to 10^{-15} [10]. A rate-8/9 LDPC code is performed on each 4KB data block.

To accurately obtain p_c, we employ reliability models to mimic the device noise. The V_{th} shift resulting from cell-to-cell interference ΔV_{c2c} can be modeled by [7]

$$\Delta V_{c2c} = \sum_{k=0} \Delta V_p^{(k)} \times \gamma^{(k)}. \quad (2)$$

Here, $\Delta V_p^{(k)}$ denotes the V_{th} shift of the interfering cell after programming and $\gamma^{(k)}$ is the coupling ratio. k represents coupling directions. In even/odd bitline structure, capacitance coupling exists in three directions. Coupling ratios in the three directions can be denoted by γ_x, γ_y and γ_{xy}, which are respectively set to 0.07, 0.09 and 0.005 [17]. The distribution of retention time noise induced V_{th} shift follows $N(\mu_d, \sigma_d^2)$ [7] where μ_d and σ_d^2 can be expressed by:

$$\begin{cases} \mu_d = K_s(x - x_0)K_d N^{0.4} ln(1 + t/t_0), & (3) \\ \sigma_d^2 = K_s(x - x_0)K_m N^{0.5} ln(1 + t/t_0). & (4) \end{cases}$$

Here, K_s, K_d, K_m and t_0 are constants. In our evaluation, according to the data in [18], K_s, K_d and K_m are set to 0.333, 4×10^{-4} and 2×10^{-6}, respectively; t_0 is set to one hour. x_0 is V_{th} level 0 and can be modeled by Gaussian distribution $N(1.1, 0.35)$ [19]. x is the initial V_{th} after programming and t is storage time.

The regular MLC NAND flash cell (i.e., the normal state cell) is used as the baseline in our comparison. Three NUNMA configurations listed in Table 3 are explored to find out the optimal device parameters for BER reduction. Reduction of the BER after cell-to-cell interference (denoted as C2C BER hereafter) is shown in Figure 5. Compared with the baseline, the C2C BER is reduced by up to 6× in NUNMA 1 due to the enhanced noise margin on average. The BER of NUNMA 3 is 50% and 20% higher than NUNMA 1 and 2, respectively. This is because that the verify voltage in NUNMA 3 is higher than that in NUNMA 1 and 2, causing more prominent cell-to-cell interference.

The retention time BERs of reduced state cells are shown in Table 4. On average, the retention time BERs are reduced by 2×, 5× and 9× under the three NUNMA configurations, respectively.

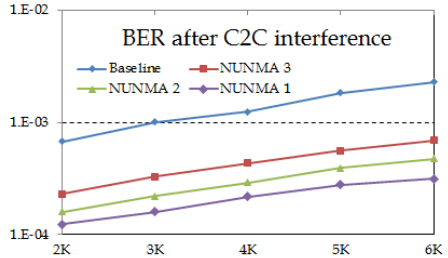


Figure 5: The BER of reduced state cells after C2C interference.

Table 4: BER comparison under three NUNMA configurations

P/E cycles	scheme	1 day	2 days	1 week	1 month
2000	Baseline	0.000638	0.000715	0.00103	0.00184
	NUNMA 1	0.000370	0.000453	0.000827	0.00149
	NUNMA 2	0.000167	0.000173	0.000243	0.000330
	NUNMA 3	0.000120	0.000133	0.000167	0.000181
3000	Baseline	0.00146	0.00169	0.00260	0.00459
	NUNMA 1	0.000677	0.000860	0.00143	0.00249
	NUNMA 2	0.000343	0.000367	0.000570	0.000807
	NUNMA 3	0.000237	0.000257	0.000293	0.000390
4000	Baseline	0.00229	0.00284	0.00456	0.00778
	NUNMA 1	0.00117	0.00149	0.00240	0.00402
	NUNMA 2	0.000443	0.000633	0.000820	0.00150
	NUNMA 3	0.000327	0.000343	0.000457	0.000633
5000	Baseline	0.00359	0.00457	0.00699	0.0120
	NUNMA 1	0.00177	0.00233	0.00349	0.00545
	NUNMA 2	0.000690	0.000853	0.00123	0.00227
	NUNMA 3	0.000460	0.000540	0.000713	0.00109
6000	Baseline	0.00484	0.00613	0.00961	0.0161
	NUNMA 1	0.00218	0.00288	0.00446	0.00672
	NUNMA 2	0.00100	0.00131	0.00192	0.00324
	NUNMA 3	0.000623	0.000627	0.000973	0.00151

NUNMA 3 achieves the lowest retention time BER because high verify voltage provides larger retention time noise margin.

