

# Using Data Postcompensation and Predistortion to Tolerate Cell-to-Cell Interference in MLC NAND Flash Memory

Guiqiang Dong, *Student Member, IEEE*, Shu Li, and Tong Zhang, *Senior Member, IEEE*

**Abstract**—With the appealing storage-density advantage, multilevel-per-cell (MLC) NAND Flash memory that stores more than 1 bit in each memory cell now largely dominates the global Flash memory market. However, due to the inherent smaller noise margin, the MLC NAND Flash memory is more subject to various device/circuit variability and noise, particularly as the industry is pushing the limit of technology scaling and a more aggressive use of MLC storage. Cell-to-cell interference has been well recognized as a major noise source responsible for raw-memory-storage reliability degradation. Leveraging the fact that cell-to-cell interference is a deterministic data-dependent process and can be mathematically described with a simple formula, we present two simple yet effective data-processing techniques that can well tolerate significant cell-to-cell interference at the system level. These two techniques essentially originate from two signal-processing techniques being widely used in digital communication systems to compensate communication-channel intersymbol interference. The effectiveness of these two techniques have been well demonstrated through computer simulations and analysis under an information theoretical framework, and the involved design tradeoffs are discussed in detail.

**Index Terms**—Cell-to-cell interference, NAND flash memory, postcompensation, predistortion

## I. INTRODUCTION

OVER THE past several years, NAND Flash memory has become the fastest growing segment in the global semiconductor industry and the essential driver for technology scaling. NAND Flash memory is being used in increasingly diverse applications to realize high-capacity nonvolatile data storage, from consumer electronics to personal and enterprise computers. The continuous growth of NAND Flash memory storage density has been mainly driven by aggressive technology scaling, e.g., NAND Flash memories with 43-nm and sub-35-nm CMOS technologies have been recently reported in [1] and [2], respectively. Aside from technology scaling, the multilevel-per-cell (MLC) technique, i.e., to store more

than 1 bit in each memory cell (or floating-gate MOS transistor) by programming the cell threshold voltage into one of  $l > 2$  voltage windows, has been widely used to further improve the NAND Flash memory storage density. Because of its obvious advantages in terms of storage density and, hence, cost, MLC NAND Flash memory now largely dominates the global Flash memory market. In current design practice, most MLC NAND Flash memories store 2 bits per cell, while 3- and even 4-bit-per-cell NAND Flash memories have been recently reported in the open literature [2]–[6].

In NAND Flash memory, each memory cell is a floating-gate transistor and an  $l$ -level-per-cell data storage is realized by programming the threshold voltage of each floating-gate transistor to one of  $l$  nonoverlapping voltage windows. Due to process variability and various effects such as cell-to-cell interference, program/read disturbance, and charge leakage [7], [8], adjacent threshold-voltage distribution windows may become very close to each other or even overlap, leading to nonnegligible raw bit error rates (BERs). To ensure the overall data-storage integrity, we should address these issues at both device/circuit level, which aims to *minimize* the process variability, noise, and interference through device and circuit optimizations, and system level, which aims to *tolerate* these effects through system-level data-processing and coding techniques. As the technology scaling and MLC concept are being pushed toward their limits, it is evident that appropriate system-level solutions will play an increasingly important role. In current design practice, error correction codes (ECCs), particularly BCH codes, are being used as the almost only system-level technique to tolerate the increasing raw-memory BERs. However, the use of stronger ECC tends to demand the storage of more coding redundancy, which inevitably degrades the effective memory storage capacity. Moreover, it will make the ECC encoder and decoder implementation more complicated, leading to higher silicon and energy consumption cost in NAND Flash memory controllers.

It has been well recognized that cell-to-cell interference has become the major source for floating-gate threshold-voltage distribution distortion [8]–[10]. As elaborated in Section II-B, cell-to-cell interference is a deterministic data-dependent process, which can be mathematically described with a simple formula. In fact, it is essentially the same as the communication-channel intersymbol interference encountered in many digital communication systems. In digital communication, two types of signal-processing techniques, i.e., signal equalization and signal predistortion [11]–[15], are typically used to handle

Manuscript received November 16, 2009; revised February 10, 2010; accepted March 04, 2010. Date of publication May 06, 2010; date of current version October 08, 2010. This paper was recommended by Associate Editor J. Pineda de Gyvez.

G. Dong and T. Zhang are with the Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY 12180-3590 USA (e-mail: dongguiqiang@gmail.com; tong.zhang@ieee.org).

S. Li is with Marvell Semiconductor, Inc., Sunnyvale, CA 94086 USA (e-mail: lishunys@gmail.com).

Digital Object Identifier 10.1109/TCSI.2010.2046966

intersymbol interference. This directly motivates us to investigate the potential of applying the same concepts to address cell-to-cell interference in NAND Flash memory.

In this paper, we present two such data-processing techniques that can well tolerate cell-to-cell interference and hence reduce the induced bit errors. Therefore, they can enable the use of weaker ECCs with less coding redundancy to ensure data-storage integrity, which leads to higher effective memory storage capacities. The first technique, called data postcompensation, aims to estimate and subtract cell-to-cell interference when we read data from NAND Flash memories, which essentially follows the concept of signal equalization in digital communication. The second technique, called data predistortion, aims to predict cell-to-cell interference and accordingly predistort the data when we write data to NAND Flash memories, which essentially follows the concept of signal predistortion in digital communication. We discuss these two data-processing techniques in detail and demonstrate their effectiveness through extensive computer simulations and analysis under the information theoretical framework. We also leverage these two techniques to develop tight lower bounds of NAND Flash memory information theoretical storage efficiency in the presence of significant cell-to-cell interference. We also discuss the involved design tradeoffs and overhead inherent in these two data-processing techniques and point out that applications with different data access patterns may prefer different data-processing techniques. Finally, we note that recent patents (e.g., see [16] and its references) also proposed to apply these two concepts in digital communication to improve the tolerance to cell-to-cell interference.

The remainder of this paper is organized as follows. Section II reviews the basics of NAND Flash memory and cell-to-cell interference and presents a memory-cell threshold-voltage distribution model being used throughout this study. In Section III, we elaborate on the two data-processing techniques and demonstrate their effectiveness through information theoretical analysis and computer simulations. Conclusions are drawn in Section IV.

## II. BACKGROUND

### A. NAND Flash Memory Basics

Each NAND Flash memory cell is a floating-gate transistor whose threshold voltage can be configured (or programmed) by injecting a certain amount of charges into the floating gate. Hence, data storage in an  $l$ -level-per-cell NAND Flash memory is realized by programming the threshold voltage of each memory cell into one of  $l$  nonoverlapping voltage windows. Before one memory cell can be programmed, it must be erased (i.e., remove the charges in the floating gate, which sets its threshold voltage to the lowest voltage window). A tight threshold-voltage control is typically realized by using an incremental step pulse program, i.e., a program-and-verify approach with a staircase program voltage  $V_{pp}$  [17], [18], as shown in Fig. 1, where  $\Delta V_{pp}$  is the incremental program step voltage. Under such a program-and-verify programming strategy, each programming state (except the erased state) associates with a verify voltage that is used in

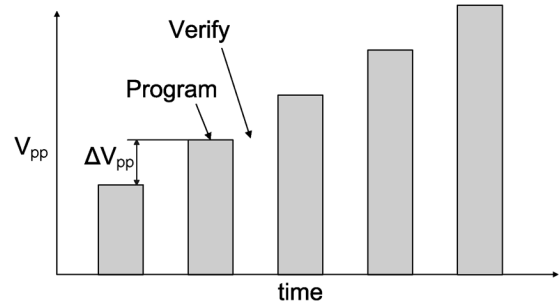


Fig. 1. Control-gate voltage pulses in program-and-verify operations.

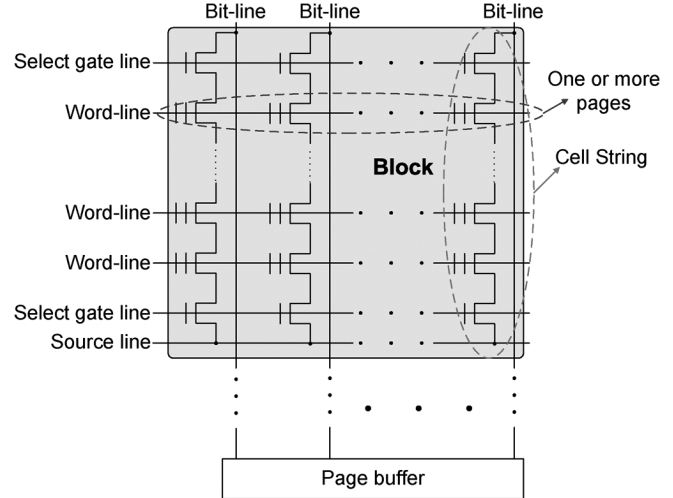


Fig. 2. Illustration of NAND Flash memory structure.

the verify operations. Denote the verify voltage of the target programming state as  $V_p$ . During each program-and-verify cycle, the floating-gate threshold voltage  $V_t$  is first boosted by up to  $\Delta V_{pp}$  and then compared with  $V_p$ . If  $V_t$  is still lower than  $V_p$ , the program-and-verify iteration will continue; otherwise, the corresponding bit line will be configured so that further programming of this cell is disabled.

