A Time-Aware Fault Tolerance Scheme to Improve Reliability of Multilevel Phase-Change Memory in the Presence of Significant Resistance Drift

Wei Xu and Tong Zhang, *Senior Member, IEEE*

***Abstract*—Because of its promising scalability potential and support of multilevel per cell storage, phase-change memory has become a topic of great current interest. However, recent studies show that structural relaxation effect makes the resistance of phase-change material drift over the time, which can severely degrade multilevel per cell phase-change memory storage reliability. This makes powerful memory fault tolerance solutions indispensable, where error correction code (ECC) will play an essential role. This work aims to develop fault tolerance solutions that can effectively compensate memory cell resistance drift. First, based upon information-theoretical study, we show that conventional use of ECC, which is unaware of memory content lifetime, can only achieve the performance with a big gap from the information-theoretical bounds. This motivates us to study the potential of time-aware memory fault tolerance, where the basic idea is to keep track the memory content lifetime and use this lifetime information to accordingly adjust how memory cell resistance is quantized and interpreted for ECC decoding. Under this time-aware fault tolerance framework, we study the use of two types of ECCs, including classical codes such as BCH that only demand hard-decision input and advanced codes such as low-density parity-check (LDPC) codes that demand soft-decision probability input. Using hypothetical four-level per cell and eight-level per cell phase-change memory with BCH and LDPC codes as test vehicles, we carry out extensive analysis and simulations, which demonstrate very significant performance advantages of such time-aware memory fault tolerance strategy in the presence of significant memory cell resistance drift.**

***Index Terms*—BCH, error correction code (ECC), low-density parity-check (LDPC), phase-change memory, resistance drift, structural relaxation.**

I. Introduction

THE past few years have experienced a resurgence of interest in search for highly scalable universal memory [1]–[3] fueled by the well-recognized scalability limits of current mainstream memory technologies such as flash memory and DRAM. Phase-change memory is one of the most promising candidates that have attracted a lot of attentions (e.g., see [4]–[11]). Beyond simply providing potential successors for flash memory and DRAM, phase-change memory has been identified as the most viable candidate to enable a truly storage-class memory [12], [13], which combines the benefits of a solid-state memory, such as high performance and robustness, with the non-volatility and low cost of conventional magnetic recording storage. Storage-class memory could fundamentally change the memory and storage hierarchy for virtually all computing platforms and hence generate unprecedented impacts on an extremely wide spectrum of real-life applications.

In phase-change memory, data storage is realized by configuring the chalcogenide material, usually $Ge_2Sb_2Te_5$ (GST), into either a crystalline or an amorphous phase with resistances differing by several orders of magnitude [14]. Due to the large resistance margin between crystalline and amorphous phases, phase-change memory can realize multilevel per cell storage based upon incomplete phase transition [7], [8], which is conceptually similar to multilevel per cell NAND flash memory in present mainstream market. Given its obvious storage density and hence cost advantages, successful development of multilevel per cell phase-change memory is highly desirable and can be critical for phase-change memory technology to enter mainstream market. However, as demonstrated in recent studies [15]–[18], resistance of phase-change material tends to drift over the time with significant variability in terms of drift speed. This could severely degrade the noise margin of multilevel per cell phase-change memory, leading to a serious reliability challenge.

Effective fault tolerance schemes are indispensable to ensure storage reliability of multilevel per cell phase-change memory in the presence of significant resistance drift. Apparently, error correction code (ECC) should be an integral component in multilevel per cell phase-change memory fault tolerance systems. Because ECC incurs coding redundancy that nevertheless degrades storage capacity, to compare different memory fault tolerance strategies, we use a metric called *cell storage efficiency* that is defined as the average number of real user bits per cell. For example, if we use a BCH code that requires 28-byte coding redundancy to protect each 512-byte user data in a 4-level/cell (i.e., 2-bit/cell) phase-change memory, then the cell storage efficiency is $512/512 + 28 \times 2 = 1.90$ bits/cell. In general, fault tolerance schemes that can achieve higher cell storage efficiencies are more desirable. Before trying to develop effective fault tolerance solutions, it is desirable to obtain the theoretical bound of phase-change memory cell storage efficiency, regardless to specific fault tolerance techniques being used. Therefore, by treating phase-change memory as

Manuscript received January 19, 2010; revised April 10, 2010; accepted June 02, 2010. Date of publication June 28, 2010; date of current version July 27, 2011.

W. Xu is with the Marvell Technology, Santa Clara, CA 95054 USA (e-mail: weixu@marvell.com).

T. Zhang is with the Department of Electrical, Computer and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY 12180 USA (e-mail: tzhang@ecse.rpi.edu).

Digital Object Identifier 10.1109/TVLSI.2010.2052640

communication channel, we first investigate the use of information theory [19] to derive phase-change memory cell storage efficiency theoretical bounds, and demonstrate it using hypothetical four-level per cell and eight-level per cell phase-change memory with memory cell resistance drift statistical parameters from recent device studies [17], [18]. Meanwhile, assuming the phase-change memory cell sensing is carried out unaware of the memory content lifetime, which is defined as the period during which the present data have been stored in the memory, we obtain the cell storage efficiency achieved by using BCH codes and low-density parity-check (LDPC) codes, which nevertheless shows a very big gap from the information-theoretical bounds in the presence of significant memory cell resistance drift.

This big gap between theoretical bounds and conventional design practice motivates us to investigate the potential of incorporating data lifetime awareness to improve memory fault tolerance performance in the presence of significant memory cell resistance drift. Intuitively, because of the time-dependent nature of phase-change memory cell resistance drift, memory fault tolerance aware of data lifetime could better compensate the resistance drift issue and hence realize a higher cell storage efficiency. Hence, the second focus of this work is to study such time-aware fault tolerance, for which the basic idea is intuitive: if we have the resistance drift statistical model and know the lifetime of each word (i.e., how long each word has been held in the memory), we should be able to estimate the resistance drift characteristics and accordingly adjust how memory cell resistance is quantized and interpreted by ECC decoders, which may largely improve the tolerance to memory cell resistance drift. Assuming the availability of phase-change memory cell resistance drift statistical model, this paper studies the most straightforward realization of such time-aware memory fault tolerance, i.e., when each word is being written to multilevel phase change memory, we append the current time as a time stamp to the word; when each word is being read, we can obtain the word lifetime by comparing the current time and word time stamp, based on which we can estimate the cell resistance statistical distribution parameters (e.g., the mean and deviation of each storage level) and accordingly adjust how the memory cell resistance is quantized and interpreted by ECC decoders. Because cost-effective multilevel per cell phase-change memory is most suitable for block-oriented data storage with relatively large memory word size (e.g., 512-byte or 4-kB user data as in current NAND flash memory and hard disk drives), the area overhead induced by the storage of time stamp can be negligible.

Assuming Gaussian distributions of phase-change memory cell resistance drift parameters as suggested by recent experimental measurements [9], [20], we mathematically derive how the data lifetime information can be used to adjust memory cell resistance quantization and interpretation for ECC decoders that either only demand hard-decision input such as BCH and RS codes or demand soft-decision probability input such as LDPC codes. We demonstrate the effectiveness of this proposed time-aware fault tolerance by applying BCH and LDPC codes on the same hypothetical four-level per cell and eight-level per cell phase-change memory. Analysis and simulation results show that such time-aware memory fault tolerance can significantly improve the overall memory storage reliability over its time-independent counterpart. This can directly translate to a much longer allowable memory content lifetime, e.g., our simulation results show that the allowable memory content lifetime in four-level per cell phase-change memory may improve by four and seven orders of magnitude when BCH and LDPC codes are being used, respectively.

