

Using Quasi-EZ-NAND Flash Memory to Build Large-Capacity Solid-State Drives in Computing Systems

Yangyang Pan, *Student Member, IEEE*,
Guiqiang Dong, *Student Member, IEEE*,
Ningde Xie, and
Tong Zhang, *Senior Member, IEEE*

Abstract—Future flash-based solid-state drives (SSDs) must employ increasingly powerful error correction code (ECC) and digital signal processing (DSP) techniques to compensate the negative impact of technology scaling on NAND flash memory device reliability. Currently, all the ECC and DSP functions are implemented in a central SSD controller. However, the use of more powerful ECC and DSP makes such design practice subject to significant speed performance degradation and complicated controller implementation. An EZ-NAND (Error Zero NAND) flash memory design strategy is emerging in the industry, which moves all the ECC and DSP functions to each memory chip. Although EZ-NAND flash can simplify controller design and achieve high system speed performance, its high silicon cost may not be affordable for large-capacity SSDs in computing systems. We propose a quasi-EZ-NAND design strategy that hierarchically distributes ECC and DSP functions on both NAND flash memory chips and the central SSD controller. Compared with EZ-NAND design concept, it can maintain almost the same speed performance while reducing silicon cost overhead. Assuming the use of low-density parity-check (LDPC) code and postcompensation DSP technique, trace-based simulations show that SSDs using quasi-EZ-NAND flash can realize almost the same speed as SSDs using EZ-NAND flash, and both can reduce the average SSD response time by over 90 percent compared with conventional design practice. Silicon design at 65 nm node shows that quasi-EZ-NAND can reduce the silicon cost overhead by up to 44 percent compared with EZ-NAND.

Index Terms—Flash memory, solid-state drive (SSD), ECC, LDPC

1 INTRODUCTION

THE steady bit cost reduction over the past decade has enabled NAND flash memory enter increasingly diverse applications, and it is now economically viable to implement large-capacity solid-state drives (SSDs) using NAND flash memory. However, as the semiconductor industry is aggressively pushing the scaling of NAND flash memory technology and the use of multilevel per cell (MLC) storage, NAND flash memory cells are subject to increasingly severe noise and distortion, in particular program/erase (P/E) cycling effects [1] and cell-to-cell interference [2]. Therefore, in order to ensure system data storage integrity and maintain sufficient PE cycling endurance and data retention, increasingly powerful and sophisticated error correction code (ECC) and digital signal processing (DSP) techniques become indispensable in future SSDs [3].

Most SSDs use a dedicated central controller to control all the NAND flash memory chips and handle I/O interface with the host. In conventional design practice, all the ECC and DSP functions are implemented in the SSD controller. Nevertheless, as more powerful ECC and DSP techniques are being used, such a conventional

design practice is subject to a critical issue: Those powerful ECC (e.g., low-density parity-check (LDPC) code [4]) and DSP (e.g., signal postcompensation/predistortion [5] for compensating cell-to-cell interference) may demand fine-grained memory cell sensing (e.g., the threshold voltage of each 2 bit/cell memory cell is quantized into 4 bits during memory sensing). This directly results in much higher flash-to-controller data transfer traffic and hence significantly degrades the SSD speed performance. In addition, as NAND flash memory I/O data transfer rate continues to increase and SSDs employ more NAND flash memory chips on each channel to improve system performance, ECC and DSP modules on the controller must meet very stringent speed requirement, which can make their silicon implementation a challenge.

Driven by the Open NAND Flash Interface (ONFI) working group [6], NAND flash memory manufacturers are currently developing EZ-NAND (Error Zero NAND) flash memory, where all the ECC and DSP functions are embedded in each NAND flash chip through die packaging. Although the EZ-NAND concept was proposed mainly for simplifying controller/host design, the use of EZ-NAND flash memory can meanwhile improve the SSD speed performance by reducing flash-to-controller data transfer traffic, i.e., NAND flash memory chips no longer need to transfer the ECC coding redundancy and any fine-grained memory sensing results to the controller. However, EZ-NAND flash memory chips may be noticeably more expensive than conventional NAND flash memory chips, especially when very sophisticated ECC and DSP functions are being used. Although this may not be a critical issue for systems with one or few NAND flash memory chips (e.g., mobile phones), it may not be affordable for large-capacity SSDs consisting of tens or hundreds of NAND flash memory chips.

In this work, we propose a *quasi-EZ-NAND* flash memory design strategy that, compared with EZ-NAND design concept, can maintain almost the same speed performance while largely reducing the cost overhead. Each quasi-EZ-NAND flash memory chip only incorporates relatively weak and hence less sophisticated ECC and DSP functions that can ensure data storage integrity with a sufficiently high probability, and the central SSD controller contains the full-strength ECC and DSP functions that are executed only when the weak ECC and DSP within quasi-EZ-NAND flash memory chips fail. Such a *hierarchical* ECC and DSP implementation strategy is particularly effective for NAND flash memory, which can be explained as follows: NAND flash memory cell storage reliability gradually degrades with P/E cycling, and the full-strength ECC and DSP are geared to the worst case raw storage reliability as memory P/E cycling reaches the endurance limit. Therefore, the full-strength ECC and DSP are essentially *stronger-than-enough* for most of the time. As a result, the weak ECC and DSP can be more likely sufficient most of the time, especially during the early lifetime of NAND flash memory chips.

