Enabling Technologies for Multilevel Phase-Change Memory

H. Pozidis, N. Papandreou, A. Sebastian, A. Pantazi, T. Mittelholzer, G. F. Close and E. Eleftheriou
IBM Research – Zurich, CH-8803 Ruschlikon, Switzerland, email: hap@zurich.ibm.com

Abstract

Phase-change memory (PCM) is currently regarded as the most promising new nonvolatile solid-state memory technology for applications that DRAM and NAND-Flash, today's incumbent memories, cannot address. Multilevel-cell (MLC) storage, the most effective way of reducing the cost-per-bit in memory technologies, is necessary for PCM to be competitive. MLC capability in PCM is challenging, as process and material variability in large memory cell arrays require iterative programming algorithms to achieve tight level distributions. Furthermore, MLC PCM is hampered by the phenomenon of resistance drift. In this paper, we describe a number of novel technologies that enable reliable MLC storage in PCM. First, various new programming schemes are described that achieve MLC with low power, low number of iterations and low latency. Notably, a new programming algorithm operating entirely in the analog domain is presented that promises large savings in latency and power, in addition to area savings in silicon implementation. Second, techniques to cope with drift are introduced. These include two new metrics of the cell-state, which are shown to be significantly robust to drift. In addition, a new drift-tolerant modulation coding scheme is introduced which offers reliable retention of MLC data over extended time periods. Experimental results on PCM test cell-arrays demonstrate the effectiveness of the proposed methods in offering high resilience to drift. Most notably, 4 levels/cell storage with raw bit-error-rates in the order of 5×10^{-5} is achieved in a 200 kcell array for over 150 days after programming at room temperature.

I. Introduction

Phase-change memory (PCM) is currently regarded as the most promising new nonvolatile solid-state memory technology for applications that DRAM and NAND-Flash, today's incumbent memories, cannot address. Phase-change memory (PCM) possesses certain features, such as high cycling endurance, low read/write latency and excellent scalability potential, that make it an interesting candidate for enabling disruptive changes in future computing systems. Its high suitability stems from the potential ability of PCM to act as both storage (non-volatile, cheap, high capacity) as well as memory (fast, durable, bit-alterable), because of its universal characteristics [1].

Multilevel cell (MLC) storage is essential for reducing the cost-per-bit of PCM technology and thus for increasing its potential for market acceptance. MLC storage is enabled by iterative programming algorithms, where a sequence of write-and-verify steps is employed to overcome the variability inherent in large memory cell arrays due to process and material variations.

In order for PCM to become a viable technology for high-volume manufacturing, a number of critical issues need to be addressed. Specifically, the reliability of the technology has to be brought to levels similar to those of existing technologies. Experimental results and simulations suggest that resistance drift is one of the most important reliability concerns in PCM. Resistance drift is a phenomenon according to which the resistivity of the amorphous phase of phase-change materials shifts upwards in time in a stochastic manner. Drift has been attributed to structural relaxation and/or stress release in the amorphous matrix [2], [3], and is particularly detrimental in MLC storage, because random fluctuations of the programmed resistance of closely-spaced levels may cause them to overlap and thus lead to decoding errors.

In this paper we focus on technologies that enable reliable MLC storage in PCM. First, we present a collection of novel iterative programming techniques that offer potential for low latency, low power and robustness to cell variability. Next, a PCM chip that implements some of the proposed iterative programming algorithms and achieves 2 bits/cell storage is described. We then address the issue of resistance drift in PCM devices. We introduce new metrics of the state of an MLC PCM cell that are more accurate estimates of the programmed quantity, i.e., the amorphous phase thickness, than the conventional metric of the low-field electrical resistance. These metrics are shown analytically and experimentally to be significantly more robust to drift than the low-field resistance. Finally, we present a new drift-tolerant modulation coding scheme for MLC PCM and demonstrate reliable storage of 4 levels/cell data for over 150 days at room temperature, with raw bit error rates in the order of 5×10^{-5}.