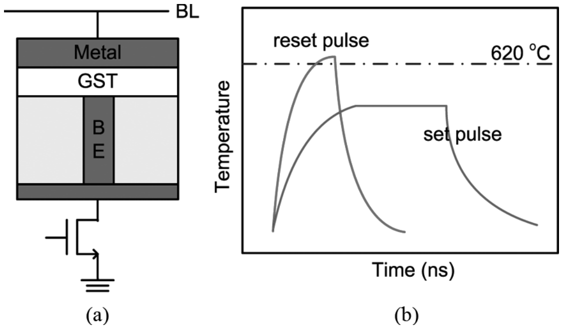


Fig. 1. (a) Phase change memory cell structure and (b) GST programming temperature pulse.

The remainder of this paper is organized as follows. The basics of phase-change memory is reviewed in Section II. Section III discusses the derivation of information-theoretical memory cell storage efficiency bounds. Section IV elaborates on the proposed time-aware memory fault tolerance and demonstrates its effectiveness. Conclusions are drawn in Section V.

II. Background

As shown in Fig. 1(a), one phase-change memory cell consists of an access control device (e.g., an nMOS transistor) and a storage element, i.e., a chalcogenide layer [$Ge_2Sb_2Te_5$ (GST)] between a top metal contact and a resistive bottom electrode (BE). The top of GST is connected to the bit-line (BL) through the top metal and the bottom of GST is connected to the drain of the nMOS transistor. Being heated by current pulses with different amplitudes and durations, the GST can switch between the amorphous (RESET) state and crystalline (SET) state. When the chalcogenide material is amorphous (or crystalline), the GST shows high (or low) resistance. The resistance may differ by several orders of magnitude, leading to excellent nonvolatile storage capability.

Fig. 1(b) illustrates the temperature pulses for the programming of phase-change elements. The amorphous state is obtained by a short current pulse with high peak value, referred to as *reset pulse*, which first raises the temperature of GST above the melting temperature (about 620 °C) and then quickly cools it down [14]. As the result, the molten GST material does not have enough time to realize the crystalline bonding and hence forms an amorphous state. To obtain the crystalline state, the GST is heated by a longer current pulse with lower peak value, referred to as *set pulse*. It heats the GST up to around 550 °C, which is lower than the melting temperature but high enough to realize the crystalline transition [14]. Moreover, due to the large resistance margin between crystalline and amorphous states, phase change memory can support multi-bit per cell storage by realizing incomplete phase transition, e.g., researchers have recently demonstrated four-level per cell (i.e., 2-bit per cell) phase-change memory cells with four distinguishable resistance states [7]: very amorphous, amorphous, semi-crystalline, and crystalline.

Phase-change memory, particularly multilevel per cell phase change memory, is subject to a severe storage reliability issue: while the crystalline phase of phase-change chalcogenide material is very stable over the time, the amorphous phase (and other phases obtained by incomplete phase transition in multilevel phase change memory) is metastable and can experience structural relaxation [15], which results in resistance drift over the time. Experimental results [15]–[17] suggest that the resistance tends to drift over the time with t^{ν}, i.e.,

$$R(t) = R_0 \left(\frac{t}{t_0}\right)^{\nu} \tag{1}$$

where R_0 is the initial resistance and t_0 is a normalized time constant, and ν is the drift exponent. Experimental results show that ν strongly depends on the initial resistance and exhibits a large degree of variability. Since R_0 and ν depend on various factors [8], [17], [18], [21], it is very difficult to accurately model their variability through theoretical analysis. Based upon test chip measurements [9], [20], R_0, and ν appear to approximately follow Gaussian distributions. Meanwhile, drift exponent ν tends to increase as the initial resistance increases due to a larger structural relaxation effect. As a result, memory cells in multilevel per cell phase-change memory will experience different resistance drift pace over the time, leading to significant degradation of memory operation margin. This makes multilevel per cell phase-change memory inherently subject to a serious storage reliability challenge.

It has been demonstrated that phase-change memory cell resistance drift can be compensated by periodic self-refresh [21], which is realized by steering a relatively low current through the phase-change material and can push the drifted resistance back to its initial value. Different initial resistance requires different amount of refresh current that must be accurately controlled. Since self-refresh can only be done when the memory chip is power on and memory cell resistance drifts all the time, this technique may not be suitable for many nonvolatile data storage applications. Moreover, it may induce non-negligible amount of energy consumption overhead and bandwidth loss. In particular, for our interested multilevel phase-change memory, self-refresh should be done with a relatively short period (e.g., tens or hundreds seconds). Even though it is still much longer than DRAM self-refresh period (e.g., 64 or 128 ms), DRAM can refresh one entire word-line at one time and one word-line contains hundreds of cells, but phase-change memory cannot refresh too many cells at one time because of the power constraint. As a result, self-refresh may induce noticeable bandwidth loss besides energy consumption overhead. Therefore, this work does not consider the use of self-refresh scheme for improving resistance drift tolerance.

III. Information-Theoretical Bound of Memory Cell Storage Efficiency

In this section, by treating phase-change memory as communication channel, we apply the information theory to formulate the channel capacity (i.e., cell storage efficiency bound) of multilevel per cell phase-change memory. Moreover, we show that conventional time-independent memory fault tolerance tends to largely lag from the theoretical bounds.

A. Theoretical Formulation

For m-level per cell phase-change memory, let random variable D represent the content stored in one memory cell and assume D has an equal probability of being any specific value, i.e., $P(D=i)=1/m$ for $i=1,\ldots,m$. Let random variable R represent the resistance of the memory cell, the cell storage efficiency information-theoretical bound can be formulated as

$$C = H(R) - H(R|D)$$

in which $H(R)$ is the entropy of memory cell resistance R and is calculated as

$$H(R) = -\int p(R) log_2 p(R) dR$$

where $p(R)$ is the probability density function (PDF) of R. $H(R|D)$ is the conditional entropy, i.e., the entropy of R given the knowledge of the content D stored in the memory cell. $H(R|D)$ is calculated as

$$\begin{aligned} H(R|D) &= -\sum_{i=1}^{m} P(D=i) \int p(R|D=i) log_2 p(R|D=i) dR \\ &= -\frac{1}{m}\sum_{i=1}^{m} \int p(R|D=i) log_2 p(R|D=i) dR \end{aligned}$$

where $p(R|D=i)$ is the conditional PDF of R given the memory cell content $D=i$ for $i=1,\ldots,m$. We note that the resistance R of phase-change memory cells is typically represented in logarithmic domain as $\lg R$, where lg represents $\log_{10}$. Since $\lg R$ is a one-to-one function of R, we have $H(R)=H(\lg R)$ and $H(R|D)=H(\lg R|D)$. Hence the information-theoretical bound of memory cell storage efficiency can be rewritten as

$$C = H(\lg R) - H(\lg R|D).$$

Therefore, we can calculate the cell storage efficiency bound by estimating $p(\lg R)$ and $p(\lg R|D=i)$ for $i=1,\ldots,m$. With the equal probability assumption of memory cell content (i.e., $P(D=i)=1/m$ for $i=1,\ldots,m$), we have that $p(\lg R)$ can be calculated as

$$\begin{aligned} p(\lg R) &= \sum_{i=1}^{m} P(D=i) p(\lg R|D=i) \\ &= \frac{1}{m}\sum_{i=1}^{m} p(\lg R|D=i). \end{aligned} \tag{2}$$

Hence, in order to calculate the multilevel phase change memory cell storage efficiency bound, we should accurately estimate the resistance distribution of all the storage levels, i.e., $p(\lg R|D=i)$ for $(i=1,\ldots,m)$. Due to the phase-change material resistance drift and the drift speed variability, the memory cell storage efficiency bound will also vary with time. As a result, we must obtain all the $p(\lg R|D=i)$ at different memory content lifetime, and accordingly estimate memory cell storage efficiency bounds at different memory content lifetime.