NAND Flash memory cells are organized in an array  $\rightarrow$  block  $\rightarrow$  page hierarchy, as shown in Fig. 2, where one NAND Flash memory array is partitioned into many blocks and each block contains a certain number of pages. Within one block, each memory-cell string (as shown in Fig. 2) typically contains 16–64 memory cells. All the memory cells within the same block must be erased at the same time. Data are programmed and fetched in the unit of page, where the page size ranges from 512-B to 8-kB user data in current design practice. All the memory-cell blocks share the bit lines and an on-chip page buffer that holds the data being programmed or fetched. Modern NAND Flash memories use either even/odd bit-line structure [19], [20] or all-bit-line structure [21], [22]. In the even/odd bit-line structure, even and odd bit lines are interleaved along each word line and are alternatively accessed. Hence, each pair of even and odd bit lines can share peripheral circuits such as sense amplifiers and buffers, leading to less silicon cost of peripheral circuits. In the all-bit-line structure, all the bit lines are accessed at the same time, which aims to trade peripheral-circuit silicon cost for better immunity to cell-to-cell interference (as elaborated later in Section II-B). Moreover, a relatively simple voltage-sensing scheme can be used in even/odd bit-line

structure, while a current-sensing scheme must be used in the all-bit-line structure. For MLC NAND Flash memory, all the bits stored in one cell belong to different pages, which can be either simultaneously programmed at the same time, referred to as full-sequence programming, or sequentially programmed at different times, referred to as multipage programming. Since full-sequence programming can achieve a higher programming throughput but incur more severe cell-to-cell interference, we mainly consider the full-sequence programming strategy in this paper, i.e., under the even/odd bit-line structure, cells in even bit line, referred to as even cells, are programmed first with the full-sequence programming, and then cells in odd bit line, referred to as odd cells, are programmed with the full-sequence programming.

### B. Cell-to-Cell Interference

In NAND Flash memory, the threshold-voltage shift of one floating-gate transistor can change the threshold voltages of its neighboring floating-gate transistors through parasitic capacitance-coupling effect [23]. This is referred to as cell-to-cell interference, which clearly tends to result in data-storage reliability degradation. As the technology continues to scale down and, hence, adjacent cells become closer, the parasitic coupling capacitance between adjacent cells continues to increase and results in increasingly severe cell-to-cell interference. Recent studies [8]–[10] have clearly identified cell-to-cell interference as the major challenge for future NAND Flash memory scaling.

According to [23], the threshold-voltage change of a victim cell due to cell-to-cell interference can be estimated as

$$F = \sum_k \left( \Delta V_t^{(k)} \cdot \gamma^{(k)} \right) \quad (1)$$

where  $\Delta V_t^{(k)}$  represents the threshold-voltage shift of the  $k$ th interfering cell which is programmed after the victim cell and the coupling ratio  $\gamma^{(k)}$  is defined as

$$\gamma^{(k)} = \frac{C^{(k)}}{C_{\text{total}}} \quad (2)$$

where  $C^{(k)}$  is the parasitic capacitance between the interfering cell and the victim cell and  $C_{\text{total}}$  is the total capacitance of the victim cell. Clearly, one cell can only be interfered by its neighbors which are programmed after this cell has been programmed. In practice, only the cell-to-cell interference from immediately adjacent cells are considered because parasitic coupling capacitance quickly diminishes as the cell-to-cell distance increases.

Let us consider the case when the even/odd bit-line structure is being used, as shown in Fig. 3, and assume that cells on even bit lines are programmed first along each word line. Meanwhile, we note that, only after all the cells on one word line have been programmed, the cells on the next word line can be programmed when the full-sequence programming strategy is being used. Hence, each even cell has five interfering neighbors (i.e., two adjacent odd cells on the same word line and three adjacent cells on the next word line), while each odd cell only has three interfering neighbors (i.e., three adjacent cells on the next word line). Therefore, we should consider the effects of cell-to-cell interference for even and odd cells separately when the even/odd

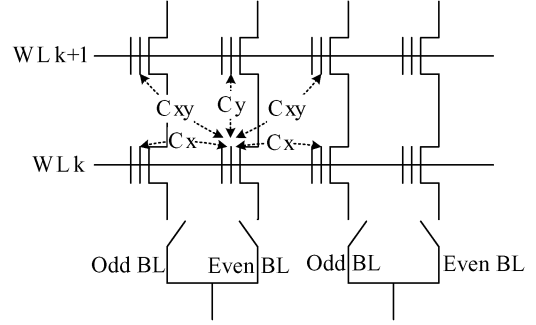


Fig. 3. Illustration of even/odd bit-line structure and parasitic coupling capacitances among adjacent cells.

bit-line structure is being used. When the all-bit-line structure is being used, the cells are subject to the same cell-to-cell interference as those odd cells in even/odd bit-line structure. Therefore, this paper only considers the scenario using the even/odd bit-line structure.

### C. Cell-Threshold-Voltage Distribution Model

In this section, we present a NAND Flash memory-cell threshold-voltage distribution model that is used in all the quantitative studies throughout this paper. As pointed out earlier, before one Flash memory cell is programmed, it must be erased first, leading to the lowest threshold voltage. The threshold voltage of an erased state tends to have a wide Gaussian-like distribution [24]. Hence, we model the threshold-voltage distribution of the erased state as

$$p_e(x) = \frac{1}{\sigma_e \sqrt{2\pi}} \exp \left\{ -\frac{(x-V_e)^2}{2\sigma_e^2} \right\} \quad (3)$$

where  $V_e$  and  $\sigma_e$  are the mean and standard deviation of the erased-state threshold-voltage distribution. For the other programmed states, as pointed out earlier, each memory cell is programmed using an iterative program-and-verify approach with a step voltage of  $\Delta V_{\text{pp}}$ . Due to the process variability, given the word-line programming step voltage during each program-and-verify cycle, different cells may experience different threshold-voltage boosts with up to  $\Delta V_{\text{pp}}$ . Since the verify voltage  $V_p$  is used to determine whether the programming should continue or stop, the threshold voltage of each programmed state tends to have a uniform distribution over  $[V_p, V_p + \Delta V_{\text{pp}}]$  with the width of  $\Delta V_{\text{pp}}$ . However, because of the inevitable noise at device and circuit levels, charge leakage, and cell-to-cell interference, such an ideal uniform threshold-voltage distribution is subject to a certain distortion. We assume that the aggregate effect of all the other distortions, except cell-to-cell interference, tend to add two symmetric Gaussian tails on both sides of the uniform distribution part, as shown in Fig. 4.

Let  $P_0$  and  $P_1$  denote the probabilities of the uniform distribution and Gaussian tails, respectively. We have the overall PDF  $p_p(x)$  as

$$p_p(x) = \begin{cases} \frac{c_p}{\sigma_p \sqrt{2\pi}} e^{-\frac{(x-V_p)^2}{2\sigma_p^2}}, & \text{if } x < V_p \\ \frac{c_p}{\sigma_p \sqrt{2\pi}} e^{-\frac{(x-V_p-\Delta V_{\text{pp}})^2}{2\sigma_p^2}}, & \text{if } x \geq V_p + \Delta V_{\text{pp}} \\ \frac{c_p}{\sigma_p \sqrt{2\pi}}, & \text{else} \end{cases} \quad (4)$$

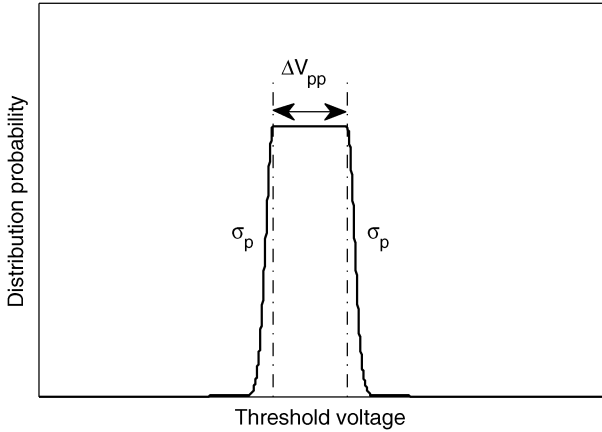


Fig. 4. Threshold-voltage distribution of programmed states in the absence of cell-to-cell interference.

where  $V_p$  is the verify voltage being used for the target programmed state. For the two parameters, i.e., the constant  $c_p$  and the standard deviation  $\sigma_p$  of the Gaussian tails, we can derive one parameter if another one is given by solving  $P_0 + P_1 = \int_{-\infty}^{+\infty} p_p(x) dx = 1$ .