To quantitatively evaluate the effectiveness of this proposed simple design concept, we assume that LDPC code and post-compensation signal processing technique are used in SSDs. Encouraged by the recent success of LDPC code in hard disk drive, the industry is very actively investigating its use for future NAND flash memory, and in the open literature the use of LDPC code in NAND flash memory has been recently discussed [4]. The postcompensation technique has been proposed in [5] as an effective way to compensate cell-to-cell interference. To facilitate the quantitative evaluation, based upon extensive open literature on flash memory devices, we develop an approximate NAND flash memory device model that quantitatively captures the P/E cycling effects and cell-to-cell interference. Using this memory cell device model and the SSD model [7] in DiskSim [8], we carry out extensive trace-based simulations, and the results clearly demonstrate that SSDs using quasi-EZ-NAND flash memory can achieve almost the same speed performance as SSDs using EZ-NAND flash memory, and both can reduce the average SSD response time (including both write and read request response

• Y. Pan, G. Dong, and T. Zhang are with the Electrical, Computer and Systems Engineering Department, Rensselaer Polytechnic Institute (RPI), Troy, NY, 12180.

E-mail: {yyangpan, dongguiqiang}@gmail.com, tzhang@ecse.rpi.edu.

• N. Xie is with Intel Corporation, Hillsboro, OR 97124.
E-mail: ningdexie@gmail.com.

Manuscript received 1 Mar. 2011; revised 26 Aug. 2011; accepted 5 Feb. 2012; published online 21 Feb. 2012.

Recommended for acceptance by L. Wang.

For information on obtaining reprints of this article, please send e-mail to: tc@computer.org, and reference IEEECS Log Number TC-2011-03-0138. Digital Object Identifier no. 10.1109/TC.2012.54.

time) by over 90 percent compared with SSDs using conventional NAND flash memory. In addition, we carry out application-specific integrated circuit (ASIC) design at 65 nm node for LDPC decoders and postcompensation module, and the results show that the use of quasi-EZ-NAND flash can reduce the silicon area overhead by up to 44 percent compared with the use of EZ-NAND flash memory.

2 BACKGROUND

2.1 Memory Erase and Program Basics

Each NAND flash memory cell is a floating gate transistor whose threshold voltage can be programmed by injecting certain amount of charges into the floating gate. Before one memory cell can be programmed, it must be erased and the threshold voltage of erased memory cells tends to have a wide Gaussian-like distribution [9]. Hence, we can approximately model the erased state as

$$p_e(x) = \frac{1}{\sigma_e \sqrt{2\pi}} e^{-\frac{(x-\mu_e)^2}{2\sigma_e^2}}, \quad (1)$$

where μ_e and σ_e are the mean and standard deviation of the erased state threshold voltage. Regarding memory program, a tight threshold voltage control is realized by incremental step pulse program (ISPP), i.e., all the memory cells on the same word-line are recursively programmed using a program-and-verify approach with a stair case program word-line voltage V_{pp} . Let ΔV_{pp} denote the incremental program step voltage. For the k th programmed state with the verify voltage $V_p^{(k)}$, ideally ISPP program results in a uniform threshold voltage distribution for each programmed state:

$$p_p^{(k)}(x) = \begin{cases} \frac{1}{\Delta V_{pp}}, & \text{if } V_p^{(k)} \leq x \leq V_p^{(k)} + \Delta V_{pp} \\ 0, & \text{else.} \end{cases} \quad (2)$$

Unfortunately, the above *ideal* memory cell threshold voltage distribution can be distorted in practice, mainly due to P/E cycling and cell-to-cell interference, which will be discussed in the remainder of this section.

2.2 Effects of P/E Cycling

Flash memory P/E cycling causes damage to the tunnel oxide of floating gate transistors in the form of charge trapping in the oxide and interface states [1], which directly results in threshold voltage shift and fluctuation and hence gradually degrades memory device noise margin. Major distortion sources include:

1. Electrons capture and emission events at charge trap sites near the interface developed over P/E cycling directly result in memory cell threshold voltage fluctuation, which is referred to as random telegraph noise (RTN) [10];
2. Interface trap recovery and electron detrapping [11] gradually reduce memory cell threshold voltage, leading to the data retention limitation.

RTN causes memory cell threshold voltage random fluctuation with exponential decay. Hence, we model the probability density function $p_r(x)$ of RTN-induced threshold voltage fluctuation as a symmetric exponential function [10]:

$$p_r(x) = \frac{1}{2\lambda_r} e^{-\frac{|x|}{\lambda_r}}. \quad (3)$$

Let N denote the P/E cycling number, λ_r scales with N in an approximate power-law fashion, i.e., $\lambda_r \propto N^\alpha$.

Interface trap recovery and electron detrapping processes approximately follow Poisson statistics [1], hence threshold voltage reduction due to interface trap recovery and electron detrapping can be approximately modeled as a Gaussian distribution $\mathcal{N}(\mu_d, \sigma_d^2)$. Both μ_d and σ_d^2 scale with N in an approximate

power-law fashion, and scale with the retention time t in a logarithmic fashion. Moreover, the significance of threshold voltage reduction is also proportional to the initial threshold voltage magnitude.

2.3 Cell-to-Cell Interference

In NAND flash memory, the threshold voltage shift of one floating gate transistor can influence the threshold voltage of its neighboring floating gate transistors through parasitic capacitance-coupling effect [12], which is referred to as cell-to-cell interference. Threshold voltage shift of a victim cell caused by cell-to-cell interference can be estimated as [12]:

$$F = \sum_k (\Delta V_t^{(k)} \cdot \gamma^{(k)}), \quad (4)$$

where $\Delta V_t^{(k)}$ represents the threshold voltage shift of one interfering cell which is programmed after the victim cell, and the coupling ratio $\gamma^{(k)} = \frac{C^{(k)}}{C_{total}}$ in which $C^{(k)}$ is the parasitic capacitance between the interfering cell and the victim cell and C_{total} is the total capacitance of the victim cell.