II. Programming algorithms for MLC PCM

In multilevel storage, the size of the amorphous region in the crystalline phase-change material is altered in a continuous manner to achieve different resistance levels. It changes as a function of the amplitude of the write pulse, as quantified by the cell's programming curve. However, process and material variations give rise ro resistance levels with broad distributions when single programming pulses are applied. A common solution is to employ iterative programming schemes, in which a sequence of write-and-verify steps is used in a feedback loop to minimize the error between the programmed and a specified target

G.F. Close is now with Melexis Technologies SA, Chemin de Buchaux, 38 - 2022 Bevaix, Switzerland resistance level. The iterative programming operation needs to be efficient in order to have practical significance. For example, low-latency programming is a requirement in memory-type applications, i.e., DRAM replacement or caching. Furthermore, high endurance is an enabling feature in both memory-caching as well as enterprise-storage applications. Finally, embedded applications dictate low overall energy consumption. Careful design of the programming algorithm is thus important in achieving a good balance of attributes depending on the intended application.

Conventionally, MLC programming starts with a RESET pulse followed by a series of pulses of increasing current in the partial-SET regime (down-going slope of the programming curve) [4]. It should be noted that by partial-SET pulses the resistance of the cell may only be decreased; in order to increase the resistance new amorphous volume needs to be created, thus melting has to be invoked. Alternatively, programming may start with a SET pulse and then melting pulses of varying amplitude in the partial-RESET regime (up-going slope of the programming curve) are used to increase or decrease the resistance [5].

A novel iterative programming scheme that uses both partial-SET and partial-RESET pulses is depicted in Fig. 1(a) [6]. Operation starts from the partial-SET regime and either terminates there if the target resistance is reached, or switches to the partial-RESET regime if the programmed resistance drops below the target level. The method combines the low energy dissipation of the partial-SET regime with the bi-directional flexibility of the partial-RESET regime. Additionally, a favorable tradeoff between programming latency (fast convergence in the left-slope) and robustness to variability (use of both programming regimes) is achieved. This method has been demonstrated experimentally to achieve tight distributions of programmed resistances. The hybrid scheme performs favorably in comparison to the conventional schemes in terms of robustness to cell variability and average power dissipation as shown in Fig. 1(b).

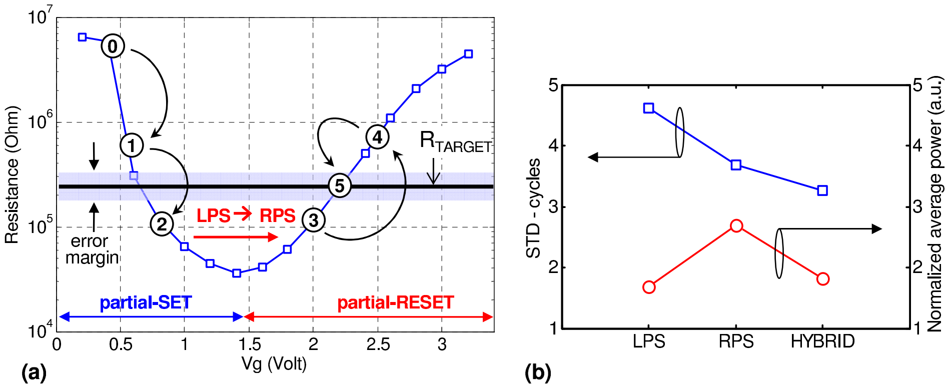


Fig. 1. (a) Illustration of hybrid programming scheme; (b) Comparison of iterative programming schemes in terms of standard deviation in the average number of algorithm iterations, indicating robustness to variability, and normalized power dissipation. LPS, RPS and HYBRID denote the scheme operating on the left slope, the right slope, and both slopes of the programming curve, respectively.