The occurrence of cell-to-cell interference will further distort the threshold-voltage distribution shown in Fig. 4, which can be estimated using the quantitative parasitic capacitance-coupling effect as shown in (1). According to [9] and [25], we set  $E\{\gamma_x\} : E\{\gamma_y\} : E\{\gamma_{xy}\} = 0.1 : 0.08 : 0.006$ , where  $E\{*\}$  represents the mean and  $\gamma_x$ ,  $\gamma_y$ , and  $\gamma_{xy}$  are coupling ratios in three corresponding directions. To study a wide range of cell-to-cell interference strength, we introduce a parameter  $s$  called *cell-to-cell coupling strength factor* and have  $E\{\gamma_x\} = 0.1 s$ ,  $E\{\gamma_y\} = 0.08 s$ , and  $E\{\gamma_{xy}\} = 0.06 s$ . Due to the process variability, we assume that all the coupling ratios  $\gamma_x$ ,  $\gamma_y$ , and  $\gamma_{xy}$  follow bounded Gaussian distributions as

$$p_r(x) = \begin{cases} \frac{c_r}{\sigma_r \sqrt{2\pi}} \cdot e^{-\frac{(x-\mu_r)^2}{2\sigma_r^2}}, & \text{if } |x - \mu_r| \leq w_r \\ 0, & \text{else} \end{cases} \quad (5)$$

where  $\mu_r$  and  $\sigma_r$  are the mean and standard deviation and  $c_r$  is chosen to ensure that the integration of this bounded Gaussian distribution is equal to one. We set  $w_r = 0.2 \mu_r$  and  $\sigma_r = 0.3 \mu_r$ . Because the wordline pitch variation introduces variations on the  $\mu_r$  of  $\gamma_y$  and  $\gamma_{xy}$ , we set the  $\mu_r$  of  $\gamma_x$  as constant while assuming that the  $\mu_r$  of  $\gamma_y$  and  $\gamma_{xy}$  also follows bounded Gaussian distributions as

$$p_\mu(t) = \begin{cases} \frac{c_t}{\sigma_t \sqrt{2\pi}} \cdot e^{-\frac{(t-\mu_t)^2}{2\sigma_t^2}}, & \text{if } |t - \mu_t| \leq w_t \\ 0, & \text{else} \end{cases} \quad (6)$$

where we set  $w_t = 0.2 \mu_t$  and  $\sigma_t = 0.2 \mu_t$  and  $c_t$  is chosen to ensure that the integration of this bounded Gaussian distribution is equal to one. To further demonstrate the effect of cell-to-cell interference, we present the following example.

*Example 2.1:* Let us consider a 2-bit/cell NAND Flash memory. For the erased state, its mean and standard deviation are set to 1.1 and 0.35 V, respectively. For the other three programmed states, we set their verify voltages  $V_p$ 's as 2.55, 3.15, and 3.75 V, respectively, and set the program step voltage

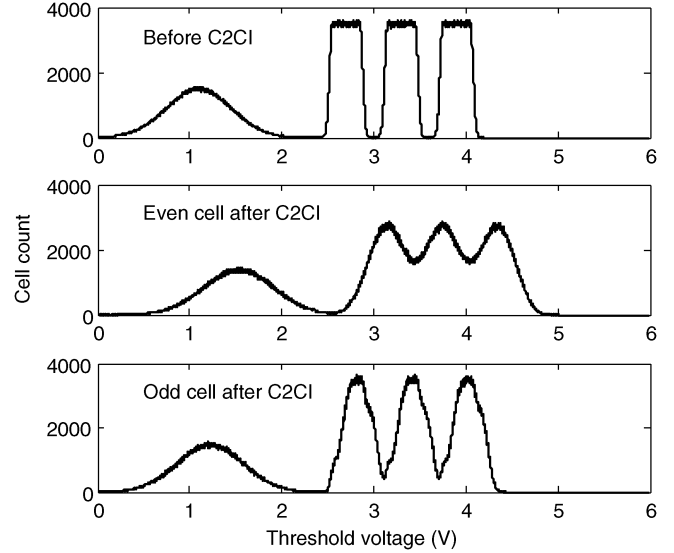


Fig. 5. Simulated cell-threshold-voltage distribution before and after the occurrence of cell-to-cell interference.

$\Delta V_{pp}$  as 0.3 V. Regarding the programming threshold-voltage distribution shown in Fig. 4, we set the standard deviation  $\sigma_p$  of double-side Gaussian tails as 0.03 and have the constant factor  $c_p = 0.2$ . Fig. 5 shows the simulation results, where cell-to-cell interference is abbreviated as C2CI for the purpose of being concise and where the means of  $\gamma_x$ ,  $\gamma_y$ , and  $\gamma_{xy}$  are set to 0.08, 0.064, and 0.0048, respectively. It shows how cell-to-cell interference may distort the cell-threshold-voltage distribution and how the impacts on even and odd cells can differ significantly. The results also suggest that, during the memory read operations, sensing reference voltages for even and odd cells should be configured differently in order to minimize sensing BERs in the presence of significant cell-to-cell interference.

### III. PROPOSED TECHNIQUES TO TOLERATE CELL-TO-CELL INTERFERENCE

As cell-to-cell interference is becoming the major challenge for further NAND Flash memory technology scaling and more aggressive use of MLC storage strategy, it is of paramount importance to develop techniques that can either minimize or tolerate cell-to-cell interference. Many prior works have been focusing on how to minimize cell-to-cell interference through device/circuit techniques such as word-line and/or bit-line shielding [26]–[28]. A straightforward option for tolerating significant cell-to-cell interference is to use stronger ECCs, which nevertheless will result in more coding redundancy, higher controller implementation complexity, and longer operational latency. To facilitate the comparison among different techniques, we use a metric called *cell-storage efficiency*, defined as the average number of real user bits per cell, to represent the NAND Flash memory storage efficiency. For example, if we use an ECC that requires a 28-B coding redundancy to protect each 512-B user data in a 2-bit/cell NAND Flash memory, then the cell-storage efficiency is  $(512/(512+28)) \times 2 = 1.90$  bits/cell.

In the following, we first study the information theoretical bounds of NAND Flash memory cell-storage efficiency in the presence of significant cell-to-cell interference and show that a

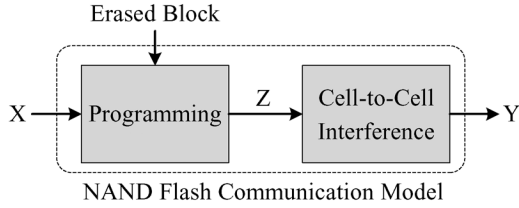


Fig. 6. NAND Flash memory channel model.

straightforward use of strong ECC tends to largely lag from information theoretical bounds. As pointed out earlier, cell-to-cell interference in NAND Flash memory is essentially the same as intersymbol interference encountered in many communication channels. This directly motivates us to investigate the potential of applying the basic concepts of signal equalization and signal predistortion, two well-known signal-processing techniques being widely used to handle communication-channel intersymbol interference, to tolerate cell-to-cell interference and hence close the gap without resorting to strong ECCs. Furthermore, we show that the use of a data-processing technique can also help to derive tighter information theoretical bounds of NAND Flash memory cell-storage efficiency.

#### A. Information Theoretical Bounds of Cell-Storage Efficiency

In this section, we will present mathematical formulations of NAND Flash memory cell-storage-efficiency bounds based upon information theory. We define the cell capacity, denoted as  $C$ , as the information theoretical bound on cell-storage efficiency, i.e., the maximum average number of user bits per cell under which error-free storage can be realized. We model NAND Flash memory as a communication channel, as shown in Fig. 6. It first models the programming of cells in the absence of cell-to-cell interference, as shown in Fig. 4, then models the cell-to-cell interference among adjacent cells. The information theoretical channel capacity of this channel is the NAND Flash memory cell capacity  $C$ . In the remainder of this paper, we denote random variables by uppercase letters ( $X$ ) and their particular realizations by lowercase letters ( $x$ ), and we write the  $n$ -tuples  $(X_1, X_2, \dots, X_n)$  and  $(x_1, x_2, \dots, x_n)$  as  $X^n$  and  $x^n$ , respectively. Due to the cell-to-cell interference, the NAND Flash communication channel is essentially a channel with memory. Assuming that the input bits are statistically independent and have equal probability to be zero and one, we can formulate the channel capacity  $C$  as

$$C = \frac{1}{n} \lim_{n \rightarrow \infty} \{I(X^n; Y^n)\} \quad (7)$$

where  $I(X^n; Y^n) = H(Y^n) - H(Y^n|X^n)$  is the mutual information between the input sequence  $X^n$  and output sequence  $Y^n$ .  $H(Y^n)$  is the entropy of output sequence  $Y^n$ , i.e.,  $H(Y^n) = -\sum_{y^n} p(y^n) \cdot \log_2 p(y^n)$ , and  $H(Y^n|X^n)$  is the conditional entropy, i.e., the entropy of  $Y^n$  given the knowledge of  $X^n$ , i.e.,  $H(Y^n|X^n) = \sum_{x^n} p(x^n) \sum_{y^n} p(y^n|x^n) \cdot \log_2 p(y^n|x^n)$ . For such a communication channel with memory, the exact calculation of conditional entropy can be almost intractable. Hence, instead of striving to derive the exact channel capacity, we intend to

estimate a pair of upper and lower bounds for the channel capacity.

Let us first consider the upper bound. Suppose we remove the cell-to-cell interference component in the channel model, and let  $C_p$  denote the channel capacity of the new channel with the input of  $X$  and output of  $Z$ , as shown in Fig. 6. According to [29], we have the fact that  $C_p \geq C$ ; hence,  $C_p$  can serve as an upper bound. This new channel only consists of the programming component; hence, it is a memoryless channel. Therefore, we can estimate  $C_p$  as  $I(X; Z) = H(Z) - H(Z|X)$ , where  $H(Z) = -\sum_z p(z) \cdot \log_2 p(z)$  and  $H(Z|X) = \sum_x p(x) \sum_z p(z|x) \cdot \log_2 p(z|x)$ . Given all the channel parameters, we can use Monte Carlo simulations to numerically estimate the values of  $p(z)$  and  $p(z|x)$  and, hence, accordingly calculate the upper bound  $C_p$ .