2.4 An Approximate Memory Device Model

Based on the above discussions, we can approximately model NAND flash memory device characteristics, using which we can simulate memory cell threshold voltage distribution and obtain memory raw storage reliability. Based upon (1) and (2), we can obtain the distortion-less threshold voltage distribution function $p_p(x)$. Recall that $p_r(x)$ denotes the RTN distribution function (see (3)), and let $p_{ar}(x)$ denote the threshold voltage distribution after incorporating RTN, which is obtained by convoluting $p_p(x)$ and $p_r(x)$, i.e.,

$$p_{ar}(x) = p_p(x) \otimes p_r(x). \quad (5)$$

Cell-to-cell interference is further incorporated based on (4). To capture inevitable process variability, we set both the vertical and diagonal coupling ratio γ_y and γ_{xy} as random variables with bounded Gaussian distributions:

$$p_c(x) = \begin{cases} \frac{c_c}{\sigma_c \sqrt{2\pi}} \cdot e^{-\frac{(x-\mu_c)^2}{2\sigma_c^2}}, & \text{if } |x - \mu_c| \leq w_c \\ 0, & \text{else,} \end{cases} \quad (6)$$

where μ_c and σ_c are the mean and standard deviation, and c_c is chosen to ensure the integration of this bounded Gaussian distribution equals to 1. We set $w_c = 0.1 \mu_c$ and $\sigma_c = 0.4 \mu_c$ in this work. Let p_{ac} denote the threshold voltage distribution after incorporating cell-to-cell interference, $p_t(x)$ denote the distribution of threshold voltage reduction during retention, the final threshold voltage distribution p_f is obtained as

$$p_f(x) = p_{ac}(x) \otimes p_t(x). \quad (7)$$

3 PROPOSED QUASI-EZ-NAND FLASH MEMORY DESIGN STRATEGY

In conventional design practice of SSDs, all the ECC and DSP functions are implemented in the controller as illustrated in Fig. 1a. As more powerful and complicated ECC and DSP techniques are being used, such a conventional design practice can result in significant SSD speed performance degradation, i.e., advanced ECC and DSP tend to demand fine-grained memory cell sensing, leading to much higher flash-to-controller data transfer latency and hence large SSD system speed performance degradation. Under the emerging EZ-NAND flash design strategy, all the ECC and DSP functions are embedded in each NAND flash chip. As a result, EZ-NAND flash memory chips always appear to be error-free to the external controller/host. As illustrated in Fig. 1b, in SSDs using

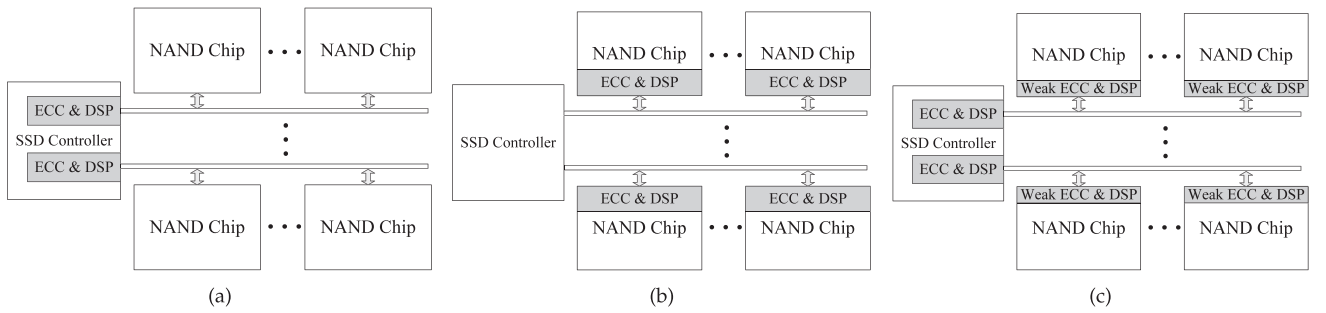


Fig. 1. Structures of SSDs using (a) conventional NAND flash memory chips, (b) EZ-NAND flash memory chips, and (c) proposed quasi-EZ-NAND flash memory chips.

EZ-NAND flash memory, all the ECC and DSP functions are off-loaded from the controller to all the individual flash memory chips, which can achieve much better system speed performance compared with conventional SSDs. However, as the ECC and DSP become increasingly sophisticated and induce higher silicon implementation cost, one EZ-NAND flash memory chip can be noticeably more expensive than its conventional NAND flash memory counterpart. As a result, large-capacity SSDs may not be able to afford the use of EZ-NAND flash memory chips.

In this work, we propose a quasi-EZ-NAND flash memory design strategy that can reduce the silicon cost overhead compared with EZ-NAND flash memory and meanwhile maintain almost the same SSD system speed performance. As illustrated in Fig. 1c, each quasi-EZ-NAND flash memory chip only incorporates relatively weak and hence less sophisticated ECC and DSP functions that can ensure data storage integrity with a certain probability, and the central SSD controller contains the full-strength ECC and DSP functions that are executed only when the weak ECC and DSP in quasi-EZ-NAND flash memory chips fail to recover the user data. The advantages of such a quasi-EZ-NAND flash memory design strategy can be intuitively justified as follows:

- *It naturally matches to the NAND flash memory cell wear-out dynamics.* From the discussions in Section 2, it is clear that NAND flash memory cell raw storage reliability gradually degrades with the P/E cycling: During the early lifetime of memory cells (i.e., the P/E cycling number N is relatively small), the aggregated P/E cycling effects are relatively less significant, which leads to a relatively large memory cell storage noise margin and hence good raw storage reliability (i.e., low raw storage bit error rate); since the aggregated P/E cycling effects scale with N in approximate power-law fashions, the memory cell storage noise margin and hence raw storage reliability gradually degrade as the P/E cycling number N increases. Given the target P/E cycling endurance limit (e.g., 10k P/E cycling), the employed ECC and DSP should ensure the storage integrity as the P/E cycling reaches the endurance limit. Therefore, in the presence of such memory cell wear-out dynamics, the weak and hence less sophisticated ECC and DSP may have a very high probability to ensure the system data storage integrity for most of the memory lifetime. This suggests that quasi-EZ-NAND flash memory chips can behave error-free for most of the memory lifetime, and hence can obtain almost the same speed improvement as using EZ-NAND flash memory chips.
- *It reduces the ECC and DSP silicon cost overhead.* By only incorporating weak and less sophisticated ECC and DSP, each quasi-EZ-NAND flash memory chip induces less silicon cost compared with its EZ-NAND flash memory counterpart. Meanwhile, since the quasi-EZ-NAND flash memory chips behave error-free most of the time, the

full-strength ECC and DSP functions in the SSD controller may not have to meet the system throughput requirement, which can be possibly leveraged to further reduce the SSD controller silicon implementation cost.

Fig. 2 further shows the operational data flow diagram of all the three scenarios discussed above. As shown in Figs. 2a and 2b, when conventional NAND and ideal EZ-NAND flash memory chips are used, all the ECC and DSP functions are executed in the central SSD controller and in each flash memory chip, respectively. In the context of the proposed quasi-EZ-NAND flash memory, ECC and DSP are carried out hierarchically at both flash memory chip and controller: each quasi-EZ-NAND flash memory chip always executes embedded weak ECC and DSP functions to recover the user data, and only when it fails the SSD controller carries out the full-strength ECC and DSP.

4 EVALUATION METHODOLOGY

In this study, we set that SSDs use LDPC code and use the postcompensation technique to compensate the cell-to-cell interference. Targeting at 4k-byte user data per page, we construct a regular rate-9/10 quasi-cyclic LDPC (QC-LDPC) code with the parity check matrix column weight of 4. The decoder uses the minimum-sum decoding algorithm [13] with up to eight decoding iterations. LDPC code decoder carries out soft-decision decoding and its error correction capability heavily depends on the finite word-length precision of the input: As we increase finite word-length precision of the input data, LDPC decoder can achieve stronger error correction capability but will occupy larger silicon area and consume more power. In this work, we set that the full-strength and weak LDPC decoder uses 5-bit and 1-bit input, respectively.

The basic idea of postcompensation is simple [5]: If we know the threshold voltage shift of interfering cells, we can estimate the corresponding cell-to-cell interference strength according to (4) and subsequently subtract it from the sensed threshold voltage of victim cells. To implement postcompensation signal processing, we have to sense the cells of both current wordline being read and its adjacent interfering wordline, and the memory sensing should be carried out with a finer granularity. This clearly leads to a longer memory sensing latency, and longer flash-to-controller data transfer latency if the finer-grained sensing results are sent to the SSD controller. We use “ $(m+n)$ -sensing” to denote the sensing scheme used in postcompensation, where each memory cell on the current wordline and adjacent interfering wordline is sensed using m and n bits, respectively. In this work, we set the maximum values of m and n as 5 and 4, respectively.

In this study, we consider 2 bits/cell NAND flash memory. We set normalized σ_e and μ_e of the erased state as 0.35 and 1.4, respectively. For the three programmed states, we set the normalized program step voltage ΔV_{pp} as 0.3, and the normalized verify voltages V_p as 2.55, 3.15, and 3.88, respectively. For the RTN distribution function $p_r(x)$, we set the parameter $\lambda_r = K_\lambda \cdot N^{0.5}$