The estimation of memory cell storage efficiency bounds can be largely simplified under the assumption of Gaussian distributions. Given a resistance drift exponent ν, we can write the cell resistance drift in logarithmic domain as

$$\lg R(t) = \lg R(1) + \nu \lg t \tag{3}$$

where $\lg R(1)$ is the initial resistance value at time $t = 1$ s. In the logarithmic domain, $\lg R(t)$ increases linearly with $\lg t$. For m-level per cell phase-change memory, let $\lg R_i$ denote the resistance value in logarithmic domain of the ith storage level $(i = 1, \ldots, m)$. We assume that the initial resistance $\lg R_i$ and the drift exponent ν_i follow Gaussian distributions $\mathcal{N}(\mu_{R_i}, \sigma_{R_i}^2)$ and $\mathcal{N}(\mu_{\nu_i}, \sigma_{\nu_i}^2)$, respectively. According to (3), the resistance value $\lg R_i(t_d)$ at the lifetime t_d will also follow a Gaussian distribution $\mathcal{N}(\mu_i(t_d), \sigma_i^2(t_d))$, where

$$\begin{cases} \mu_i(t_d) = \mu_{R_i} + \mu_{\nu_i} \cdot \lg t_d, \\ \sigma_i^2(t_d) = \sigma_{R_i}^2 + \sigma_{\nu_i}^2 \cdot (\lg t_d)^2. \end{cases} \tag{4}$$

Hence, at the lifetime t_d, the $p(\lg R | D = i)$ can be represented as

$$\begin{aligned} p(\lg R | D = i) &= p(\lg R_i) \\ &= \frac{1}{\sqrt{2\pi}\sigma_i(t_d)} e^{-(\lg R - \mu_i(t_d))^2 / 2\sigma_i^2(t_d)}. \end{aligned} \tag{5}$$

By calculating $p(\lg R | D = i)$ for $i = 1, \ldots, m$ according to (5), we can further calculate $p(\lg R)$ at different memory content lifetime t_d according to (2), and hence correspondingly estimate the memory cell storage efficiency bound at different memory content lifetime. In practice, if real phase change memory chips are available, we can carry out extensive experimental measurement to capture the distributions in stead of approximating the distributions as ideal Gaussian distributions.

B. Case Studies

We use the following case studies to demonstrate the above discussion on deriving information-theoretical cell storage efficiency bounds. Let us consider a hypothetical four-level per cell and eight-level per cell phase-change memory. We assume that the initial resistance $\lg R(1)$ of all the storage levels are equally spaced and follow the same Gaussian distribution. The resistance drift exponent ν also follows Gaussian distribution $\mathcal{N}(\mu_\nu, \sigma_\nu^2)$. Table I lists all the configurations. As demonstrated in recent experimental studies [17], a larger initial resistance tends to have a larger μ_ν and σ_ν. Hence, as shown in Table I, we assign monotonically increasing μ_ν for all the storage levels and assume that all the storage levels have the same standard deviation to mean ratio (SDMR) so that the absolute value of σ_ν increases as μ_ν increases.

TABLE I
CONFIGURATIONS OF HYPOTHETICAL FOUR-LEVEL PER CELL AND EIGHT-LEVEL PER CELL PHASE-CHANGE MEMORY

4-level/cell					
storage level	data	$\lg R(1)$		ν	
		mean	deviation	mean	SDMR
1	00	3.0	0.17	0.001	40%
2	01	4.0		0.02	
3	11	5.0		0.06	
4	10	6.0		0.10	

8-level/cell					
storage level	data	$\lg R(1)$		ν	
		mean	deviation	mean	SDMR
1	000	3.0	0.08	0.001	20%
2	001	3.5		0.01	
3	011	4.0		0.02	
4	010	4.5		0.04	
5	110	5.0		0.06	
6	111	5.5		0.08	
7	101	6.0		0.10	
8	100	6.5		0.12	

Fig. 2 shows the accordingly estimated information-theoretical bounds of cell storage efficiency vs. memory content lifetime for the hypothetical four-level per cell and eight-level per cell phase-change memory. Due to the non-negligible variability of drift exponent ν, resistance window of each storage level not only shifts but also becomes wider over the time. This results in more and more significant overlaps between adjacent storage levels over the time, leading to increasingly worse memory storage reliability and hence less memory cell storage efficiency as illustrated in Fig. 2. For example, at the lifetime of 10^{10} s, the cell storage efficiency bound of four-level per cell phase-change memory drops from the initial 1.99 to 1.94 bits/cell, while the bound of eight-level per cell phase-change memory drops from the initial 2.99 to 2.88 bits/cell.

As pointed out in the above, ECC should be used to ensure the overall data storage integrity of multilevel per cell phase-change memory. In this work, we consider two types of ECC, including 1) classical hard-decision ECC such as BCH code, where the decoding only needs hard decision of memory content (i.e., memory sensing only needs to make binary estimation on each bit) and 2) advanced soft-decision ECC such as LDPC code, where the decoding needs probability information of memory content (i.e., memory sensing should not only make binary estimation on each bit but also provide the probability or confidence of the decision). Since we are interested in the use of multilevel per cell phase-change memory as block-oriented data storage, we set each ECC codeword protects 4 kB user data similar to present NAND flash memory based solid-state drive. We construct three binary BCH codes over $GF(2^{16})$, BCH (34816, 32768, 128), BCH (36864, 32768, 256), and BCH (30960, 32768, 512), where each BCH codeword protects 32768 user bits and can correct up to 128-, 256-, and 512-bit errors, respectively. In terms of LDPC codes, we focus on the use of quasi-cyclic LDPC (QC-LDPC) code, a special family of LDPC codes that can largely simplify decoder design and hence being widely used in practice. The parity check matrix of a QC-LDPC code can be written as

$$H = \begin{bmatrix} H_{1,1} & H_{1,2} & \ldots & H_{1,n} \\ H_{2,1} & H_{2,2} & \ldots & H_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ H_{m,1} & H_{m,2} & \ldots & H_{m,n} \end{bmatrix}$$

where each sub-matrix $\mathbf{H}_{i,j}$ is a circulant matrix over $GF(2)$. We note that a circulant matrix is a $p \times p$ square matrix in which each row is the cyclic shift of the row above it, and the first row is the cyclic shift of the last row. The weight γ of the circulant matrix is the number of 1 s in each row/column. In this work, we focus on randomly constructed regular QC-LDPC codes, where all the circulant matrices have the same weight. Given the matrix structural parameters m and n, the LDPC code rate is $(n-m)/n$. In this work, we construct a length-34816, rate-0.94 QC-LDPC code and a length-36864, rate-0.89 QC-LDPC code. The parity check matrix of the length-34816, rate-0.94 QC-LDPC code contains an array of 2×34 circulant matrices (i.e., $m = 2$, $n = 34$), and each circulant matrix is 1024×1024 (i.e., $p = 1024$) and has a weight γ of 2; The parity check matrix of the length-36864, rate-0.89 QC-LDPC code contains an array of 4×36 circulant matrices (i.e., $m = 4$, $n = 36$), and each circulant matrix is 1024×1024 (i.e., $p = 1024$) and has a weight γ of 1. All the circulant matrices in this work are constructed randomly subject to the constraint that the entire code parity check matrix corresponds to a bipartite graph without any four-cycles.

We estimate the cell storage efficiency achieved by these BCH and LDPC codes when simply using a time-independent static memory cell resistance quantization, i.e., the memory cell resistance quantization is completely unaware of the memory content lifetime and is simply carried out based on the original resistance distribution. Following the convention in NAND flash memory, we refer each phase-change memory word containing 4 kB user data as a page. We set the page error rate (PER), i.e., the ECC codeword decoding failure rate, of 10^{-20} and below as the objective of using ECC to ensure overall memory storage integrity. For the three binary BCH codes, we first calculate the raw bit error rate (BER) (i.e., the BER if we determine the binary decision of each bit directly based on memory sensing results without ECC processing) under which they can achieve an PER below 10^{-20}. Let P_e denote the raw BER, N_{word} denote the BCH codeword length, and t denote the maximum number of errors that BCH code can correct in each codeword. The PER is given by

$$\text{PER} = 1 - \sum_{i=0}^{t} \left(\binom{N_{\text{word}}}{i} \cdot (1-P_e)^{(N_{\text{word}}-i)} \cdot P_e^i \right)$$

and we search the raw BER that satisfies $\text{PER} \leq 10^{-20}$. Then we search for the lifetime t_d which will result in the corresponding raw BER given the memory configurations listed in Table I. The results are shown in Fig. 2, where the three points correspond to the three BCH codes being used. With respect to LDPC codes, because of the lack of accurate analytical methods for calculating/predicting LDPC code decoding failure rates, we have to rely on computer-based Monte Carlo simulations. Because of the extremely prohibitive computational complexity required for simulating LDPC code decoding at PER of 10^{-20}, we simulate the LDPC code decoding at PER of 10^{-5} and above, based on which we extrapolate the curve down to the PER of 10^{-20}. The min-sum LDPC decoding algorithm [22] is used with up to 32 internal decoding iterations. The results are shown in Fig. 2, where the two points correspond to the two LDPC codes being used. Regarding to the finite word-length configuration used in the analysis and simulation, we use three integer bits and four fractional bits to represent each memory cell resistance quantization threshold, and set each LDPC decoding message as 3 bits.