With respect to the lower bound, according to [30], for a channel with memory, we have

$$I(X^n; Y^n) \geq \sum_{i=1}^n I(X_i; Y_i) \quad (8)$$

where  $I(X_i; Y_i)$  is the mutual information between each pair of input and output. Since all the  $X_i$ 's are independent and identically distributed random variables, a lower bound of  $C$  can be estimated as  $I(X; Y) = H(Y) - H(Y|X)$ , where  $H(Y) = -\sum_y p(y) \cdot \log_2 p(y)$  and  $H(Y|X) = \sum_x p(x) \sum_y p(y|x) \cdot \log_2 p(y|x)$ . Again, we can use Monte Carlo simulations to numerically estimate the values of  $p(y)$  and  $p(y|x)$  and, hence, accordingly calculate  $I(X; Y)$  as a lower bound of  $C$ . We use the following example to demonstrate the previous discussions.

*Example 3.1:* Let us again consider a 2-bit/cell NAND Flash memory and keep all the parameters the same as those in Example 2.1 in Section II-C. We carry out extensive simulations to estimate the probabilities  $p(z)$ ,  $p(z|x)$ ,  $p(y)$ , and  $p(y|x)$  over a wide range of cell-to-cell interference strength factor  $s$ , based on which we obtain the upper and lower bounds of the 2-bit/cell NAND Flash memory cell capacity, as shown in Fig. 7. We note that, since the cell-to-cell interference is ignored when estimating the upper bound, the estimated upper bound is independent on the cell-to-cell interference strength factor  $s$  (i.e., it is about 1.9995 bits/cell in this case study, as shown in Fig. 7). We note that, since even and odd cells have different capacity bounds, the bounds shown in Fig. 7 are obtained by averaging these two different bounds. We further calculate the cell-storage efficiency achieved by using BCH codes to ensure a page error rate of less than  $10^{-20}$ , where the page size is 512-B user data. The input to BCH code decoder is obtained by directly sensing and quantizing the cell threshold voltage into 2 bits. As shown in Fig. 5, even and odd cells tend to have different threshold-voltage distributions. Hence, we set the sensing reference voltages differently for even and odd cells in order to minimize the raw BER, which are shown in Fig. 8.

The aforementioned example clearly shows a big gap between cell capacity and achievable storage efficiency when using only ECC to tolerate cell-to-cell interference. In the remainder of this section, we will present two signal-processing techniques, referred to as data postcompensation and data predistortion, which can be used to largely close the gap. These two

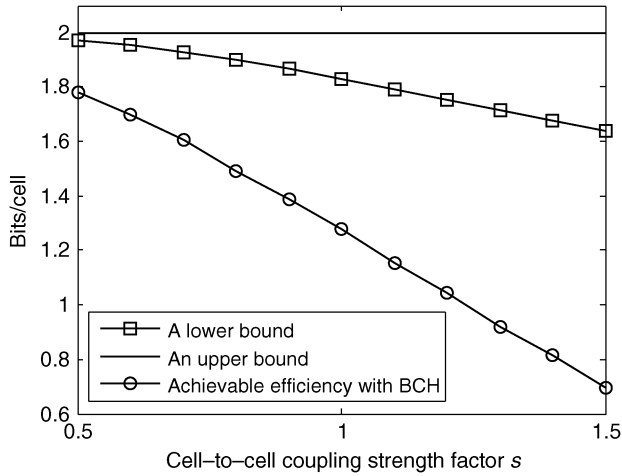


Fig. 7. Calculated upper and lower bounds of cell capacity and the achievable cell efficiency when using BCH codes.

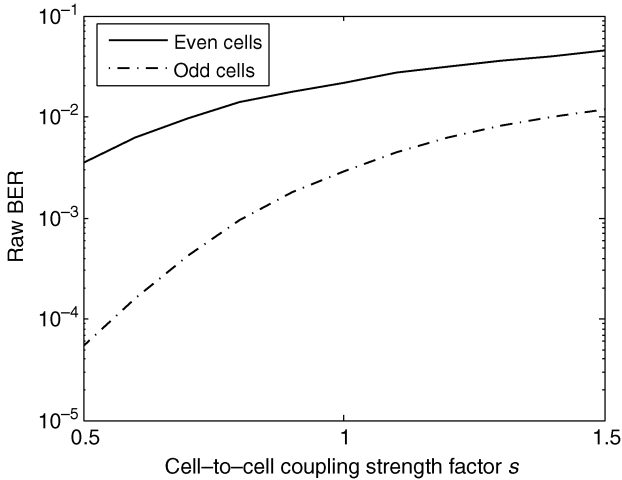


Fig. 8. Simulated raw BER when quantizing the cell threshold voltage into 2 bits.

techniques are directly motivated by signal equalization and predistortion techniques widely used in digital communication, where the key is very intuitive: Because of the data-dependent nature of cell-to-cell interference and its simple deterministic approximate mathematical model, if we know the interfering data when the victim data are being programmed or read, we should be able to explicitly compensate the effect of cell-to-cell interference and hence reduce the BERs. As a result, we can use a weaker ECC with less coding redundancy, leading to an improved cell-storage efficiency.

### B. Technique I: Data Postcompensation

It is clear that if we know the threshold-voltage shift of interfering cells, we can estimate the corresponding cell-to-cell interference strength according to (1) and subsequently subtract it from the sensed threshold voltage of victim cells. By letting  $\tilde{V}_t^{(k)}$  denote the sensed threshold voltage of the  $k$ th interfering cell and  $\bar{V}_e$  denote the mean of the erased state, we can estimate the threshold-voltage shift  $\Delta\tilde{V}_t^{(k)}$  of each interfering cell as  $(\tilde{V}_t^{(k)} - \bar{V}_e)$ . By letting  $\bar{\gamma}^{(k)}$  denote the mean of the corresponding coupling ratio, we can estimate the strength of cell-to-cell interference as

$$\tilde{F} = \sum_k \left( (\tilde{V}_t^{(k)} - \bar{V}_e) \cdot \bar{\gamma}^{(k)} \right). \quad (9)$$

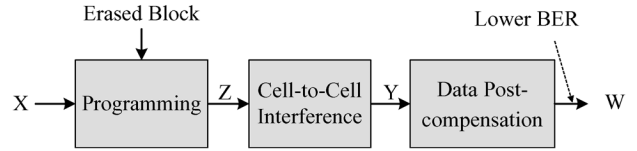


Fig. 9. Memory channel model when data postcompensation is being used.

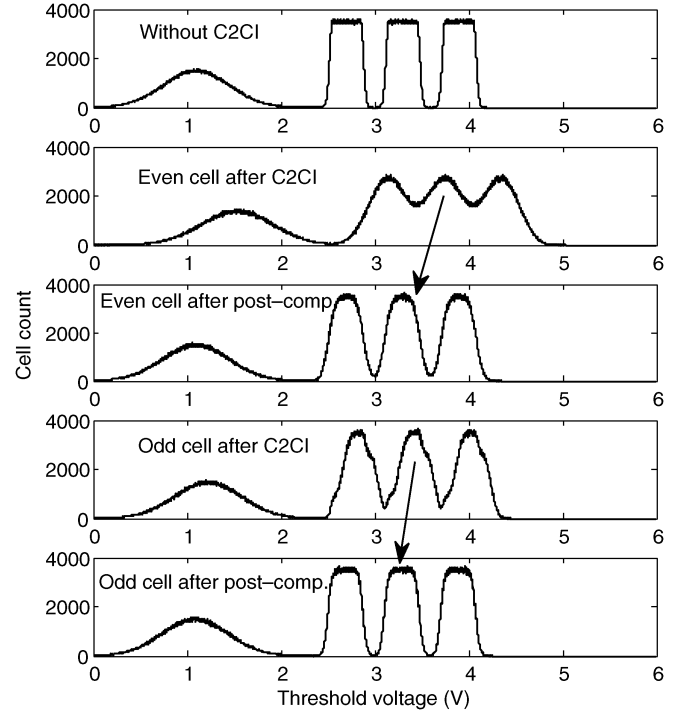


Fig. 10. Simulated victim-cell threshold-voltage distribution before and after data postcompensation.

Therefore, we can postcompensate cell-to-cell interference by subtracting the estimated  $\tilde{F}$  from the sensed threshold voltage of victim cells. This intuitive data postcompensation strategy can be further shown in Fig. 9.

For the purpose of illustration, using the same parameters as in Example 3.1, we carry out Monte Carlo simulations and obtain the victim-cell threshold distributions before and after the postcompensation is used, with the cell-to-cell interference strength factor  $s = 0.8$ , as shown in Fig. 10, where we assume that all the threshold voltages of interfering and victim cells are available in the floating-point precision. It can clearly illustrate the potential of using postcompensation to handle significant cell-to-cell interference. Clearly, practical realizations of this simple data postcompensation strategy involve two issues.

- 1) *Read amplification.* Since the interfering data within other pages should also be read, the actual number of pages to be read will increase, which is referred to as read amplification. This clearly tends to increase read latency, incur energy overhead in both Flash memory chips and controllers, and increase the load of the chip-to-chip link between Flash memory chips and controllers.
- 2) *Fine-grained memory sensing.* In order to provide sufficient accuracy for calculating and compensating cell-to-cell interference, fine-grained cell-threshold-voltage sensing must be used (e.g., for a 2-bit/cell Flash memory, we may need to read the threshold voltage of each cell

with a 4-bit sensing precision). This can increase off-chip communication link load and degrade memory read speed and, possibly, on-chip page-buffer silicon overhead.