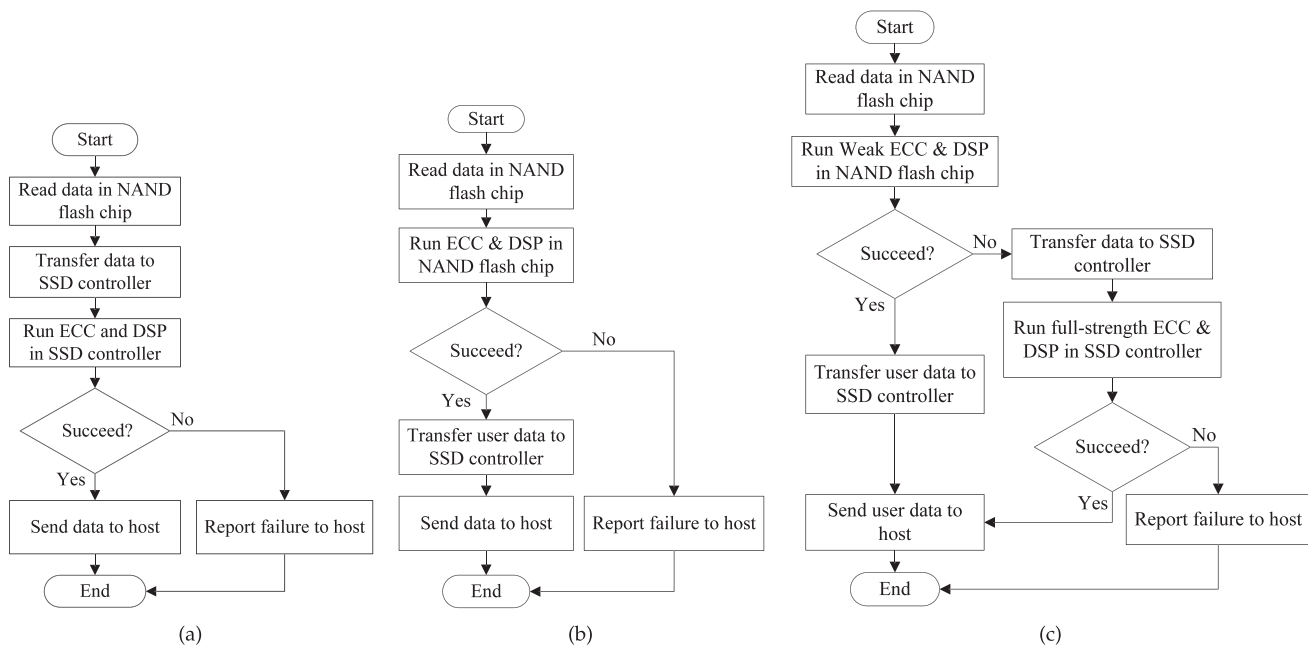


Fig. 2. Operational data flow diagrams of SSDs using (a) conventional NAND flash memory, (b) EZ-NAND flash memory, and (c) proposed quasi-EZ-NAND flash memory.

where $K_\lambda = 5 \times 10^{-4}$. Regarding cell-to-cell interference, according to [14], we set the means of γ_y and γ_{xy} as 0.08 and 0.0048, respectively. For the function $\mathcal{N}(\mu_d, \sigma_d^2)$ to capture interface trap recovery and electron detrapping, according to [1], we set that μ_d scale with $N^{0.5}$ and σ_d^2 scales with $N^{0.6}$, and both scale with $\ln(1 + t/t_0)$, where t denote the memory retention time and t_0 is an initial time and can be set as 1 hour. Since both μ_d and σ_d^2 also depend on the initial threshold voltage, we set that both approximately scale $K_s(x - x_0)$, where x is the initial threshold voltage, and x_0 and K_s are constants. Therefore, we have

$$\begin{cases} \mu_d = K_s(x - x_0)K_dN^{0.5}\ln(1 + t/t_0) \\ \sigma_d^2 = K_s(x - x_0)K_mN^{0.6}\ln(1 + t/t_0), \end{cases} \quad (8)$$

where we set $K_s = 0.388$, $x_0 = 1.4$, $K_d = 2.4 \times 10^{-4}$, and $K_m = 2.4 \times 10^{-6}$ by fitting the measurement data presented in [1]. Targeting at page error rate (PER) below 10^{-15} , we estimate that the use of full-strength 5-bit-precision LDPC decoding and postcompensation with (5 + 4)-sensing can achieve P/E cycling endurance of 10k with retention of 10 years. For the three SSD implementation scenarios, we have:

- *SSDs using conventional NAND flash memory.* The SSD controller contains a set of 5-bit-precision LDPC decoders and postcompensation circuits with (5 + 4)-sensing, and each set handles one SSD channel.
- *SSDs using EZ-NAND flash memory.* Each flash memory chip has its own set of 5-bit-precision LDPC decoder and postcompensation module with (5 + 4)-sensing, and the SSD controller does not implement any ECC and DSP functions.
- *SSDs using quasi-EZ-NAND flash memory.* Each flash memory chip has its own set of 1-bit-precision LDPC decoder and postcompensation circuits with (5 + 4)-sensing, and the SSD controller contains a set of 5-bit-precision LDPC decoders. We keep the full-strength postcompensation function in each flash chip since it consumes much less silicon cost compared with LDPC decoder.

Since LDPC decoder and postcompensation demand finer-grained memory sensing and the sensing precision directly affects

the memory sensing latency and flash-to-controller data transfer latency, we need to develop appropriate memory sensing strategies. Intuitively, we can use a progressive memory sensing strategy to reduce the latency cost, i.e., we always start with an initial sensing configuration with less precision (e.g., (3 + 2)-sensing), based on which we carry out postcompensation and LDPC decoding, and only if LDPC decoding fails, we progressively increase the sensing precision and retry the decoding until LDPC decoding succeeds. As long as the initial sensing configuration can ensure sufficiently low LDPC decoding failure rate (e.g., $10^{-2} \sim 10^{-3}$), the savings gained from less memory sensing precision can easily offset the extra latency due to the less frequent fail-and-retry operations. In addition, since NAND flash memory device wears out gradually with the P/E cycling, we can dynamically adjust the initial sensing configuration adaptive to the P/E cycling number. Therefore, we use a *P/E-cycling-aware progressive memory sensing* strategy in our evaluation.

Using the SSD module [7] in DiskSim [8], we carry out trace-based simulations to evaluate these design strategies under realistic workloads including Postmark [7], Finance1 and Finance2 from [15], and Trace1 from [16]. Each NAND flash memory chip contains two dies that share an 8-bit I/O bus and a number of common control signals, and each die contains four planes and each plane contains 2,048 blocks. Following the ONFI 2.0 specification [17], we set the NAND flash memory chip interface bus bandwidth as 133 MB/s. We set the NAND flash memory program latency as 800 μ s and erase latency as 3 ms, and due to the fully serial nature of memory sensing (i.e., the m -bit sensing latency is roughly proportional to $2^m - 1$), we set the latency for 2-bit sensing, 3-bit sensing, 4-bit sensing, and 5-bit sensing as 25.7, 60, 128.6, and 265.7 μ s, respectively.