Given a three-terminal access device, such as a FET, the programming current can be controlled by the word-line (WL) and/or the bit-line (BL) voltage. Fig. 2(a) shows a three-dimensional programming surface for a PCM cell obtained by biasing the FET access device at different gate voltages while varying the drain voltage. By proper selection of a small set of biasing conditions, e.g., gate voltages for a FET, a new multi-trajectory and multi-level current-control algorithm can be realized for MLC programming (Fig. 2(b)) [6]. Resistance levels are assigned to different trajectories for programming. Characteristics of this method are an adjustable programming current/voltage window per target level, which affects the resolution of the iterative algorithm, and a faster speed of convergence due to the more efficient current control compared to conventional schemes.

In PCM cells, it is mainly the volume and topology of the amorphous phase that determines the state of the cell duing MLC programming [7]. Analysis of experimental I-V measurements for different cell states has shown [8], [9] that the effective amorphous thickness in the active element depends on the type of the programming pulse, i.e., partial-SET or partial-RESET pulse (Fig. 3(a)). A two-dimensional (2-D) programming scheme in which the cell state is defined not only by the resistance value, but also by the type of the programming pulse employed has been devised [6]. Experimental results in Fig. 3(b) show the ability of the 2-D scheme to store 8 levels in each of two programming regimes (partial-SET or partial-RESET) and to retrieve 16 levels of information using an appropriate read-out scheme, thus doubling the PCM cell storage capacity.

The programming schemes described above rely on data converters and digital logic blocks for implementation. This leads to significant latency and power consumption and entails considerable area penalty. An analog iterative programming scheme has been conceived that addresses these issues [6]. The essential idea is to implement the measurement of the current error and the subsequent corrections to the amplitude of the write pulse in the analog domain. The programming waveform applied to the cell is a repetition of a pulse form consisting of a low-amplitude part that demarcates the read phase and a high-amplitude part used for programming (Fig. 4). During the read phase the error with respect to the target current is calculated and then integrated to produce the corrected output that will be applied during the subsequent write phase. A proof-of-concept experiment was

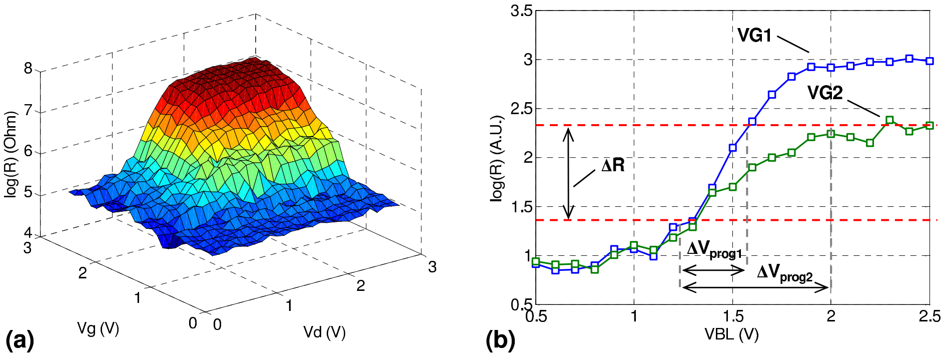


Fig. 2. Multi-trajectory current control: (a) 3-D PCM cell programming surface, and (b) programming curves with Vd (VBL) as a variable and Vg as a parameter showing the concept of multiple trajectories.

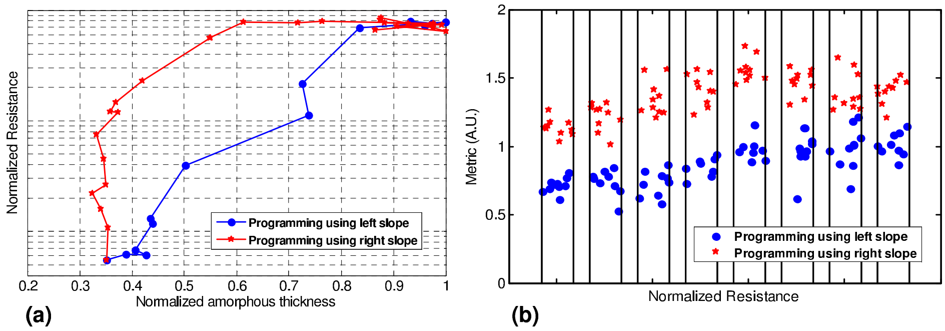


Fig. 3. Two-dimensional programming: (a) normalized cell resistance as a function of the equivalent amorphous volume in the left and right slopes, and (b) illustration of 16-levels storage.