With respect to the first issue aforementioned, the run-time read-amplification factor essentially depends on the number of continuous pages being read consecutively by the controller. If only one page is read, to execute postcompensation, we need to read several extra interfering pages only for the purpose of compensating cell-to-cell interference for the selected page, which results in serious read amplification. On the other hand, if a large number of physically consecutive pages are read, a lot of them can be shared in compensating, and hence, postcompensation can be very naturally realized without being subject to significant read amplification. The more physically consecutive pages are read, the lower the average read amplification tends to be. Hence, the data postcompensation strategy should be more preferable for applications dominated by the continuous pages read, such as multimedia data storage. Moreover, as the industry is quickly adopting the use of NAND Flash memory in enterprise systems, it is not uncommon for high-performance computing systems to use a large block size (e.g., 256 kB per block) [31], which naturally leads to accesses of a large amount of pages at one time.

Regarding the second issue aforementioned in terms of fine-grained sensing, the sensing quantization precision directly determines the tradeoff between the cell-to-cell interference compensation effectiveness and induced overhead. For an  $n$ -bit-per-cell NAND Flash memory, an  $m$ -bit fine-grained sensing may increase on-chip page buffer and off-chip link load by  $m/n$  times and increase on-chip sensing latency by roughly  $2^{m-n}$  times. To quantitatively demonstrate the tradeoff, we carry out further simulations under the same configurations as in Example 3.1, where different sensing quantization precisions are considered. Figs. 11 and 12 show the simulated BER versus cell-to-cell coupling strength factor  $s$  for even and odd pages, where 32- and 16-level uniform sensing quantization schemes are considered. Simulation results clearly show the impact of sensing precision on the BER performance. Using 32-level sensing postcompensation could provide large BER performance improvement for both even and odd cells, but 16-level sensing degrades the odd cells' performance when the cell-to-cell interference factor is small.

Moreover, appropriate system design can further reduce the overhead induced by postcompensation. For example, we may always try to first read pages from NAND Flash memory without using postcompensation, and when ECC decoding fails, we execute postcompensation. For the realization of postcompensation, we may gradually increase the granularity of memory sensing, which may reduce the on-chip page-buffer cost and average chip-to-chip link load.

Finally, we note that the use of data postcompensation can also help to derive a new lower bound on cell capacity, which can be much tighter compared with the one obtained in Section III-A. In this context, we treat the combination of Flash memory channel and the data postcompensation module as a new channel, as shown in Fig. 9. According to the data-processing theorem [29], the channel capacity  $C_e$  of this new channel cannot be larger than the Flash memory channel

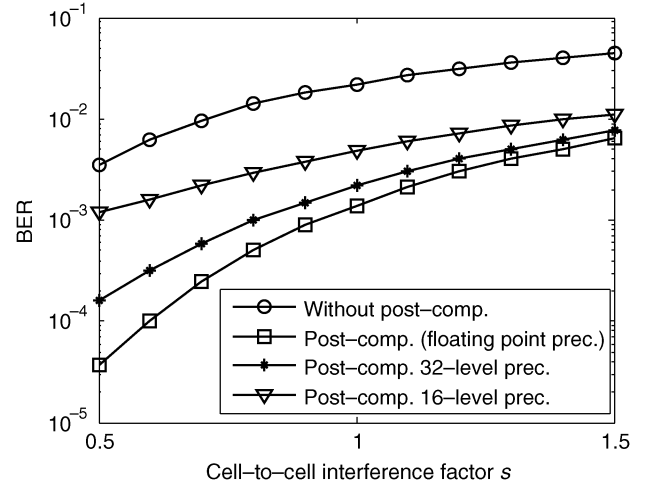


Fig. 11. Simulated BER performance of even cells when data postcompensation is being used.

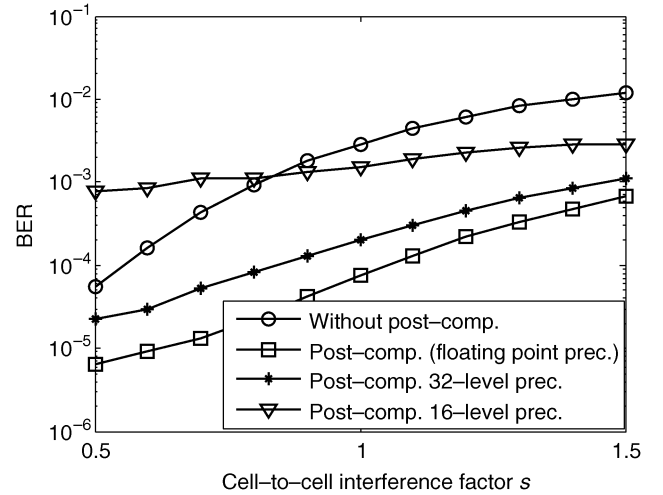


Fig. 12. Simulated BER performance of odd cells when data postcompensation is being used.

capacity  $C$ . Hence, we can use  $C_e$  as a lower bound of  $C$ . By letting  $W$  denote the output of this new channel, as shown in Fig. 9, we have  $C_e = (1/n) \lim_{n \rightarrow \infty} \{I(X^n; W^n)\}$ . Since this new channel is still a communication channel with memory, following the discussion in Section III-A, we can estimate the lower bound of  $C_e$  as  $I(X; W)$ , which can be used as a new lower bound of Flash memory channel capacity  $C$ . Similarly, we can use Monte Carlo simulations to numerically estimate the values of  $p(w)$  and  $p(w|x)$  in order to estimate  $I(X; W)$ . Again, using exactly the same configurations as in Example 3.1, we estimate the new lower bound, as shown in Fig. 13, and found that this new lower bound is very close to the upper bound obtained in Section III-A and much tighter than the previous one. This can be intuitively explained as follows. By largely removing the effect of cell-to-cell interference using data postcompensation, this new channel has much less memory compared with the original channel. As a result, if we simply treat the channel as memoryless and use  $I(X; W)$  to approximate its lower bound, we can expect a much less accuracy loss, which directly leads to a much tighter lower bound.

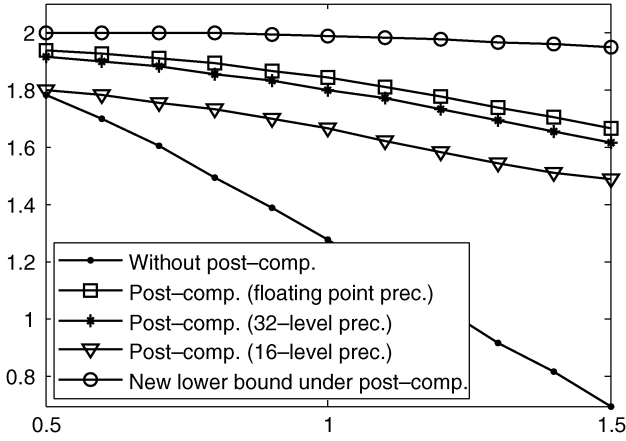


Fig. 13. New lower bound on cell capacity obtained with the help of data post-compensation.

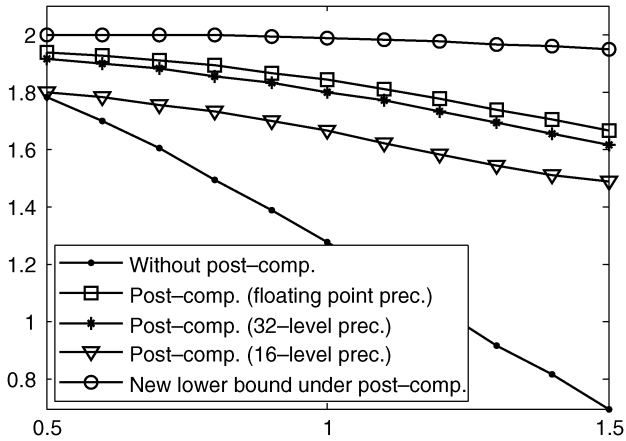


Fig. 14. Achievable cell-storage efficiency when BCH codes are being used and each page stores 512-B user data.

We further calculate the overall cell-storage efficiency achieved by using BCH codes on the output of postcompensation to ensure a page error rate of less than  $10^{-20}$ , where the page size is 512-B user data. The results are shown in Fig. 14, where different sensing precisions are considered and compared against the one without using data postcompensation. Results clearly demonstrate that data postcompensation can largely improve the cell-storage efficiency in the presence of significant cell-to-cell interference. Note that the overall efficiency of post-compensation under 16-level sensing quantization is still larger than that without postcompensation, due to the BER performance improvement of even cells.

### C. Technique II: Data Predistortion

As discussed earlier, the data postcompensation technique is inherently subject to the read-amplification issue, which can be particularly problematic for applications that tend to read one or a few random pages at a time. To avoid this read-amplification issue, this section presents another technique, called data predistortion. Following the same concept underlying data predistortion techniques widely used in data communication, the key idea here is simple: Before a page is programmed, if its interfering pages are also known, we can predict the threshold-voltage shift induced by cell-to-cell interference for each victim

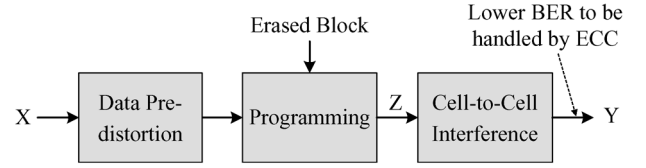


Fig. 15. Memory channel model when data predistortion is being used.

cell and then correspondingly predistort the victim-cell target programming voltage. Hence, after its interfering pages are programmed, the predistorted victim-cell threshold voltages will be shifted to its desired location by cell-to-cell interference. The corresponding channel model is shown in Fig. 15. The prediction on the threshold-voltage shift by cell-to-cell interference is carried out by the NAND Flash memory controller, which feeds Flash memory chips with the desired target programming voltage.