5 SIMULATION RESULTS

5.1 Advantages of Integrating P/E Cycling Awareness in Progressive Memory Sensing

As we pointed out in Section 4, in order to exploit the NAND flash memory cell wear out dynamics, we dynamically adjust the initial sensing configuration in progressive memory sensing adaptive to P/E cycling number. In this work, we first quantitatively evaluate the potential gains when integrating the P/E cycling awareness in

TABLE 1
Five-bit LDPC Decoding Page Error Rate under Different Sensing Configurations and P/E Cycling Numbers

P/E cycling number	2000	4000	6000	8000	10000
(2+1)-sensing	1	1	1	1	1
(3+1)-sensing	3.7E-3	0.16	1	1	1
(3+2)-sensing	1.1E-7	7.6E-5	2.8E-3	0.11	0.94
(4+1)-sensing	< E-7	< E-7	< E-7	5.6E-6	1.3E-4

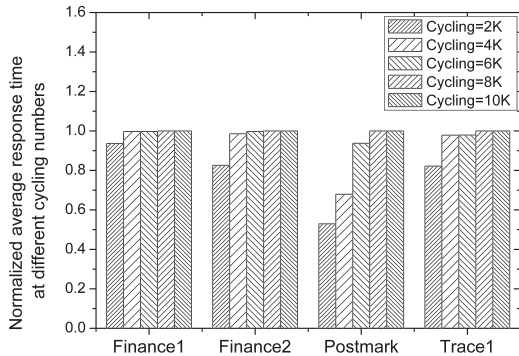


Fig. 3. Simulated normalized average response time when using P/E-cycling-aware progressive memory sensing and each SSD channel contains four flash chips.

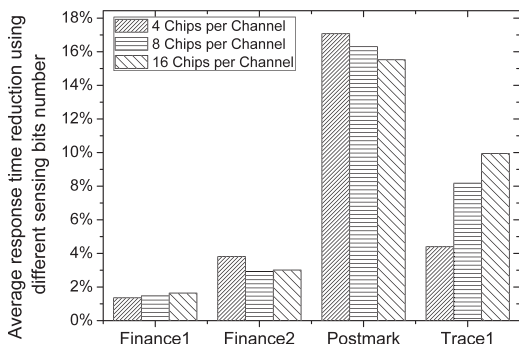


Fig. 4. Comparison of average response time reduction by integrating P/E cycling awareness into the progressive memory sensing.

progressive memory sensing. Here, we use SSDs with conventional NAND flash memory chips as a test vehicle. To set up the appropriate memory sensing configurations, we carry out finite precision C simulations and obtain the 5-bit-precision LDPC decoding page error rate under different sensing configurations and P/E cycling numbers as listed in Table 1.

Therefore, given the target 10k P/E cycling endurance limit, if we adjust the initial sensing configuration every 2,000 P/E cycles, we should use (3 + 1)-sensing, (3 + 2)-sensing, (3 + 2)-sensing, (4 + 1)-sensing, and (4 + 1)-sensing during the first, second, third, fourth, and fifth 2,000 P/E cyclings, respectively. For the purpose of comparison, we set the baseline scenario as the case of using progressive memory sensing without any adaptation to P/E cycling. Hence, the initial memory sensing configuration in the baseline scenario is fixed as (4 + 1)-sensing throughout the memory lifetime.

We carry out DiskSim-based simulations to evaluate the average SSD response time (including both write and read request response time) with and without P/E cycling awareness in the progressive memory sensing. We use the first-come first-serve (FCFS) scheduling policy in the simulations. Since the SSD channel parallelism (i.e., the number of NAND flash memory chips on each SSD channel) can affect the SSD speed performance, we consider the scenarios when each SSD channel has four NAND flash memory chips. The results as shown in Fig. 3 clearly suggest the advantage of integrating P/E cycling awareness.

TABLE 2
One-bit LDPC Decoding Page Error Rate under Different Sensing Configurations and P/E Cycling Numbers

P/E cycling number	2000	4000	6000	8000	10000
(3+1)-sensing	0.805	1	1	1	1
(3+2)-sensing	0.25	1	1	1	1
(4+1)-sensing	< E-7	1E-4	5.6E-2	0.25	1
(4+2)-sensing	< E-7	< E-7	6.3E-7	1.5E-5	9.3E-3

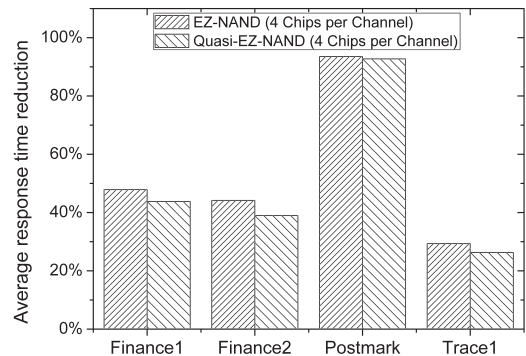


Fig. 5. Average SSD response time reduction.

Based on the above simulation results, we can obtain the average response time reduction over the baseline scenario with fixed initial (4 + 1)-sensing as shown in Fig. 4. Intuitively, those traces with higher read request ratios (e.g., postmark and Trace1) tend to benefit more from the integration of P/E cycling awareness, as demonstrated in Fig. 4.

5.2 SSD Speed Performance

We further carry out trace-based simulations to evaluate and compare the speed performance of SSDs using conventional NAND flash memory, EZ-NAND flash memory, and quasi-EZ-NAND flash memory, respectively. The P/E-cycling-aware progressive memory sensing strategy is used in all of the simulations. For SSDs using conventional NAND flash memory and EZ-NAND flash memory, where 5-bit-precision LDPC decoding is always executed, we obtain the initial sensing configurations based upon Table 1 in the above, i.e., we should use (3 + 1)-sensing, (3 + 2)-sensing, (3 + 2)-sensing, (4 + 1)-sensing, and (4 + 1)-sensing as the initial sensing configuration during the first, second, third, fourth, and fifth 2,000 P/E cyclings, respectively. For SSDs using quasi-EZ-NAND flash memory, where 1-bit-precision LDPC decoding is executed first, we carry out corresponding finite precision C simulations and obtain the 1-bit-precision LDPC decoding page error rate under different sensing configurations and P/E cycling numbers as listed in Table 2. Accordingly, we should use (3 + 2)-sensing, (4 + 1)-sensing, (4 + 1)-sensing, (4 + 1)-sensing, and (4 + 2)-sensing as the initial sensing configuration during the first, second, third, fourth, and fifth 2,000 P/E cyclings, respectively.