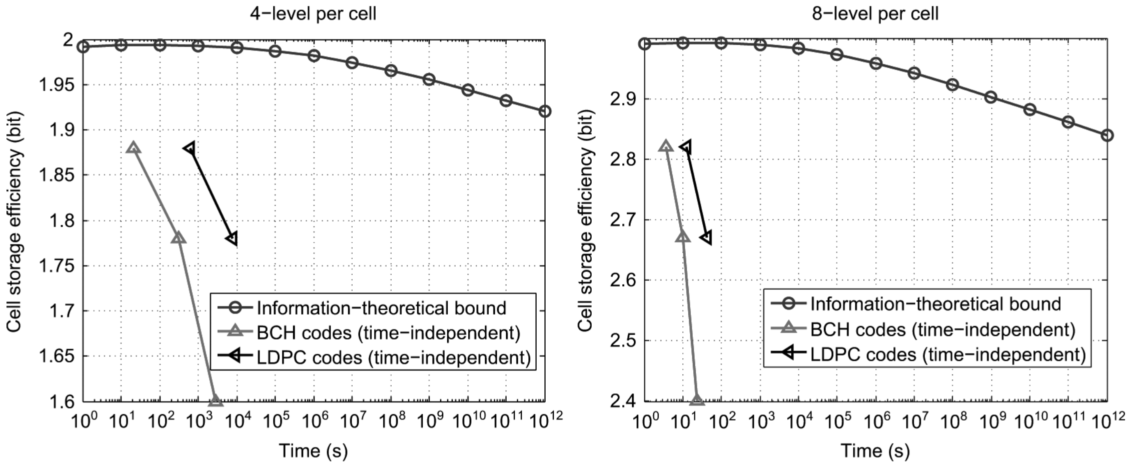
Fig. 2 clearly shows a very large gap between the information-theoretical bounds and achievable performance when using BCH and LDPC codes with time-independent static memory cell resistance quantization. As shown in Fig. 2, the cell storage efficiency information-theoretical bounds of four-level per cell and eight-level per cell phase-change memory at the memory cell content lifetime of 10^7 s (i.e., about four months) are 1.97- and 2.94-bit, respectively, which means the phase-change memory can achieve error-free storage with code rates as high as $1.97/2 = 0.985$ for four-level per cell phase-change memory and with code rates as high as $2.94/3 = 0.98$ for eight-level per cell phase-change memory at memory cell content lifetime of 10^7 s. However, even with much lower coding rates hence much stronger error correcting capability, those BCH and LDPC codes considered above can only enable memory cell content lifetime up to 10^4s (i.e., about 3 h). Fig. 3 further shows the calculated and simulated PER performance of the BCH and LDPC codes with the code rate of 0.94 using the time-independent static memory cell resistance quantization. The simulation results clearly show very abrupt PER drops over the time, leading to very short allowable memory cell content lifetime. For example, the four-level per cell phase-change memory can only achieve a PER of 10^{-3} as the memory cell content lifetime reaches 10^4 s (i.e., about 3 h) even using the powerful LDPC code as shown in Fig. 3, and it becomes much worse for the eight-level per cell phase-change memory as shown in Fig. 3.

The above results clearly suggest that the conventional time-independent realization of memory fault tolerance leaves a very big gap from the information-theoretical bounds. This directly motivates us to investigate the potential of incorporating time awareness into memory fault tolerance, as described in the remainder of this paper.

IV. Proposed Time-Aware Memory Fault Tolerance

A. *Intuitive Basic Idea*

With the objective of reducing the gap from the information-theoretical bounds, we propose a time-aware memory fault tolerance design method, which aims to keep track of memory content lifetime and accordingly adjust how to quantize and interpret memory cell resistance to improve ECC decoding performance. Its basic idea is very intuitive and can be illustrated in Fig. 4. It shows the resistance drift of four-level per cell phase-change memory cells. Since a larger initial resistance tends to have a larger drift exponent, higher storage levels with larger initial resistance are subject to bigger resistance drift, as illustrated in Fig. 4. Assume memory cells are written at $t = 0$ and are read at $t = T$. As illustrated in Fig. 4, if we use a static memory cell resistance quantization, i.e., the memory cell resistance quantization thresholds are fixed, a noticeable amount of memory read errors will be induced. On the other hand, if we can keep track of the lifetime of memory cells, we can estimate the resistance drift parameters based upon resistance drift statistical model, and hence dynamically adjust the memory cell resistance quantization thresholds to minimize the memory read



(a) (b)

Fig. 2. Information-theoretical cell storage efficiency bounds and cell storage efficiency achieved by BCH and LDPC codes without knowledge of memory content lifetime for (a) four-level per cell and (b) eight-level per cell phase-change memory.

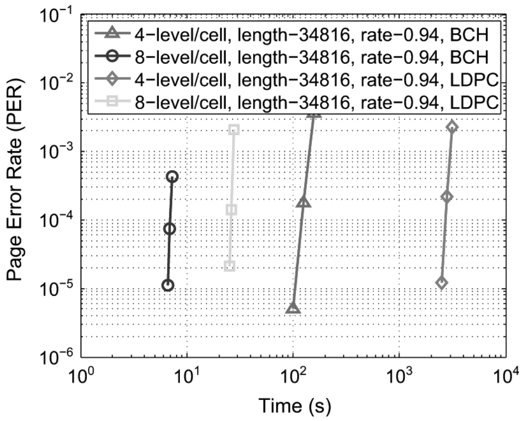


Fig. 3. PER performance of using rate-0.94 BCH and LDPC codes without knowledge of memory content lifetime for four-level per cell eight-level per cell phase-change memory.

errors, as illustrated in Fig. 4. Clearly, the time-aware memory cell resistance quantization as illustrated in Fig. 4 is only sufficient for hard-decision ECC. If soft-decision ECC is being used, we have to use a finer-grained quantization and carry out more sophisticated computation to estimate the probability information for each bit in a time-aware manner, which will be elaborated later.

Fig. 5 shows a generic system structure using time-aware memory fault tolerance. During the write operation, the user data are first encoded by an ECC encoder within a separate memory controller chip, then the codeword is written into the multilevel per cell phase-change memory data array. At the same time, a small memory called *time tag* is used to record the present time. When the data is being read from the memory data array, its associated time information is also read from the time tag, based on which we can estimate the data lifetime t_d. If hard-decision ECC is being used, we can use the estimated lifetime t_d to determine the optimal quantization thresholds that can minimize the hard-decision BER. If soft-decision ECC is being used, the estimated lifetime t_d is used to derive the probability information of each bit. Therefore, both fine-grained quantization results and estimated lifetime t_d should be delivered to the controller chip in the case of soft-decision ECC, as illustrated in Fig. 5.