By letting  $V_t^{(k)}$  denote the expected threshold voltage of the  $k$ th interfering cell after programming and  $\bar{V}_e$  denote the mean of the erased state, we can predict the cell-to-cell interference experienced by the victim cell as

$$\hat{F} = \sum_k \left( (V_t^{(k)} - \bar{V}_e) \cdot \bar{\gamma}^{(k)} \right). \quad (10)$$

By letting  $V_p$  denote the target verify voltage of the victim cell in programming operation, we can predistort the victim cell by shifting the verify voltage from  $V_p$  to  $V_p - \hat{F}$ . Through such data predistortion processing, we can expect that the threshold voltage of the victim cell will be shifted toward its desired location after the occurrence of cell-to-cell interference. It should be emphasized that, since we cannot change the threshold voltage if the victim cell should stay at the erased state, this predistortion scheme can only handle cell-to-cell interference for those programmed states but is not effective for the erased state. Fig. 16 further shows the underlying idea of this data predistortion technique, where we assume that the verify voltage  $V_p$  can be adjusted with a floating-point precision. Clearly, this technique can be considered as a counterpart of the data postcompensation technique presented in Section III-B.

Clearly, this data predistortion scheme is preferred when a large number of continuous pages are being programmed at a time. Because NAND Flash memory controllers contain on-chip cache and carry out sophisticated management functions such as wear leveling and garbage collection at the Flash transaction layer to improve NAND Flash endurance and throughput performance [32], it is not uncommon that NAND Flash memory programming patterns are dominated by continuous physical page programming in practice.

Using the same configurations as in Example 3.1, we carry out simulations to quantitatively evaluate the effectiveness of data predistortion. Fig. 17 shows the cell threshold distribution with the cell-to-cell interference strength factor  $s = 0.8$ , where we assume that the predistortion can be done by adjusting the verify voltage  $V_p$  with a floating-point precision. We note that, when using predistortion, the distribution of programmed states becomes close to that without cell-to-cell interference, while the erased state still suffers considerable cell-to-cell interference, as shown in Fig. 17. As pointed out earlier, data predistortion can



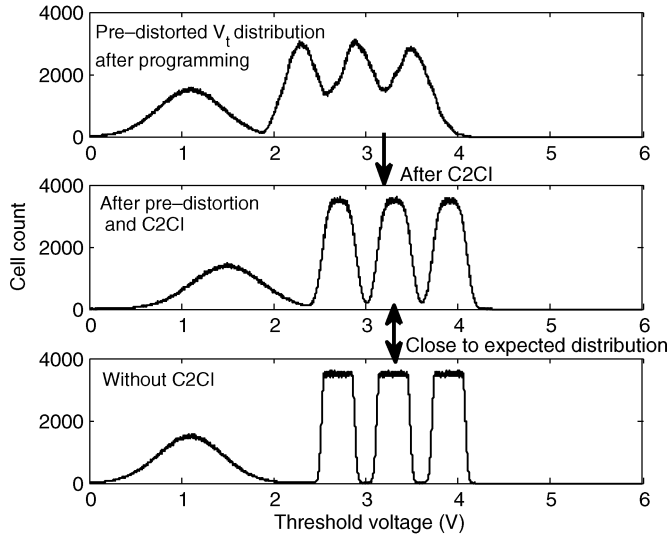


Fig. 16. Illustration of threshold-voltage distribution of victim even cells when data predistortion is being used.

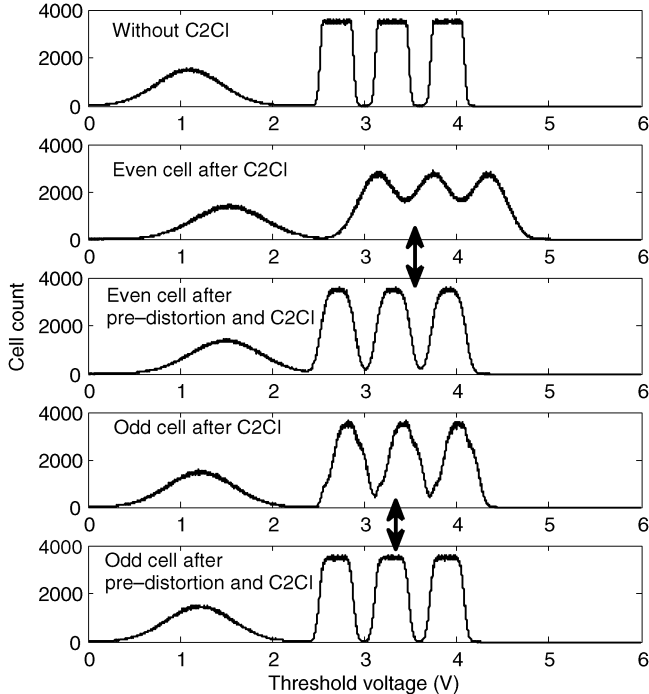


Fig. 17. Simulated  $V_t$  distribution when using the predistortion technique.

only be used for programmed states. Hence, cell-to-cell interference will shift and widen the threshold-voltage distribution of the erased state, which cannot be compensated by data predistortion and will result in a higher probability of errors.

Figs. 18 and 19 show the simulated BER over a range of cell-to-cell interference strength factor  $s$ . Aside from the ideal floating point precision, we also consider predistortion with finite precision, where the range of predistorted  $V_p$  is quantized into either 16 or 32 levels. Clearly, as we increase the finite quantization precision of predistorted  $V_p$ , it can achieve a better tolerance to cell-to-cell interference, at the cost of increased programming latency, a larger page buffer to hold the data, and higher chip-to-chip communication load.

Similar to the discussion in Section III-B, we can combine the data predistortion module and the Flash memory as a new

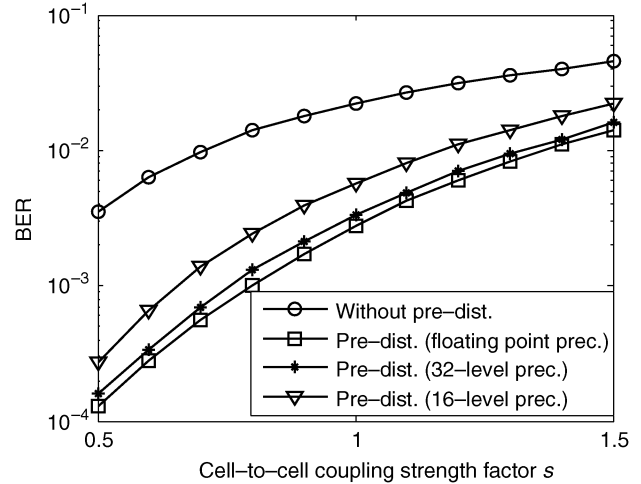


Fig. 18. Simulated BER performance of even cells when data predistortion is being used.

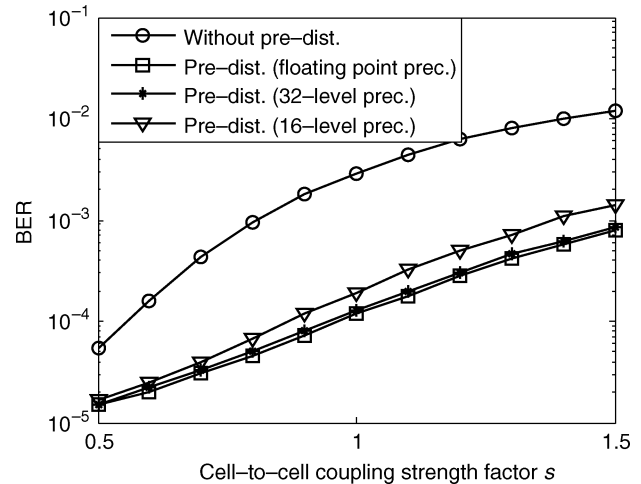


Fig. 19. Simulated BER performance of odd cells when data predistortion is being used.

channel with input  $X$  and output  $Y$ , as shown in Fig. 15, which can be used to derive another new lower bound of the Flash memory channel. Fig. 20 shows the calculated lower bound when using data predistortion, which is further compared with the lower bound obtained when using data postcompensation. It shows that the use of data postcompensation can help to obtain a tighter Flash memory cell capacity lower bound. This is mainly because of the following: 1) Data predistortion cannot compensate the effect of cell-to-cell interference on the erased state, and 2) data postcompensation can estimate the cell-to-cell interference more precisely with the sensed threshold voltage than data predistortion that only uses the means of erased or programmed states to predict the actual threshold voltage.

Fig. 21 further shows the achievable NAND Flash memory cell-storage efficiency when using data predistortion under different finite precision configurations. Similarly, when estimating the achievable cell-storage efficiency, BCH codes are used to ensure a page error rate of less than  $10^{-20}$ , where the page size is 512-B user data. The results show that predistortion can also largely improve the cell-storage efficiency.