Using SSDs with conventional NAND flash memory as the baseline scenario, Fig. 5 shows the simulated average SSD response time reduction when each SSD channel contains four NAND flash memory chips. As pointed out in the above, the proposed quasi-EZ-NAND flash memory design strategy aims to achieve almost the same SSD speed performance as the ideal EZ-NAND flash memory at less silicon cost. The results shown in Fig. 5 clearly demonstrate that SSDs using either EZ-NAND flash memory or quasi-EZ-NAND flash memory have almost the same average response time. Compared with the baseline scenario, SSDs using either EZ-NAND flash memory or quasi-EZ-NAND flash memory can reduce the average response time by up to 92, 94, and 96 percent, when each SSD channel contains 4, 8, and 16 memory chips. The results also

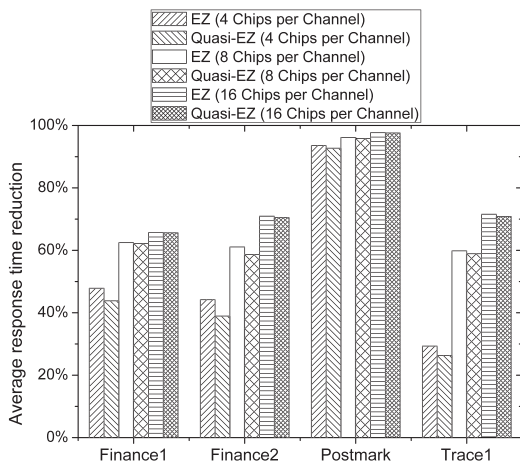


Fig. 6. Average response time reduction over the baseline scenario with fixed initial (4 + 1)-sensing configuration in progressive memory sensing.

show that using more memory chips on each SSD channel can directly improve the SSD speed performance, which can be intuitively justified.

In addition, Fig. 6 shows SSD speed improvement when the fixed initial (4 + 1)-sensing is used in the baseline scenario. The results show that, by integrating P/E cycling awareness in the progressive memory sensing, we can further improve the average response time reduction gained from using either EZ-NAND or quasi-EZ-NAND design strategy up to about 93, 96, and 98 percent when each SSD channel contains 4, 8 and 16 memory chips, respectively.

5.3 Silicon Area of EZ-NAND versus Quasi-EZ-NAND

The above simulation results show that the proposed quasi-EZ-NAND flash memory design strategy indeed can maintain almost the same speed performance as the EZ-NAND flash memory design strategy. To evaluate the silicon cost advantage of quasi-EZ-NAND over EZ-NAND, we further carry out ASIC design using 65 nm CMOS standard cell and SRAM libraries, where Synopsys tools are used throughout the design hierarchy down to place and route. The LDPC decoder is implemented using the partially parallel decoder architecture presented in [18]. Results show that, to achieve 2 Gbps decoding throughput, each 5-bit-precision LDPC decoder occupies 1.47 mm² (0.66 mm² of SRAM and 0.81 mm² of logic), each 1-bit-precision LDPC decoder occupies 0.61 mm² (0.40 mm² of SRAM and 0.21 mm² of logic), and each postcompensation module occupies 0.27 mm².

Fig. 7 shows the aggregated silicon area of LDPC decoders and postcompensation modules under different SSD channel parallelism when SSDs contain five channels. The results show that, by using quasi-EZ-NAND flash memory instead of EZ-NAND flash memory, we can reduce the silicon cost by 28.3, 38.8, and 44.1 percent when each SSD channel contains 4, 8, and 16 NAND flash memory chips. Therefore, the results above clearly demonstrate that, compared with EZ-NAND flash memory, our proposed quasi-EZ-NAND flash memory design strategy can noticeably reduce the silicon cost while maintaining almost the same SSD speed performance.

6 RELATED WORK

As technology continues to scale down, future NAND flash memories demand the use of more powerful and sophisticated ECC and DSP to ensure the data storage integrity. Maeda and Kaneko [4] proposed to apply LDPC in future MLC NAND flash memories. The industry is developing EZ-NAND flash memory

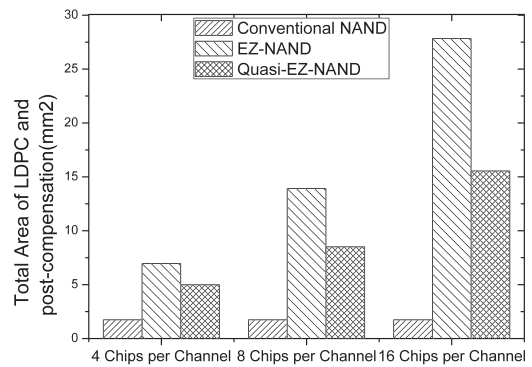


Fig. 7. Aggregated silicon area of LDPC decoders and postcompensation modules under different SSD channel parallelism when SSDs contain five channels.

products that aim to remove the burden of the host controller and to improve the performance of the system. Carla [19] demonstrated the potential advantages of using EZ-NAND flash memory and also compared the performance gain of copyback enabled by EZ-NAND with conventional NAND without copyback. Feeley [3] proposed emerging architectures of EZ-NAND and compared the speed performance of conventional NAND and EZ-NAND. However, for high-capacity SSDs that contains many NAND flash memory chips, the use of EZ-NAND flash memory chips can result in nonnegligible extra silicon cost.