performed using discrete electronic components to achieve 16 distinct resistance levels within a narrow resistance margin of 1.3 orders of magnitude (Fig. 5). The analog iterative programming scheme has multiple advantages. First, significant savings in latency can be realized as there is no need to exercise data converters during iterative programming. Secondly, an elaborate digital logic block implementing the controller is not needed. Last but not least, significant savings in the area allocated for the programming circuitry can be realized. This could lead to large parallelism in the design of PCM chips and thus high write bandwidths, as required in most practical applications.

III. MLC PCM CHIPS

A PCM chip has been fabricated that, among others, implements the hybrid iterative MLC programming scheme of the previous section [10]. The 256 Mcells chip was fabricated in a 90nm CMOS process with 6 levels of Cu interconnect, occupying 15×8 mm^2. It is organized in an 8x8 array of 4-Mcell tiles. Two rows of 8 tiles each are selected in parallel during readout, with one cell per tile. The chip block diagram is shown in Fig. 6, while Figs. 7(a-c) show the chip micrograph along with TEM and SEM images of the PCM mushroom cells fabricated with a sub-lithographic keyhole process described in [11].

For reading, the sensed BL current, after settling, is first mirrored with a self-adjustable gain. This feature extends the range of cell currents that can be captured by the following 6b ADC. The current, after reference subtraction, is integrated by a capacitor and then digitized by a single-ended cyclic 1.5b/stage ADC. The readout voltage, reference current and integration time can all be configured on the fly, enabling adjustment of the ADC's dynamic range to the resistance range of the PCM cells. For programming, two 8b DACs generate the programming pulses. The ADC code is fed to an iterative fixpoint PID controller which calculates the next programming pulse value according to the hybrid write-and-verify scheme of section II. The PID gains and pulse durations are adjustable, and multiple modes of cell programming are supported.

MLC programming at 2 bits/cell has been accomplished using the MLC PCM chip; in particular, convergence of the programming algorithm is achieved in less than 13 iterations for 99.9% of the cells, resulting in write access time of 9.8us [10]. The readout access time measured at the chip interface was 320ns, with the readout rate reaching 230Mb/sec in pipeline mode.

In addition to the MLC chip above, a smaller prototype chip including a 2×2 Mcell array, as well as addressing, readout, and write driver circuitry has been fabricated for grater flexibility in developing drift mitigation schemes [12]. For programming, a voltage generated off-chip is converted on-chip into a programming current. This current is then mirrored into the selected BL for the desired duration of the programming pulse. The memory chip was fabricated in a 90nm CMOS process and occupies 2.7×2.2 mm^2. The prototype PCM chip is controlled by a FPGA that implements the read and write algorithms for MLC operation and controls the off-chip peripherals, including power supplies, reference voltages, and external DACs. The FPGA also implements a data acquisition module for rapid collection of measurements and diagnostics.

IV. Dealing with the problem of drift

Resistance drift is a characteristic phenomenon observed in phase-change memory cells, which manifests itself as a steady increase in the electrical resistance of the stored cell-state over time [3]. Drift adversely affects the reliability of MLC storage in PCM, because the distance between adjacent levels is small and stochastic fluctuations of the resistance are more likely to cause level overlap over time than in binary storage. As it has been shown in Section II, efficient MLC schemes are able to produce tight and dense resistance distributions. However, these distributions are shifted and broaden significantly almost immediately after programming due to noise and drift.