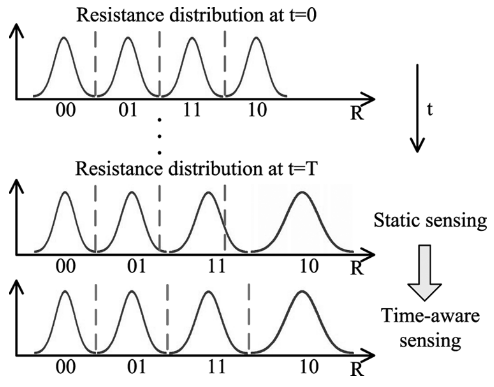

Resistance distribution at t=0
00
01
11
10
R
t
Resistance distribution at t=T
00
01
11
10
R
Static sensing
Time-aware sensing
00
01
11
10
R

Fig. 4. Simple example to illustrate the basic idea of time-aware memory fault tolerance.

B. Use of Estimated Lifetime t_d

We further elaborate on how the estimated data lifetime t_d should be used for both hard-decision and soft-decision ECC once we assume memory parameters follow Gaussian distributions. As discussed in Section III, in the logarithmic domain, $\lg R(t)$ increases linearly with $\lg t$. Hence, because the initial resistance $\lg R_i$ and the drift exponent ν_i follow Gaussian distributions, the resistance value $\lg R_i(t_d)$ at the lifetime t_d will also follow a Gaussian distribution $\mathcal{N}(\mu_i(t_d), \sigma_i^2(t_d))$, which can be calculated according to (4). For hard-decision ECC, we could use the calculated $\{\mu_i(t_d), \sigma_i^2(t_d)\}$ for $i = 1, \ldots, m$ to determine the $m-1$ optimal memory cell resistance quantization thresholds that can minimize the hard-decision BER. Let $T_i(t_d)$ denote the quantization threshold between the memory cell resistance level i and $i+1$, we can calculate the error probability for each resistance level i, denoted as RER_i, as

$$\begin{cases} Q(\frac{T_i(t_d)-\mu_i(t_d)}{\sigma_i(t_d)}), & i=1 \\ 1-Q(\frac{T_{i-1}(t_d)-\mu_i(t_d)}{\sigma_i(t_d)}), & i=m \\ 1-Q(\frac{T_{i-1}(t_d)-\mu_i(t_d)}{\sigma_i(t_d)})+Q(\frac{T_i(t_d)-\mu_i(t_d)}{\sigma_i(t_d)}), & \text{otherwise} \end{cases}$$

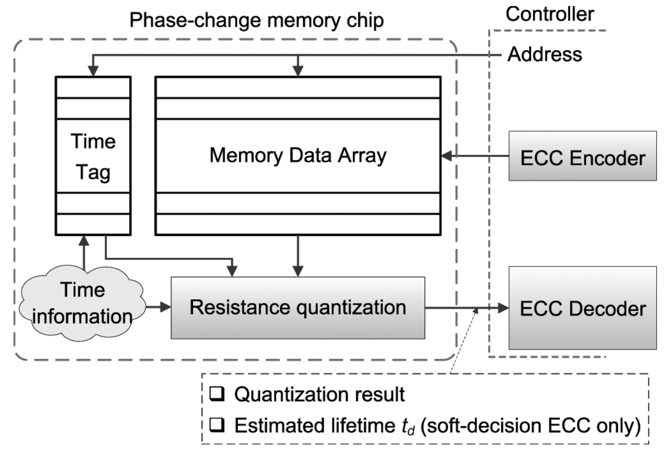


Fig. 5. Generic multilevel phase-change memory system with time-aware memory fault tolerance.

where $Q(x)$ is the Q-function for the standard normal distribution, which is defined as $\int_x^\infty 1/\sqrt{2\pi}e^{-t^2/2}dt$. Assuming all the m memory cell resistance levels are equally possible with a probability of $1/m$, the total memory read error rate $\mathrm{RER}_{\mathrm{tot}}$ can be represented as

$$\begin{aligned}\mathrm{RER}_{\mathrm{tot}} &= \frac{1}{m}\sum_{i=1}^{m}\mathrm{RER}_i\\ &= \frac{1}{m}\sum_{i=1}^{m-1}\left(1-Q\left(\frac{T_i(t_d)-\mu_{i+1}(t_d)}{\sigma_{i+1}(t_d)}\right)\right.\\ &\qquad\left.+Q\left(\frac{T_i(t_d)-\mu_i(t_d)}{\sigma_i(t_d)}\right)\right). \qquad (6)\end{aligned}$$

Clearly, the objective is to adjust all the $m-1$ quantization thresholds $T_i(t_d)$ for $i=1,\ldots,m-1$ so that the memory cell hard-decision quantization error rate $\mathrm{RER}_{\mathrm{tot}}$ is minimized, i.e.,

$$\underset{T_1(t_d),\ldots,T_{m-1}(t_d)}{\arg\min}\left(\mathrm{RER}_{\mathrm{tot}}\right). \qquad (7)$$

Since all the items in (6) for $i=1,\ldots,m-1$ are independent to each other, minimization of $\mathrm{RER}_{\mathrm{tot}}$ is equivalent to the minimization of all the items separately. Therefore, the selection of $T_i(t_d)$ only depends on the resistance distributions of the resistance level i and $i+1$, i.e.,

$$\underset{T_i(t_d)}{\arg\min}\left(1-Q\left(\frac{T_i(t_d)-\mu_{i+1}(t_d)}{\sigma_{i+1}(t_d)}\right)+Q\left(\frac{T_i(t_d)-\mu_i(t_d)}{\sigma_i(t_d)}\right)\right).$$

Hence, each optimal memory cell resistance quantization threshold $T_i(t_d)$ should satisfy

$$\frac{\partial\left(1-Q\left(\frac{T_i(t_d)-\mu_{i+1}(t_d)}{\sigma_{i+1}(t_d)}\right)+Q\left(\frac{T_i(t_d)-\mu_i(t_d)}{\sigma_i(t_d)}\right)\right)}{\partial T_i(t_d)}=0.$$

Recall that $Q(x)=\int_x^\infty 1/\sqrt{2\pi}e^{-t^2/2}dt$, the above equation can be reduced to

$$\frac{T_i(t_d)-\mu_{i+1}(t_d)}{\sigma_{i+1}(t_d)}-\frac{T_i(t_d)-\mu_i(t_d)}{\sigma_i(t_d)}=0$$

which directly leads to

$$T_i(t_d)=\frac{\mu_{i+1}(t_d)\sigma_i(t_d)+\mu_i(t_d)\sigma_{i+1}(t_d)}{\sigma_{i+1}(t_d)+\sigma_i(t_d)}. \qquad (8)$$

Therefore, given the time-aware resistance level statistical distribution parameters, we can use (8) to calculate the $m-1$ optimal memory cell resistance quantization thresholds when hard-decision ECCs are being used.

When soft-decision ECCs are being used, more sophisticated soft decision [i.e., log-likelihood ratio (LLR)] should be estimated for each bit. Let $\lg R(t_d)$ denote the quantized memory cell resistance value with finer granularity and assume each bit has equal probability of being 0 and 1, the LLR for each bit x_i can be expressed as