7 CONCLUSION

This paper presents a quasi-EZ-NAND flash memory design strategy that can enable the economic use of powerful ECC and DSP functions in future large-capacity SSDs at low silicon cost overhead. Strategy hierarchically distributes ECC and DSP functions on both NAND flash memory chips and SSD controller. Compared with the emerging EZ-NAND design strategy, it can maintain almost the same speed performance while noticeably reducing silicon cost overhead. Simulation results show that SSD using quasi-EZ-NAND flash can maintain the same speed as SSDs using EZ-NAND flash and both can reduce the average SSD response time by over 90 percent compared with SSDs using conventional NAND flash. ASIC design results demonstrate that, compared with the case of using EZ-NAND flash, the use of quasi-EZ-NAND can reduce the silicon cost overhead by up to 44 percent.

ACKNOWLEDGMENTS

This material is based in part upon work supported by US National Science Foundation (NSF) under Grant Number CCF-0937794.

REFERENCES

- [1] N. Mielke, H. Belgal, I. Kalastirsky, P. Kalavade, A. Kurtz, Q. Meng, N. Righos, and J. Wu, "Flash EEPROM Threshold Instabilities due to Charge Trapping During Program/Erase Cycling," *IEEE Trans. Device and Materials Reliability*, vol. 4, no. 3, pp. 335-344, Sept. 2004.
- [2] K. Kim et al., "Future Memory Technology: Challenges and Opportunities," *Proc. Int'l Symp. VLSI Technology, Systems and Applications*, pp. 5-9, Apr. 2008.
- [3] P. Feeley, "Nand Flash Scaling is EZ," *Proc. Flash Memory Summit*, Aug. 2010.
- [4] Y. Maeda and H. Kaneko, "Error Control Coding for Multilevel Cell Flash Memories Using Nonbinary Low-Density Parity-Check Codes," *Proc. 24th IEEE Int'l Symp. Defect and Fault Tolerance in VLSI Systems*, pp. 367-375, Oct. 2009.
- [5] G. Dong, S. Li, and T. Zhang, "Using Data Post-Compensation and Pre-Distortion to Tolerate Cell-to-Cell Interference in MLC NAND Flash Memory," *Trans. Circuits and Systems-I: Regular Papers*, vol. 57, pp. 2718-2728, 2010.
- [6] GlobeNewswire, <http://www.globenewswire.com/newsroom/news.html?d=180990>, 2012.

- [7] N. Agrawal, V. Prabhakaran, T. Wobber, J.D. Davis, M. Manasse, and R. Panigrahy, "Design Tradeoffs for SSD Performance," *Proc. USENIX Ann. Technical Conf.*, pp. 57-70, 2008.
- [8] J.S. Bucy, J. Schindler, S.W. Schlosser, and G.R. Ganger, "The DiskSim Simulation Environment Version 4.0 Reference Manual," Technical Report CMU-PDL-08-101, Carnegie Mellon Univ., Parallel Data Laboratory, May 2008.
- [9] K. Takeuchi, T. Tanaka, and H. Nakamura, "A Double-Level-Vth Select Gate Array Architecture for Multilevel NAND Flash Memories," *IEEE J. Solid-State Circuits*, vol. 31, no. 4, pp. 602-609, Apr. 1996.
- [10] C.M. Compagnoni, M. Ghidotti, A.L. Lacaita, A.S. Spinelli, and A. Visconti, "Random Telegraph Noise Effect on the Programmed Threshold-Voltage Distribution of Flash Memories," *IEEE Electron Device Letters*, vol. 30, no. 9, pp. 984-986, Sept. 2009.
- [11] N. Mielke, H.P. Belgal, A. Fazio, Q. Meng, and N. Righos, "Recovery Effects in the Distributed Cycling of Flash Memories," *Proc. IEEE Int'l Reliability Physics Symp.*, pp. 29-35, 2006.
- [12] J.-D. Lee, S.-H. Hur, and J.-D. Choi, "Effects of Floating-Gate Interference on NAND Flash Memory Cell Operation," *IEEE Electron Device Letters*, vol. 23, no. 5, pp. 264-266, May 2002.
- [13] J. Chen, A. Dholakia, E. Eleftheriou, M.P.C. Fossorier, and X.-Y. Hu, "Reduced-Complexity Decoding of LDPC Codes," *IEEE Trans. Comm.*, vol. 53, no. 8, pp. 1288-1299, Aug. 2005.
- [14] K. Prall, "Scaling Non-Volatile Memory Below 30 nm," *Proc. IEEE Non-Volatile Semiconductor Memory Workshop*, pp. 5-10, Aug. 2007.
- [15] Storage Performance Council, "SPC Trace File Format Specification," technical report, Revision 1.0.1, <http://traces.cs.umass.edu/index.php/Storage/Storage>, 2002.
- [16] C. Dirik and B. Jacob, "The Performance of PC Solid-State Disks (SSDs) as a Function of Bandwidth, Concurrency, Device Architecture, and System Organization," *SIGARCH Computer Architecture News*, vol. 37, no. 3, pp. 279-289, 2009.
- [17] Hynix Semiconductor, et al., "Open NAND Flash Interface Specification," technical report, 2009.
- [18] H. Zhong, W. Xu, N. Xie, and T. Zhang, "Area-Efficient Min-Sum Decoder Design for High-Rate Quasi-Cyclic Low-Density Parity-Check Codes in Magnetic Recording," *IEEE Trans. Magnetics*, vol. 43, no. 12, pp. 4117-4122, Dec. 2007.
- [19] C. Lay, "Improving NAND Performance Using Upcoming Feature," *Proc. Flash Memory Summit*, Aug. 2010.

► For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.