In order to investigate the evolution of the programmed resistance levels due to drift, a 200 kcell sub-unit of our prototype PCM array is programmed at 4 levels-per-cell. The two corner levels are programmed with “single-shot” RESET and SET pulses respectively, whereas the two intermediate levels are programmed using iterative write-and-verify steps. Each programming pulse is a box-type rectangular pulse. The RESET pulse is a programming pulse of high current, while for SET a trapezoidal pulse of long trailing edge is used to allow sufficient crystallization. Table I lists the definition of the two intermediate levels, in read current, along with convergence statistics. The small number of iterations and high rate of convergence attest to the effectiveness of the iterative write-and-verify algorithm to achieve multilevel programming.

TABLE I
Level allocation for MLC programming

level	I_{TARGET}	I_{MARGIN}	av. no. iterations	convergence rate
0	I_{min}	n/a	n/a	n/a
1	8.3 uA	0.6 uA	2.7	98.2%
2	12.8 uA	0.6 uA	5.6	99.5%
3	I_{max}	n/a	n/a	n/a

To assess drift of the programmed levels, the programmed 200 kcell array was monitored at different time instances. Fig. 8 shows cumulative histograms of the programmed resistance distributions over time. One observation from these plots is that drift affects all stored levels starting at very short time scales, i.e., µs after programming. The spread of each resistance distribution is due to read noise, cell variability and non-uniform drift dynamics across cells. From the measured data of Fig. 8 one can estimate the average drift behavior as a function of the programmed resistance. Using the drift power-law model, i.e., $R(t)/R(t_0) = (t/t_0)^v$ [2], one can fit the evolution of the programmed resistance levels over time and extract the average drift exponent v for each resistance level (Fig. 9). Here, t_0 is set to 40 µs after programming. From Fig. 9 it can be observed that the drift power law model appears to hold for over 9 orders of magnitude in time, over which measurements were made.

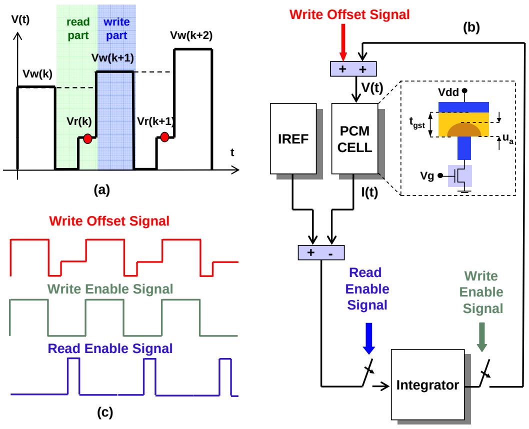


Fig. 4. Analog iterative programming concept: (a) pulse waveform comprising read and write phases, (b) schematic illustrating a possible implementation of the scheme, (c) waveforms used in (b).

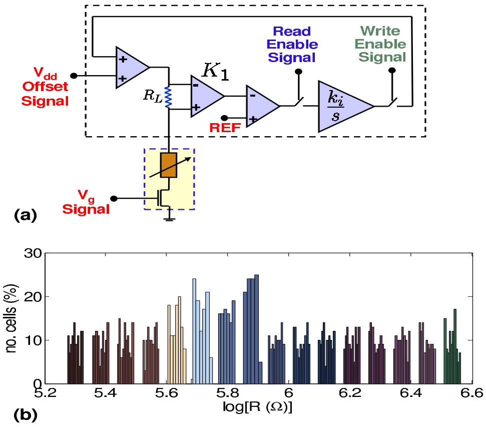
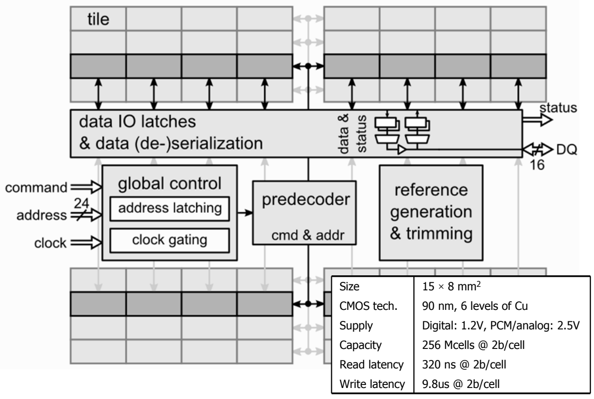


Fig. 5. Analog iterative programming experiment: (a) schematic showing an implementation using discrete components, and (b) histogram of programmed cell resistances at algorithm termination.