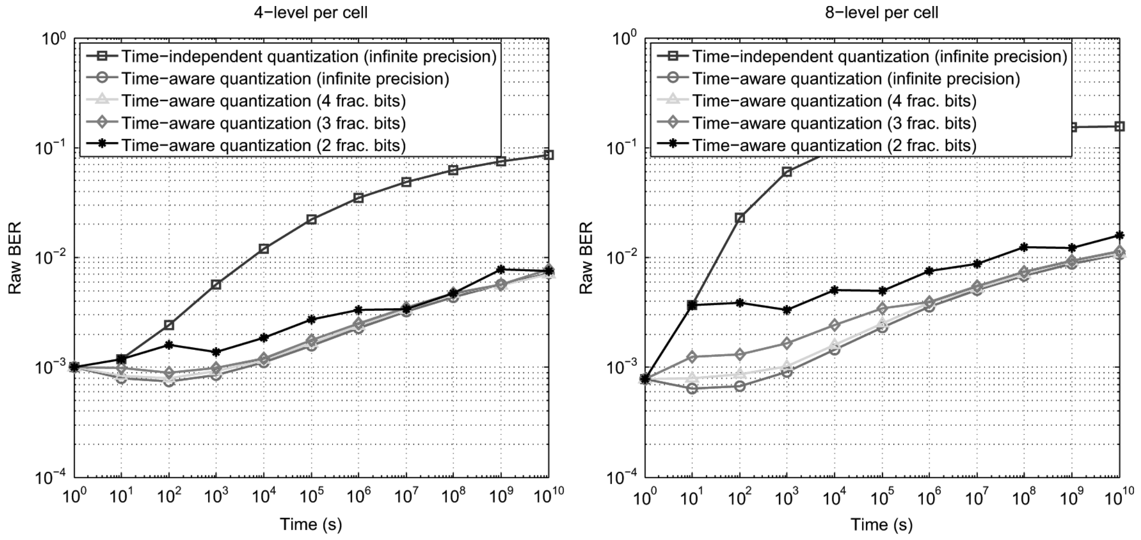
$$\begin{aligned}L(x_i|\lg R(t_d)) &= \ln\frac{P(\lg R(t_d)|x_i=1)}{P(\lg R(t_d)|x_i=0)}\\ &= \ln P(\lg R(t_d)|x_i=1)\\ &\quad-\ln P(\lg R(t_d)|x_i=0). \qquad (9)\end{aligned}$$

Suppose each memory cell stores n bits (i.e., $m=2^n$), and let $\mathbf{s}^k$ denote the n-bit binary vector represented by the resistance level k and s_i^k represent the ith bit within the vector $\mathbf{s}^k$, then we can calculate the probabilities $P(\lg R(t_d)|x_i=1)$ and $P(\lg R(t_d)|x_i=0)$ as

$$\begin{aligned}P(\lg R(t_d)|x_i{=}1) &= \frac{1}{m}\sum_{s_i^k=1,k\in[1,m]}\\ &\quad\times\frac{1}{\sqrt{2\pi}\sigma_k(t_d)}e^{-(\lg R(t_d)-\mu_k(t_d))^2/2\sigma_k^2(t_d)}\\ &= \frac{1}{\sqrt{2\pi}m}\sum_{s_i^k=1,k\in[1,m]}\\ &\quad\times e^{-(\lg R(t_d)-\mu_k(t_d))^2/2\sigma_k^2(t_d)-ln\sigma_k(t_d)}\\ P(\lg R(t_d)|x_i{=}0) &= \frac{1}{m}\sum_{s_i^k=0,k\in[1,m]}\\ &\quad\times\frac{1}{\sqrt{2\pi}\sigma_k(t_d)}e^{-(\lg R(t_d)-\mu_k(t_d))^2/2\sigma_k^2(t_d)}\\ &= \frac{1}{\sqrt{2\pi}m}\sum_{s_i^k=0,k\in[1,m]}\\ &\quad\times e^{-(\lg R(t_d)-\mu_k(t_d))^2/2\sigma_k^2(t_d)-ln\sigma_k(t_d)}.\end{aligned}$$

To simplify the computation of $\ln P(\lg R(t_d)|x_i=1)$ and $\ln P(\lg R(t_d)|x_i=0)$ as presented above, it is a convention to apply the MAX-LOG-MAP approximation, i.e., $\ln(e^x+e^y)\approx\max(x,y)$, and ignore the common factor $1/\sqrt{2\pi}m$, hence we have

$$\begin{aligned}\ln P(\lg R(t_d)|x_i{=}1) &\approx \max_{s_i^k=1,k\in[1,m]}\\ &\quad\times\left\{-\frac{(\lg R(t_d)-\mu_k(t_d))^2}{2\sigma_k^2(t_d)}-\ln\sigma_k(t_d)\right\}\\ &= -\min_{s_i^k=1,k\in[1,m]}\\ &\quad\times\left\{\frac{(\lg R(t_d)-\mu_k(t_d))^2}{2\sigma_k^2(t_d)}+\ln\sigma_k(t_d)\right\}\end{aligned} \qquad (10)$$



(a) (b)

Fig. 6. Raw BER performance comparison when using time-independent and time-aware quantization schemes for (a) four-level per cell and (b) eight-level per cell phase-change memory.

and

$$\begin{aligned}\ln P(\lg R(t_d)|x_i=0)\approx & \max_{s_i^k=0,k\in[1,m]} \\ & \times\left\{-\frac{(\lg R(t_d)-\mu_k(t_d))^2}{2\sigma_k^2(t_d)}-\ln\sigma_k(t_d)\right\} \\ =- & \min_{s_i^k=0,k\in[1,m]} \\ & \times\left\{\frac{(\lg R(t_d)-\mu_i(t_d))^2}{2\sigma_k^2(t_d)}+\ln\sigma_k(t_d)\right\}.\end{aligned} \tag{11}$$

The above approximation can largely reduce the computational complexity at relatively small accuracy degradation. Combining (9)–(11), we have that the LLR for each bit x_i can be estimated as

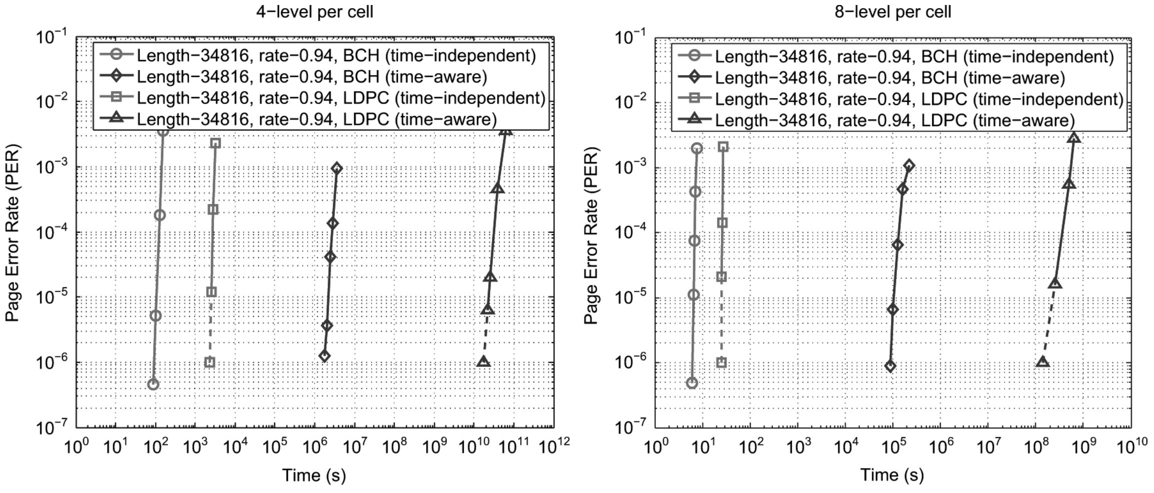
$$\begin{aligned}&L(x_i|\lg R(t_d)) = \\ &\min_{s_i^k=0,k\in[1,m]}\left\{\frac{(\lg R(t_d)-\mu_k(t_d))^2}{2\sigma_k^2(t_d)}+\ln\sigma_k(t_d)\right\} \\ &-\min_{s_i^k=1,k\in[1,m]}\left\{\frac{(\lg R(t_d)-\mu_k(t_d))^2}{2\sigma_k^2(t_d)}+\ln\sigma_k(t_d)\right\}.\end{aligned}$$

For the practical implementation of the above LLR computation, the value $1/2\sigma_k^2(t_d)$ can be precomputed to largely reduce the division operations, and $\ln\sigma_k(t_d)$ can be obtained through a fixed lookup table (LUT).