Practical realization of the predistortion incurs penalty in terms of programming latency and on-chip page-buffer size.

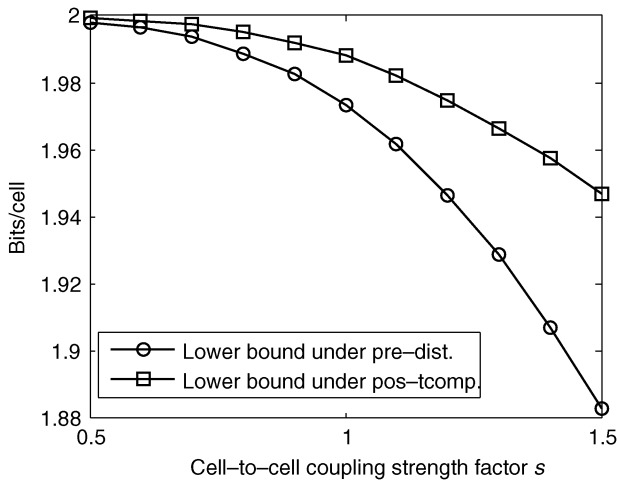


Fig. 20. Different Flash memory cell capacity lower bounds under postcompensation and predistortion.

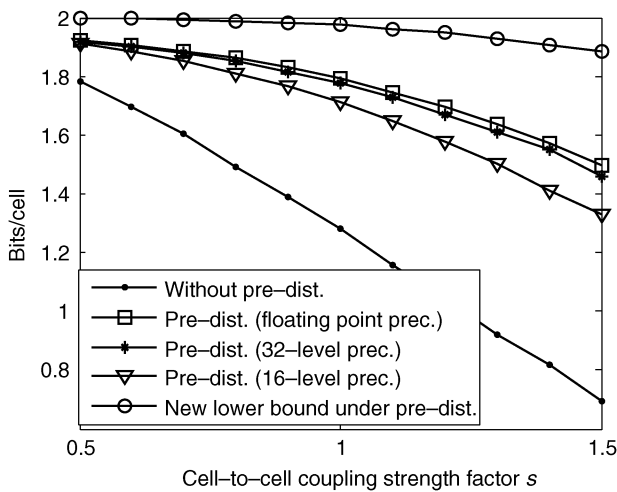


Fig. 21. Achievable cell-storage efficiency with predistortion when BCH codes are being used and each page stores 512-B user data.

Since the programming step voltage (i.e.,  $\Delta V_{pp}$ ) remains the same, the extra programming latency mainly comes from fine-grained verification during the iterative program-and-verify procedure. The on-chip page-buffer size has to accordingly increase because of the fine-grained predistorted programming voltages. Again, the overhead depends on the specific predistortion finite granularity, and there is an essential tradeoff between the implementation overhead and cell-to-cell interference tolerance effectiveness.

#### IV. CONCLUSION

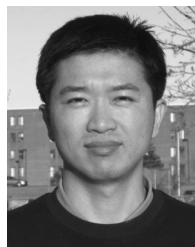
This paper has presented two simple yet effective data-processing techniques that can well tolerate significant cell-to-cell interference in MLC NAND Flash memory and hence reduce raw-memory bit errors. As a result, they can enable the use of weaker ECC with less coding redundancy, leading to higher memory cell-storage efficiency. This paper has been essentially motivated by the similarity between intersymbol interference in data communication channels and cell-to-cell interference in NAND Flash memory, and the two presented data-processing

techniques directly originate from signal equalization and predistortion strategies being widely used to handle intersymbol interference in communication channels. We have carried out extensive computer simulations and information theoretical analysis to demonstrate the effectiveness of these two techniques. We have also discussed the involved design tradeoffs and overheads for their practical implementations. In spite of the incurred implementation overhead, given the significant potential performance advantages, as shown in this paper, we believe that these data-processing techniques provide viable system-level solutions to handle significant cell-to-cell interference and hence enable more aggressive technology scaling for future MLC NAND Flash memory.

#### REFERENCES

- [1] K. Kanda, M. Koyanagi, T. Yamamura, K. Hosono, M. Yoshihara, T. Miwa, Y. Kato, A. Mak, S.L. Chan, F. Tsai, R. Cernea, Le B., E. Makino, T. Taira, H. Otake, N. Kajimura, S. Fujimura, Y. Takeuchi, M. Itoh, M. Shirakawa, D. Nakamura, Y. Suzuki, Y. Okukawa, M. Kojima, K. Yoneya, T. Arizono, T. Hisada, S. Miyamoto, M. Noguchi, T. Yaegashi, M. Higashitani, F. Ito, T. Kamei, G. Hemink, T. Maruyama, K. Ino, and S. Ohshima, "A 120 mm<sup>2</sup> 16 Gb 4-MLC NAND flash memory with 43 nm CMOS technology," in *Proc. IEEE ISSCC*, 2008, pp. 430–431, 625.
- [2] T. Futatsuyama, N. Fujita, N. Tokiwa, Y. Shindo, T. Edahiro, T. Kamei, H. Nasu, M. Iwai, K. Kato, Y. Fukuda, N. Kanagawa, N. Abiko, M. Matsumoto, T. Himeno, T. Hashimoto, Y.-C. Liu, H. Chibvongdze, T. Hori, M. Sakai, H. Ding, Y. Takeuchi, H. Shiga, N. Kajimura, Y. Kajitani, K. Sakurai, K. Yanagidaira, T. Suzuki, Y. Namiki, T. Fujimura, M. Mui, H. Nguyen, S. Lee, A. Mak, J. Lutze, T. Maruyama, T. Watanabe, T. Hara, and S. Ohshima, "A 113 mm<sup>2</sup> 32 Gb 3 b/cell NAND flash memory," in *Proc. IEEE Int. Solid-State Circuits Conf.*, Feb. 2009, pp. 242–243.
- [3] Y. Li, S. Lee, Y. Fong, F. Pan, T.-C. Kuo, J. Park, T. Samaddar, H.T. Nguyen, M.L. Mui, K. Htoo, T. Kamei, M. Higashitani, E. Yero, G. Kwon, P. Kliza, J. Wan, T. Kaneko, H. Maejima, H. Shiga, M. Hamada, N. Fujita, K. Kanebako, E. Tam, A. Koh, I. Lu, C.C.-H. Kuo, T. Pham, J. Huynh, Q. Nguyen, H. Chibvongdze, M. Watanabe, K. Oowada, G. Shah, B. Woo, R. Gao, J. Chan, J. Lan, P. Hong, L. Peng, D. Das, D. Ghosh, V. Kalluru, S. Kulkarni, R.-A. Cernea, S. Huynh, D. Pantelakis, C.-M. Wang, and K. Quader, "A 16 Gb 3-bit per cell (X3) NAND flash memory on 56 nm technology with 8 MB/s write rate," *IEEE J. Solid-State Circuits*, vol. 44, no. 1, pp. 195–207, Jan. 2009.
- [4] S.-H. Chang, S.-K. Lee, S.-J. Park, M.-J. Jung, J.-C. Han, I.-S. Wang, K.-H. Lim, J.-H. Lee, J.-H. Kim, W.-K. Kang, T.-K. Kang, H.-S. Byun, Y.-J. Noh, L.-H. Kwon, B.-K. Koo, M. Cho, J.-S. Yang, and Y.-H. Koh, "A 48 nm 32 Gb 8-level NAND flash memory with 5.5 MB/s program throughput," in *Proc. IEEE Int. Solid-State Circuits Conf.*, Feb. 2009, pp. 240–241.
- [5] N. Shibata, H. Maejima, K. Isobe, K. Iwasa, M. Nakagawa, M. Fujiu, T. Shimizu, M. Honma, S. Hoshi, T. Kawaai, K. Kanebako, S. Yoshikawa, H. Tabata, A. Inoue, T. Takahashi, T. Shano, Y. Komatsu, K. Nagaba, M. Kosakai, N. Motohashi, K. Kanazawa, K. Imamiya, H. Nakai, M. Lasser, M. Murin, A. Meir, A. Eyal, and M. Shlick, "A 70 nm 16 Gb 16-level-cell NAND flash memory," *IEEE J. Solid-State Circuits*, vol. 43, no. 4, pp. 929–937, Apr. 2008.
- [6] C. Trinh, N. Shibata, T. Nakano, M. Ogawa, J. Sato, Y. Takeyama, K. Isobe, B. Le, F. Moogat, N. Mokhlesi, K. Kozakai, P. Hong, T. Kamei, K. Iwasa, J. Nakai, T. Shimizu, M. Honma, S. Sakai, T. Kawaai, S. Hoshi, J. Yuh, C. Hsu, T. Tseng, J. Li, J. Hu, M. Liu, S. Khalid, J. Chen, M. Watanabe, H. Lin, J. Yang, K. McKay, K. Nguyen, T. Pham, Y. Matsuda, K. Nakamura, K. Kanebako, S. Yoshikawa, W. Igarashi, A. Inoue, T. Takahashi, Y. Komatsu, C. Suzuki, K. Kanazawa, M. Higashitani, S. Lee, T. Murai, J. Lan, S. Huynh, M. Murin, M. Shlick, M. Lasser, R. Cernea, M. Mofidi, K. Schuegraf, and K. Quader, "A 5.6 MB/s 64 Gb 4 b/Cell NAND flash memory in 43 nm CMOS," in *Proc. IEEE Int. Solid-State Circuits Conf.*, Feb. 2009, pp. 246–247.
- [7] N. Mielke, T. Marquart, N. Wu, J. Kessenich, H. Belgal, E. Schares, F. Trivedi, E. Goodness, and L. R. Nevill, "Bit error rate in NAND flash memories," in *Proc. IEEE Int. Rel. Phys. Symp.*, 2008, pp. 9–19.