Size	15 × 8 mm^2
CMOS tech.	90 nm, 6 levels of Cu
Supply	Digital: 1.2V, PCM/analog: 2.5V
Capacity	256 Mcells @ 2b/cell
Read latency	320 ns @ 2b/cell
Write latency	9.8us @ 2b/cell

Fig. 6. MLC PCM chip block diagram.

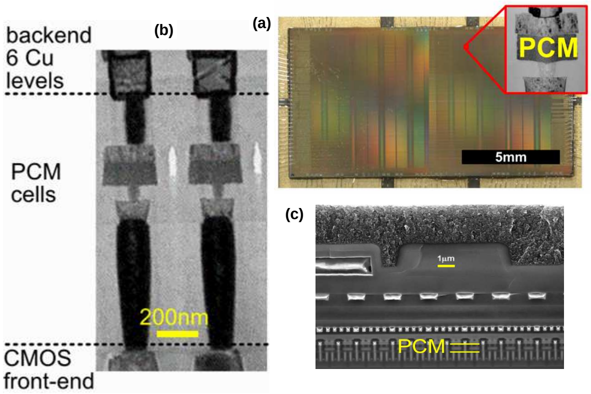


Fig. 7. 256 Mcell MLC PCM chip: (a) chip picture, (b) SEM cut across WL direction, (c) TEM cut across BL direction.

In the following we describe a number of technological solutions to the problem of drift. First, we present a set of novel metrics of the cell state in MLC PCM that are shown to be largely resilient to drift, in sharp contrast to the cell resistance. Second, we describe a novel modulation coding technique that is highly tolerant to drift.

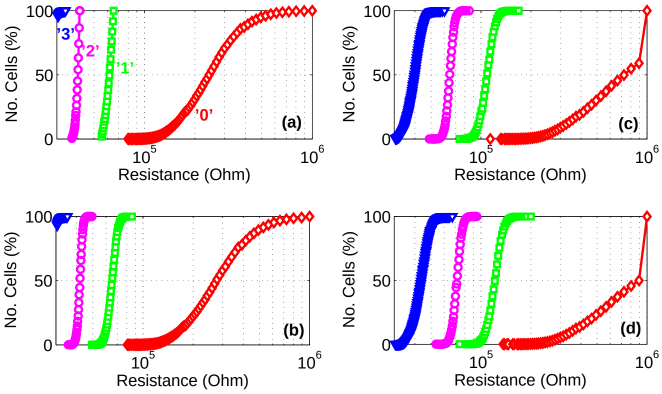


Fig. 8. Time evolution of programmed resistance distributions of 200 kcells due to drift: (a) as programmed, and (b) 40 μs, (c) 1000 s, (d) 46,000 s after programming.

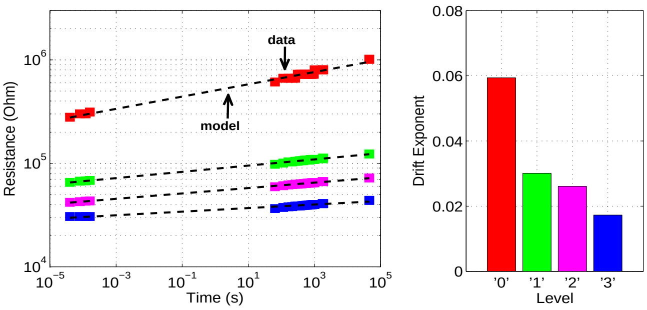


Fig. 9. Evolution of mean programmed resistance values over time and extracted mean drift exponent according to resistance-drift power-law model.