Based on the simulated p_c and the method in [2], we estimate the overhead of LDPC code that provides the desired UBER. The required extra sensing levels of LDPC code of the baseline MLC NAND flash cell are shown in Table 5. From the table, we can see that LDPC code introduces significant sensing overhead. At a P/E cycle of 6000, soft-decision LDPC code requires as much as six extra sensing levels for successful decoding. In comparison, Table 4 shows that NUNMA 3 configuration can keep both the C2C and retention time BERs below the BER limit that triggers extra sensing levels (4×10^{-3}). Hence, no extra sensing levels are incurred during LDPC decoding.

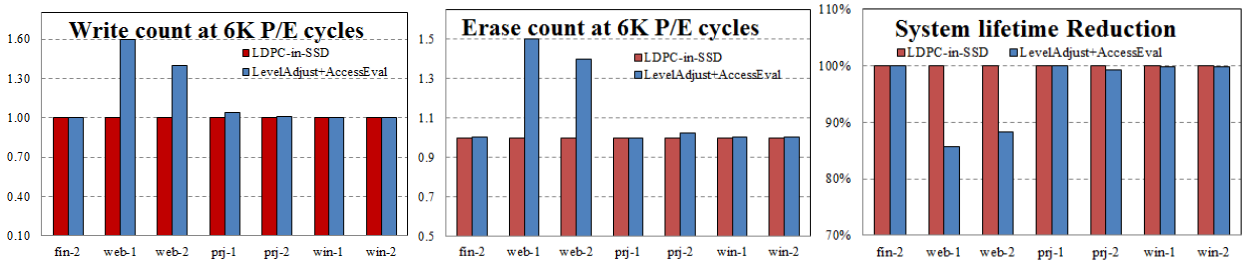
6.2 AccessEval Performance Evaluation

The proposed AccessEval technique is evaluated by Flashsim [20]. We modified the simulator by adding a write-back write buffer and incorporating AccessEval technique. The simulated system has a capacity of 256GB and 27% over-provisioning space. MLC NAND parameters are summarized in Table 6. In reduced state cells, NUNMA 3 configuration is adopted and no extra sensing overhead is introduced. Four storage systems are tested: 1) the system without any scheme (baseline); 2) the one with LDPC-in-SSD [2]; 3) the one only having LevelAdjust design (LevelAdjust-only) and 4) the one incorporating both LevelAdjust and AccessEval (LevelAdjust+AccessEval). In AccessEval, the size of storage space that can be used for LevelAdjust is limited to 64GB. L_f and $L_{sensing}$ are both set to 2. Seven benchmarks from various applications are used in the experiments: fin-2 represents OLTP application; web-1 and web-2 represent search engine on web server; prj-1 and prj-2 are research project workloads; win-1 and win-2 are two PC workloads.

The reduction of the overall response time is summarized in Fig-

Table 5: Required extra LDPC soft sensing levels

P/E cycles	0 day	1 day	2 day	1 week	1 month
3000	0	0	0	0	1
4000	0	0	0	1	4
5000	0	0	1	2	4
6000	0	1	2	4	6

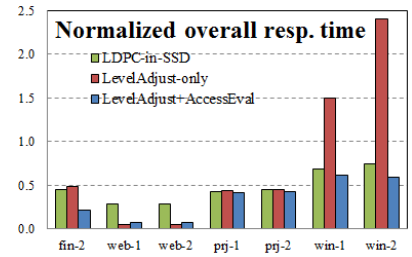


(a) (b) (c)

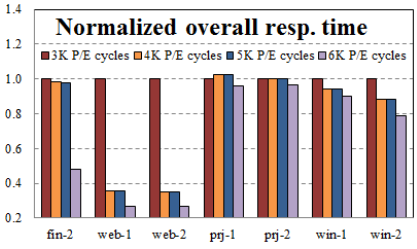
Figure 7: (a) The write count increase under LevelAdjust+AccessEval.(b) The erase count increase under LevelAdjust+AccessEval. (c) The lifetime increase under LevelAdjust+AccessEval. (Note: LDPC-in-SSD has no effects on write count and erase count)

Table 6: Specification of most recent MLC NAND flash memory

Capacity	Block Size	Block Number	Page Size
	1MB	4096	16KB
Timing	Program Latency	Read Latency	Erase Latency
	1000 μs	90μs	3ms



(a)



(b)

Figure 6: (a) Normalized overall average response time under AccessEval. (b) Normalized average response time under different P/E cycle counts compared with LDPC-in-SSD.

ure 6(a). The overall response time of LevelAdjust increases by 27% compared with LDPC-in-SSD due to frequent garbage collection incurred by over-provisioning space loss. Compared with baseline and LDPC-in-SSD, the overall system response time of AccessEval+LevelAdjust reduces by 66% and 33% on average, respectively. The performance gain of AccessEval+LevelAdjust increases with P/E cycle count compared to the LDPC-in-SSD system is shown in Figure 6(b). The average response time reduction achieved by AccessEval+LevelAdjust w.r.t. LDPC-in-SSD system increases from 21% to 33% on average when the P/E cycle increases from 4000 to 6000.