- [8] K. Kim, "Future memory technology: Challenges and opportunities," in *Proc. Int. Symp. VLSI Technol., Syst. Appl.*, Apr. 2008, pp. 5–9.
- [9] K. Prall, "Scaling non-volatile memory below 30 nm," in *Proc. IEEE 2nd Non-Volatile Semiconductor Memory Workshop*, Aug. 2007, pp. 5–10.
- [10] H. Liu, S. Grootuis, C. Mouli, J. Li, K. Parat, and T. Krishnamohan, "3D simulation study of cell-cell interference in advanced NAND flash memory," in *Proc. IEEE Workshop Microelectron. Electron. Devices*, Apr. 2009, pp. 1–3.
- [11] E. A. Lee and D. G. Messerschmidt, *Digital Communication*. Norwell, MA: Kluwer, 1994.
- [12] J. Chen, Y. Gu, and K. K. Parhi, "Novel FEXT cancellation and equalization for high speed ethernet transmission," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 56, no. 6, pp. 1272–1285, Jun. 2009.
- [13] T. M. Hollis, D. J. Comer, and D. T. Comer, "Mitigating ISI through self-calibrating continuous-time equalization," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 53, no. 10, pp. 2234–2245, Oct. 2006.
- [14] S. Hoyos, J. A. Garcia, and G. R. Arce, "Mixed-signal equalization architectures for printed circuit board channels," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 51, no. 2, pp. 264–274, Feb. 2004.
- [15] Y. H. Chung and S. M. Phoong, "Unitary precoders for ST-OFDM systems using alamouti STBC," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, no. 9, pp. 2860–2869, Oct. 2008.
- [16] Y. Li and Y. Fong, "Compensating for coupling based on sensing a neighbor using coupling," USA Patent 7 522 454, Apr. 2009.
- [17] K.-D. Suh, B.-H. Suh, Y.-H. Lim, J.-K. Kim, Y.-J. Choi, Y.-N. Koh, S.-S. Lee, S.-C. Kwon, B.-S. Choi, J.-S. Yum, J.-H. Choi, J.-R. Kim, and H.-K. Lim, "A 3.3 V 32 Mb NAND flash memory with incremental step pulse programming scheme," *IEEE J. Solid-State Circuits*, vol. 30, no. 11, pp. 1149–1156, Nov. 1995.
- [18] R. Bez, E. Camerlenghi, A. Modelli, and A. Visconti, "Introduction to flash memory," *Proc. IEEE*, vol. 91, no. 4, pp. 489–502, Apr. 2003.
- [19] K. Takeuchi, Y. Kameda, S. Fujimura, H. Otake, K. Hosono, H. Shiga, Y. Watanabe, T. Futatsuyama, Y. Shindo, M. Kojima, M. Iwai, M. Shirakawa, M. Ichige, K. Hatakeyama, S. Tanaka, T. Kamei, J.-Y. Fu, A. Cernea, Y. Li, M. Higashitani, G. Hemink, S. Sato, K. Oowada, S.-C. Lee, N. Hayashida, J. Wan, J. Lutze, S. Tsao, M. Mofidi, K. Sakurai, N. Tokiwa, H. Waki, Y. Nozawa, K. Kanazawa, and S. Ohshima, "A 56-nm CMOS 99- mm<sup>2</sup> 8-Gb multi-level NAND flash memory with 10-mb/s program throughput," *IEEE J. Solid-State Circuits*, vol. 42, no. 1, pp. 219–232, Jan. 2007.
- [20] K.-T. Park, M. Kang, D. Kim, S.-W. Hwang, B.Y. Choi, Y.-T. Lee, C. Kim, and K. Kim, "A zeroing cell-to-cell interference page architecture with temporary LSB storing and parallel MSB program scheme for MLC NAND flash memories," *IEEE J. Solid-State Circuits*, vol. 40, no. 4, pp. 919–928, Apr. 2008.
- [21] Y. Li, S. Lee, Y. Fong, F. Pan, T.-C. Kuo, J. Park, T. Samaddar, H. Nguyen, M. Mui, K. Htoo, T. Kamei, M. Higashitani, E. Yero, G. Kwon, P. Kliza, J. Wan, T. Kaneko, H. Maejima, H. Shiga, M. Hamada, N. Fujita, K. Kanebako, E. Tarn, A. Koh, I. Lu, C. Kuo, T. Pham, J. Huynh, Q. Nguyen, H. Chibvongodze, M. Watanabe, K. Oowada, G. Shah, B. Woo, R. Gao, J. Chan, J. Lan, P. Hong, L. Peng, D. Das, D. Ghosh, V. Kalluru, S. Kulkarni, R. Cernea, S. Huynh, D. Pantelakis, C.-M. Wang, and K. Quader, "A 16 Gb 3 b/Cell NAND flash memory in 56 nm with 8 MB/s write rate," in *Proc. IEEE ISSCC*, Feb. 2008, pp. 506–632.
- [22] R.-A. Cernea, L. Pham, F. Moogat, S. Chan, B. Le, Y. Li, S. Tsao, T.-Y. Tseng, K. Nguyen, J. Li, J. Hu, J.H. Yuh, C. Hsu, F. Zhang, T. Kamei, H. Nasu, P. Kliza, K. Htoo, J. Lutze, Y. Dong, M. Higashitani, J. Yang, H.-S. Lin, V. Sakhamuri, A. Li, F. Pan, S. Yadala, S. Taigor, K. Pradhan, J. Lan, J. Chan, T. Abe, Y. Fukuda, H. Mukai, K. Kawakami, C. Liang, T. Ip, S.-F. Chang, J. Lakshminpathi, S. Huynh, D. Pantelakis, M. Mofidi, and K. Quader, "A 34 MB/s MLC write throughput 16 Gb NAND with all bit line architecture on 56 nm technology," *IEEE J. Solid-State Circuits*, vol. 44, no. 1, pp. 186–194, Jan. 2009.
- [23] J.-D. Lee, S.-H. Hur, and J.-D. Choi, "Effects of floating-gate interference on NAND flash memory cell operation," *IEEE Electron. Device Lett.*, vol. 23, no. 5, pp. 264–266, May 2002.
- [24] 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.
- [25] N. Shibata, H. Maejima, K. Isobe, K. Iwasa, M. Nakagawa, M. Fujii, T. Shimizu, M. Honma, S. Hoshi, T. Kawaai, K. Kanebako, S. Yoshikawa, H. Tabata, A. Inoue, T. Takahashi, T. Shano, Y. Komatsu, K. Nagaba, M. Kosakai, N. Motohashi, K. Kanazawa, K. Imamiya, and H. Nakai, "A 70 nm 16 Gb 16-level-cell NAND flash memory," in *Proc. IEEE Symp. VLSI Circuits*, 2007, pp. 190–191.
- [26] G. Matamis, T. Pham, H. Chien, and H. Fang, "Bitline direction shielding to avoid cross coupling between adjacent cells for NAND flash memory," USA Patent 7 221 008, May 22, 2007.
- [27] J. W. Lutze and N. Mokhlesi, "Shield plate for limiting cross coupling between floating gates," USA Patent 7 355 237, Apr. 2008.
- [28] H. Chien and Y. Fong, "Deep Wordline Trench to Shield Cross Coupling Between Adjacent Cells for Scaled NAND," USA Patent 7 170 786, Jan. 30, 2007.
- [29] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Hoboken, NJ: Wiley, 1991.
- [30] R. J. McEliece, *The Theory of Information and Coding*. Cambridge, U.K.: Cambridge Univ. Press, 2002.
- [31] J. M. May, *Parallel I/O for High Performance Computing*. San Mateo, CA: Morgan Kaufmann, 2000.
- [32] E. Gal and S. Toledo, "Algorithms and data structures for flash memories," *ACM Comput. Surv.*, vol. 37, no. 2, pp. 138–163, Jun. 2005.



**Guiqiang Dong** (S'09) received the B.S. and M.S. degrees from the University of Science and Technology of China, Hefei, China, in 2004 and 2008, respectively. He is currently working toward the Ph.D. degree in the Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY.

His research interests include coding theory, signal processing for data-storage systems, and fault-tolerant system design for various digital memories.



**Shu Li** received the B.S. and M.S. degrees in electrical engineering from Tsinghua University, Beijing, China, in 2003 and 2005, respectively, and the Ph.D. degree in electrical engineering from Rensselaer Polytechnic Institute, Troy, NY, in 2009.

He is currently a Senior Engineer with Marvell Semiconductor, Inc., Sunnyvale, CA. His research interests include VLSI architecture and circuit design for communication and storage systems.



**Tong Zhang** (M'02–SM'08) received the B.S. and M.S. degrees in electrical engineering from Xi'an Jiaotong University, Xi'an, China, in 1995 and 1998, respectively, and the Ph.D. degree in electrical engineering from the University of Minnesota, Minneapolis, in 2002.

He is currently an Associate Professor with the Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY. His research activities span over circuits and systems for various data-storage and computing applications.

He currently serves as an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II and the IEEE TRANSACTIONS ON SIGNAL PROCESSING.