A. *Drift-resilient cell-state metrics*

By varying the amplitude of the programming pulse, the size of the amorphous region and its thickness (u_A) can be increased or decreased in a continuous manner, this being the basis for multi-level storage in PCM. Conventionally, the electrical resistance of the cell is used to measure u_A. To measure the resistance, the PCM cell is typically biased with a constant read voltage (V_R), which is low enough to avoid threshold switching for all cell states. The low-field electrical resistance, however, is subject to drift, posing significant challenge to the realization of MLC PCM.

To gain insight into the resistance metric we computed analytically the I-V characteristic of the mushroom cell, starting from the modified Poole-Frenkel transport model proposed in [13]. It was found that the computed I-V characteristic is well approximated by the I-V behavior corresponding to a PCM cell with cylindrical geometry given by [14]

$$I = 2q\frac{\pi r_{Eeff}^2}{\tau_0} N_T \Delta_z e^{\frac{-(E_c - E_f)}{kT}} \sinh\left[\frac{q\Delta_z V}{2kTu_{Aeff}}\right] \tag{1}$$

where r_{Eeff} is an effective bottom electrode radius and u_{Aeff} is an effective amorphous thickness. This is also substantiated by experimental studies presented in [8], [9]. From Eq. (1), the low field electrical resistance (for small V) is given by

$$R = \frac{kTu_{Aeff}\tau_0 e^{\frac{(E_c - E_f)}{kT}}}{q^2 \pi r_{Eeff}^2 N_T} \Delta_z^2 \tag{2}$$

As seen in Eq. (2), one detrimental drawback of the resistance metric is that it is a strong function of the activation energy, $E_a = E_c - E_f$. The temporal drift behavior observed in the resistance metric is attributed to variations in the activation energy with time [2].

This clearly demonstrates the need for a metric that is a strong function of the effective amorphous thickness, yet less dependent on the activation energy. Such a metric may be constructed by exploring the high field regime for every programmed state. To achieve this, the read voltage is progressively increased until a certain pre-defined current level, denoted by I_R, is reached. This current level is chosen to be a safe value much below the current needed for threshold switching to occur. The time needed to reach I_R can be considered as a measure of the programmed state [14]. Using Eq. (1) and considering the scenario in which the read voltage is varied as a linear function of time, i.e. $V_R = k_{slope}t$, the new metric, M, is expressed by

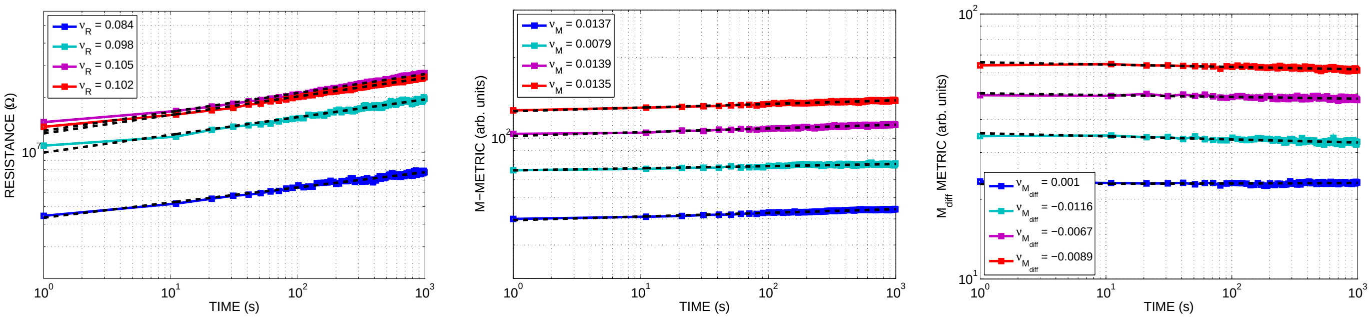
$$M(I_R) = \frac{2kTu_{Aeff}}{q\Delta_z k_{slope}} \sinh^{-1}\left[\frac{I_R \tau_0 e^{\frac{(E_c - E_f)}{kT}}}{2q\pi r^2_{Eeff} N_T \Delta_z}\right] \tag{3}$$