C. Case Studies

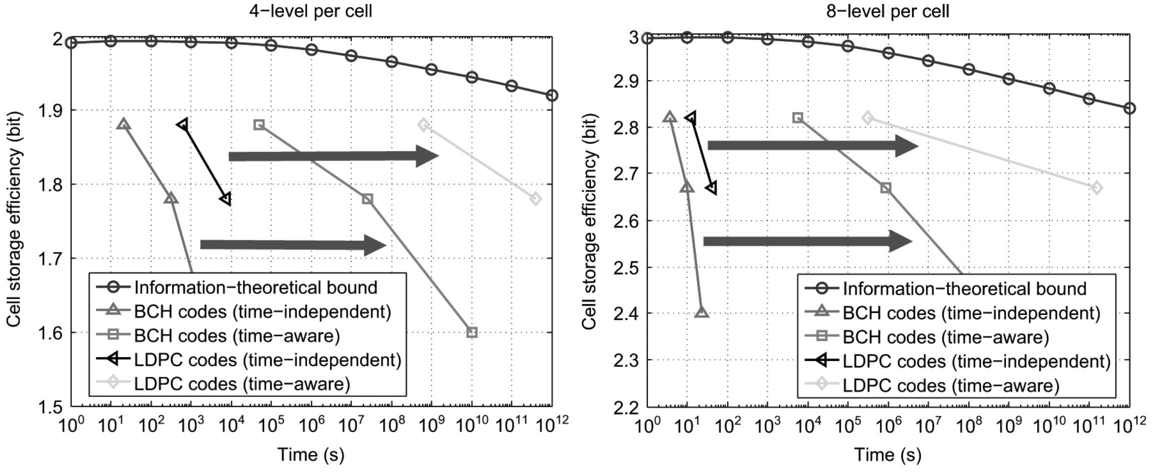
Following the same configuration as listed in Table I for a hypothetical four-level per cell and eight-level per cell phase-change memory, we evaluate the above presented time-aware memory fault tolerance using the same three binary BCH and two QC-LDPC codes as in Section III. Throughout this paper, we assume there is an external resource to feed the phase-change memory chip with accurate 33-bit time information including 7-bit for year, 4-bit for month, 5-bit for day, 5-bit for hour, 6-bit for minute, and 6-bit for second. We should use 1-bit per cell configuration for the time tag block to ensure the storage integrity. Because each entry in the time tag memory block only needs 33 memory cells, the area overhead of this time tag block is completely negligible compared with the main data array in which each entry stores 4 kB user data.

First, we study and compare the raw BER when using time-independent and time-aware hard-decision memory cell resistance quantization. In the case of time-aware hard-decision memory cell resistance quantization, we consider different finite precisions (i.e., three integer bits and $2\sim 4$ fractional bits) of quantization thresholds calculated based on (8). Fig. 6 shows the results and comparisons, which clearly demonstrates that time-aware hard-decision memory cell resistance quantization can largely reduce the raw BER for both four-level per cell and eight-level per cell scenarios. As shown in Fig. 6, when using time-aware hard-decision quantization, the raw BER slightly decreases at the beginning and then moderately increases over the time. This counter-intuitive behavior can be intuitively explained as follows. Since memory cells with higher initial resistance value have larger resistance drift exponents, the distance between the mean of adjacent resistance levels will increase over the time, which tends to reduce the raw BER. On the other hand, the resistance distribution of each level spreads over the time due to the variability of resistance drift exponent, which tends to increase distribution overlaps between adjacent storage levels and hence increase the raw BER. At the beginning, the resistance distribution spread is relatively small and hence its impact on raw BER can be offset by the increased distance between mean of adjacent resistance levels, leading to a net slight decrease of raw BER. However, the resistance distribution spread continues to accumulate over the time and quickly become dominant compared with increased distance between mean of adjacent resistance levels, leading to continuous net increase of raw BER, as shown in Fig. 6.



(a) (b)

Fig. 7. PER performance comparison between time-independent and time-aware fault tolerance when using BCH and LDPC codes for (a) four-level per cell and (b) eight-level per cell phase-change memory.



(a) (b)

Fig. 8. Comparison of cell storage efficiency theoretical bounds and cell storage efficiency achieved by BCH and LDPC codes for (a) four-level per cell and (b) eight-level per cell phase-change memory.

Next, we consider the scenarios when BCH and LDPC codes are being used. As pointed out earlier, BCH code decoder only requests the hard-decision of each bit, while LDPC code decoder requests the soft-decision LLR of each bit. Following the configurations in the case studies presented in Section III, we set three integer bits and four fractional bits to represent the quantization thresholds, and each LLR for LDPC code decoding is represented with 3 bits and the internal LDPC decoding messages also use 3 bits. Similarly, the LDPC code decoding is carried out using the min-sum decoding algorithm with up to 32 internal decoding iterations. Fig. 7 shows the corresponding PER, for both four-level per cell and eight-level per cell phase-change memory. Similarly, because simulation of LDPC code decoding is very computation intensive, the solid parts of LDPC performance curves are obtained from Monte Carlo simulations while the dashed parts are simply extrapolated, as shown in Fig. 7.

As discussed and demonstrated in Section III, using time-independent static quantization, the PER quickly increases over the time, leading to very short allowable memory content lifetime. Results in Fig. 7 clearly show that such a simple time-aware design strategy can significantly increase the allowable memory content lifetime by several orders of magnitude. As shown in Fig. 7, when BCH code is being used, the PER with time-aware memory cell resistance quantization achieves 10^{-6} at 10^6 s and 10^5 s for four-level per cell and eight-level per cell phase-change memory, representing about four orders of magnitude improvement over its time-independent counterpart. When LDPC code is being used, the PER with time-aware LLR estimation achieves 10^{-6} at 10^{10} s and 10^8 s for four-level per cell and eight-level per cell phase-change memory, representing about seven orders of magnitude improvement over its time-independent counterpart. Moreover, we note that, although both four-level per cell and eight-level per cell phase-change memory have the same raw BER at the beginning, the four-level per cell phase-change memory can allow a much longer memory content lifetime compared with its eight-level per cell counterpart, which suggests a clear trade-off between the multilevel phase-change memory storage density and memory content lifetime.

We further carry out analysis and simulation to estimate the achievable cell storage efficiency when using BCH and LDPC codes under the framework of time-aware memory fault tolerance. We use the same three BCH code and two LDPC codes as in Section III, we set the PER of 10^{-20} as the target and extrapolate the LDPC code performance from the simulation result at the PER of 10^{-5}. The results are shown in Fig. 8, in which the three points on the BCH curves correspond to the three BCH codes and the two points on the LDPC curves correspond to the two LDPC codes. This clearly shows a large performance improvement and much reduced gap from the information-theoretical bounds when using the proposed time-aware fault tolerance design strategy. Results suggest that the proposed time-aware fault tolerance design strategy yields a better performance improvement for LDPC as compared to BCH code. It can be intuitively explained as follow: For BCH codes, we only estimate the hard decision value of each data bit (i.e., logic 0 or 1) according to the updated resistance distribution, while for LDPC codes, besides the hard decision value, we also estimate more sophisticated soft information, i.e., the reliability of being logic 0 or 1, which can more accurately capture time-dependent resistance drift. LDPC code decoding can utilize such soft information to achieve superior error correction performance than BCH code decoding. Finally, we note that the resistance drift over time will cause the ever-increasing resistance level overlapping, and eventually could make memory fault tolerance unfunctional, no matter how powerful the fault tolerance schemes are. The objective of the above presented design method is to slow down the effect of the resistance drift on memory read reliability and hence increase memory content lifetime, as successfully demonstrated in the above case studies.

V. Conclusion

In this work, we study and demonstrate the use of time-aware memory fault tolerance for multilevel per cell phase-change memory in the presence of significant memory cell resistance drift. Due to the inherent structural relaxation effect, phase-change material is subject to resistance drift with significant variability over the time, which inevitably results in storage reliability degradation and severe constraint on allowable memory content lifetime. This paper presents a simple yet effective time-aware memory fault tolerance design strategy to address this challenge. Under this time-aware fault tolerance design framework, we discuss the use of two types of ECCs that demand either hard-decision input or soft-decision probability input for the decoding. We demonstrate its effectiveness using hypothetical four-level per cell and eight-level per cell phase change memory as test vehicles and show that it can largely reduce the gap from the information-theoretical bounds compared with its time-independent counterpart.