Finally, we evaluate the impact of our techniques on system endurance. Simulation is carried out at a P/E cycle of 6000. Figure 7(a) shows that compared with the LDPC-in-SSD system, the write count of the LevelAdjust+AccessEval system increases by 15% on average. The write count increase comes from the data migration between the normal state cells and the reduced state cells. The maximum relative write increase happens in web-1 and web-2 workloads simply because their original write numbers are low. Figure 7(b) shows the erase count when LevelAdjust+AccessEval is applied. On average, the erase count increases by 13% across all the simulated workloads. Since LevelAdjust+AccessEval only applies when the system BER is high enough to incur extra sensing levels, its impact on the system lifetime is quite moderate: Table 5 shows that LevelAdjust+AccessEval is needed only when the P/E cycle exceeds 4000. Hence, the average lifetime reduction across all the workloads is only 6%, as shown in Figure 7(c).

7. CONCLUSION

In this work, we propose FlexLevel technique to reduce LDPC-induced read latency. The proposed device-level LevelAdjust technique can dynamically reduce BER via V_{th} level reduction. As a result, extra sensing levels can be effectively reduced and read performance is improved. To balance the performance improvement and the storage capacity loss induced by LevelAdjust, we propose the AccessEval technique at system level. Simulation results show that compared with the previous works, our proposed design can achieve up to 33% read speedup with only 6% capacity loss.

8. REFERENCES

[1] S. Tanakamaru *et al.*, "Error-prediction analyses in 1x, 2x and 3xnm nand flash memories for system-level reliability improvement of solid-state drives (ssds)," in *IRPS*, pp. 3B.3.1–3B.3.6, 2013.

[2] K. Zhao *et al.*, "Ldpc-in-ssd: making advanced error correction codes work effectively in solid state drives.," in *FAST*, pp. 243–256, 2013.

[3] G. Dong *et al.*, "Reducing data transfer latency of nand flash memory with soft-decision sensing," in *ICC*, pp. 7024–7028, 2012.

[4] G. Dong *et al.*, "On the use of soft-decision error-correction codes in nand flash memory," *TCAS*, vol. 58, no. 2, pp. 429–439, 2011.

[5] Y. Cai *et al.*, "Error patterns in mlc nand flash memory: measurement, characterization, and analysis," in *DATE*, pp. 521–526, 2012.

[6] N. Mielke *et al.*, "Bit error rate in nand flash memories," in *IRPS*, pp. 9–19, 2008.

[7] Y. Pan *et al.*, "Quasi-nonvolatile ssd: trading flash memory nonvolatility to improve storage system performance for enterprise applications," in *HPCA*, pp. 1–10, 2012.

[8] J. Guo and et al., "Dpa: A data pattern aware error prevention technique for nand flash lifetime extension.," in *ASP-DAC*, pp. 592–597, 2014.

[9] N. H. Seong *et al.*, "Tri-level-cell phase change memory: toward an efficient and reliable memory system," in *ISCA*, pp. 440–451, 2013.

[10] R. Liu *et al.*, "Optimizing nand flash-based ssds via retention relaxation," in *FAST*, 2012.

[11] Y. Cai *et al.*, "Threshold voltage distribution in mlc nand flash memory: characterization, analysis, and modeling," in *DATE*, pp. 1285–1290, 2013.

[12] S. Chao *et al.*, "Scm capacity and nand over-provisioning requirements for scm/nand flash hybrid enterprise ssd," in *IMW*, pp. 64–67, 2013.

[13] D. Park *et al.*, "Hot and cold data identification for flash memory using multiple bloom filters," in *FAST*, 2011.

[14] T. Luo *et al.*, "hstorage-db: heterogeneity-aware data management to exploit the full capability of hybrid storage systems," *VLDB Endowment*, vol. 5, no. 10, pp. 1076–1087, 2012.

[15] J. Guo and et al., "Da-raid-5: a disturb aware data protection technique for nand flash storage systems," in *DATE*, pp. 380–385, 2013.

[16] K. Prall, "Scaling non-volatile memory below 30nm," in *NVSMW*, pp. 5–10, 2007.

[17] H. Sun *et al.*, "Quantifying reliability of solid-state storage from multiple aspects," in *SNAPI*, 2011.

[18] G. Dong *et al.*, "Using data postcompensation and predistortion to tolerate cell-to-cell interference in mlc nand flash memory," *TCAS*, vol. 57, no. 10, pp. 2718–2728, 2010.

[19] "A simulator for various ftl scheme." http://csl.cse.psu.edu/?q=node/322.