References

[1] C. H. Lam, “The quest for the universal semiconductor memory,” in *Proc. IEEE Conf. Electron Devices Solid-State Circuits*, Dec. 2005, pp. 327–331.

[2] K. Kim and G. Jeong, “Memory technologies for sub-40 nm node,” in *Proc. IEEE Int. Electron Devices Meet. (IEDM)*, Dec. 2007, pp. 27–30.

[3] J. A. Hutchby, R. Cavin, V. Zhirnov, J. E. Brewer, and G. Bourianoff, “Emerging nanoscale memory and logic devices: A critical assessment,” *Computer*, vol. 41, pp. 28–32, May 2008.

[4] F. Bedeschi, R. Fackenthal, C. Resta, E. M. Donze, M. Jagasivamani, E. C. Buda, F. Pellizzer, D. W. Chow, A. Cabrini, G. Calvi, R. Faravelli, A. Fantini, G. Torelli, D. Mills, R. Gastaldi, and G. Casagrande, “A bipolar-selected phase change memory featuring multi-level cell storage,” *IEEE J. Solid-State Circuits*, vol. 44, no. 1, pp. 217–227, Jan. 2009.

[5] R. Bez, “Chalcogenide PCM: A memory technology for next decade,” presented at the IEEE Int. Electron Devices Meet. (IEDM), Baltimore, MD, Dec. 2009.

[6] S. Raoux, G. W. Burr, M. J. Breitwisch, C. T. Rettner, Y.-C. Chen, R. M. Shelby, M. Salinga, D. Krebs, S.-H. Chen, H.-L. Lung, and C. H. Lam, “Phase-change random access memory: A scalable technology,” *IBM J. Res. Developm.*, vol. 52, pp. 465–479, Jul. 2008.

[7] F. Bedeschi, R. Fackenthal, C. Resta, E. M. Donze, M. Jagasivamani, E. Buda, F. Pellizzer, D. Chow, A. Cabrini, G. M. A. Calvi, R. Faravelli, A. Fantini, G. Torelli, D. Mills, R. Gastaldi, and G. Casagrande, “A multi-level-cell bipolar-selected phase-change memory,” in *Proc. Int. Solid-State Circuits Conf.*, San Francisco, CA, Feb. 2008, pp. 428–625.

[8] T. Nirschl, J. B. Phipp, T. D. Happ, G. W. Burr, B. Rajendran, M. H. Lee, A. Schrott, M. Yang, M. Breitwisch, C. F. Chen, E. Joseph, M. Lamorey, R. Cheek, S. H. Chen, S. Zaidi, S. Raoux, Y. C. Chen, Y. Zhu, R. Bergmann, H. L. Lung, and C. Lam, “Write strategies for 2 and 4-bit multi-level phase-change memory,” in *Proc. Int. Electron Devices Meet. (IEDM)*, Washington, DC, Dec. 2007, pp. 461–464.

[9] S. Kang, W. Y. Cho, B. H. Cho, K. J. Lee, C. S. Lee, H. R. Oh, B. G. Choi, Q. Wang, H. J. Kim, M. H. Park, Y. H. Ro, S. Kim, C. D. Ha, K. S. Kim, Y. R. Kim, D. E. Kim, C. K. Kwak, H. G. Byun, G. Jeong, H. Jeong, K. Kim, and Y. Shin, “A 0.1-μm 1.8-V 256-Mb phase-change random access memory (PRAM) with 66-MHz synchronous burst-read operation,” *IEEE J. Solid-State Circuits*, vol. 42, no. 1, pp. 210–218, Jan. 2007.

[10] H. Oh, B. Cho, W. Y. Cho, S. Kang, B. Choi, H. Kim, K. Kim, D. Kim, C. Kwak, H. Byun, G. Jeong, H. Jeong, and K. Kim, “Enhanced write performance of a 64-MB phase-change random access memory,” *IEEE J. Solid-State Circiuts*, vol. 41, no. 1, pp. 122–126, Jan. 2006.

[11] K. Osada, T. Kawahara, R. Takemura, N. Kitai, N. Takaura, N. Matsuzaki, K. Kurotsuchi, H. Moriya, and M. Moniwa, “Phase change RAM operated with 1.5-V CMOS as low cost embedded memory,” in *Proc. Custom Integr. Circuits Conf.*, San Jose, CA, Sep. 2005, pp. 431–434.

[12] R. F. Freitas and W. W. Wilcke, “Storage-class memory: The next storage system technology,” *IBM J. Res.Developm.*, vol. 52, pp. 439–447, Jul. 2008.

[13] G. Bur, B. Kurdi, J. Scott, C. Lam, K. Gopalakrishnan, and R. Shenoy, “Overview of candidate device technologies for storage-class memory,” *IBM J. Res. Developm.*, vol. 52, pp. 449–464, Jul. 2008.

[14] A. L. Lacaita, “Phase change memories: State-of-the-art, challenges and perspectives,” *Solid-State Electron.*, vol. 50, pp. 24–31, Jan. 2006.

[15] A. Pirovano, A. L. Lacaita, F. Pellizzer, S. A. Kostylev, A. Benvenuti, and R. Bez, “Lowfield amorphous state resistance and threshold voltage drift in chalcogenide materials,” *IEEE Trans. Electron Devices*, vol. 51, no. 5, pp. 714–719, May 2004.

[16] D. Ielmini, A. L. Lacaita, and D. Mantegazza, “Recovery and drift dynamics of resistance and threshold voltages in phase-change memories,” *IEEE Trans.Electron Devices*, vol. 54, no. 2, pp. 308–315, Feb. 2007.

[17] D. Ielmini, D. Sharma, S. Lavizzari, and A. L. Lacaita, “Reliability impact of chalcogenide-structure relaxation in phase-change memory (PCM) cells— part I: Experimental study,” *IEEE Trans. Electron Devices*, vol. 56, no. 5, pp. 1070–1077, May 2009.

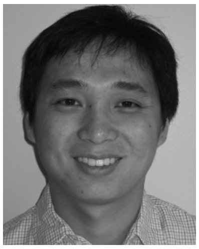
[18] S. Braga, A. Cabrini, and G. Torelli, “Dependence of resistance drift on the amorphous cap size in phase change memory arrays,” *Appl. Phys. Lett.*, vol. 94, pp. 092112–092112-3, 2009.

[19] T. M. Cover and J. A. Thomas, *Element of Information Theory*. New York: Wiley, 1991.

[20] M. Boniardi, D. Ielmini, S. Lavizzari, A. L. Lacaita, A. Redaelli, and A. Pirovano, “Statistical and scaling behavior of structural relaxation effects in phase-change memory (PCM) devices,” in *Proc. IEEE Int. Reliab. Phys. Symp.*, Apr. 2009, pp. 122–127.

[21] S. Kostylev, “Drift of programmed resistance in electrical phase change memory devices,” presented at the Euro. Phase Change Ovonics Symp. (EPCOS), Prague, Czech, Sep. 2008.

[22] J. Chen, A. Dholakia, E. Eleftheriou, M. P. C. Fossorier, and X.-Y. Hu, “Reduced-complexity decoding of LDPC codes,” *IEEE Trans. Commun.*, vol. 53, no. 7, pp. 1288–1299, Jul. 2005.



Wei Xu received the B.S. and M.S. degrees from Fudan University, Shanghai, China, in 2003 and 2006, respectively, and the Ph.D. degree in electrical engineering from Rensselaer Polytechnic Institute, Troy, NY, in 2009.

He joined the Marvell Technology, Santa Clara, CA, in 2009 as a Senior Design Engineer. His research interests include circuit and system design for memory and data storage systems, including phase change memory, spin-transfer torque magnetoresistive memory, solid-state drive and hard disk drive.

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

Currently, he is an Associate Professor with the Electrical, Computer, and Systems Engineering Department, Rensselaer Polytechnic Institute, Troy, NY. His current research interests include algorithm and architecture codesign for communication and data storage systems, variation-tolerant signal processing IC design, fault-tolerant system design for digital memory, and computer architecture.