It can be seen that M is proportional to the effective amorphous thickness, yet it is a weak function of the activation energy (nearly linear as opposed to exponential), suggestive of a significant tolerance to drift. To further reduce the dependence of the M metric on E_a, one could detect the time it takes for the current to increase from a current level $I_{R0} < I_R$ to the current level I_R [14]. This differential variant of M is likely to be even more tolerant to drift. This becomes evident by considering an approximation of Eq. (3), finally arriving at the expression

$$M_{diff} = M(I_R) - M(I_{R0}) = \frac{2kTu_{Aeff}}{q\Delta_z k_{slope}} \log\left(\frac{I_R}{I_{R0}}\right) \tag{4}$$

This variant of the M metric is independent of the activation energy while preserving the linear relationship with the effective amorphous thickness. However, note that I_R and I_{R0} have to be sufficiently high for the approximation in Eq. (4) to be valid.

Experimental results are presented next that illustrate the drift tolerance of the new metrics compared with the low-field resistance metric. A group of thirty PCM cells were programmed to four different states. The resistance metric (obtained with a V_R equal to 0.2 V) and the M metric were measured for these cells over a period of 1000 s. The resulting average drift behavior over the thirty cells is shown in Fig. 10(a,b). Impressively, there is almost one order of magnitude improvement in the drift coefficient of the M metric, denoted by v_M. Also shown in Fig. 10(c) is the drift behavior of the M_{diff} variant with $I_{R0} = 0.1\mu A$ and $I_R = 1\mu A$. The drift coefficient associated with M_{diff} is negligible for all levels. These results clearly demonstrate the ability of the new metric and its variants to address one of the most important challenges for the realization of multi-level PCM.



(a)

(b)

(c)

Fig. 10. Drift performance of (a) the low-field resistance, (b) the metric M, and (c) the metric M_{diff}.

B. Drift-tolerant coding

In this section, a new modulation coding method to enhance the reliability of PCM devices in the presence of drift is presented [15]. The main idea of the proposed coding technique lies in the fact that, in the majority of cases, the relative order of cells programmed in different resistance levels does not change due to drift. To exploit this basic drift characteristic, information is encoded in the relative order of cell resistance levels within a small group of cells that forms the codeword. In most cases drift does not change this ordering, thus the codeword can be correctly decoded.

To assess the performance of the proposed drift-tolerant coding scheme, data were encoded and then stored in a sub-array of 200 kcells of the PCM prototype chip described above. The codeword length was 7, and the rate of the particular code used was 1.57 bits/cell. Other codes can be constructed that have higher rate, at the expense of encoder/decoder complexity or performance. Two detection methods were considered. The first method treats the stored data as uncoded, and applies appropriate level thresholds to detect the stored levels. The thresholds are adaptive, and are calculated at each time instant using a collection of reference cells [16]. For fairness of comparison, the number of cells that were used as reference cells corresponds to the same amount of capacity loss, i.e., redundancy, as the proposed code. The second method decodes the particular code applied to the cell array. The performance of the two methods is illustrated in Fig. 11, where 'REF-CELL' and 'CODE' denote the first and the second detection method, respectively. Although the two decoding methods have the same overhead, drift-tolerant coding is superior by more than one order of magnitude in error-rate.

It is quite impressive that drift-tolerant coding exhibits a raw error rate around 5×10^{-5} even after 157 days at room temperature, a result that cannot be matched by a reference-cell-based scheme. With such error rates, simple, low-redundancy error-correction codes would be sufficient to bring the overall error rate down to levels around 10^{-15} or less, which are required for practical memory devices.

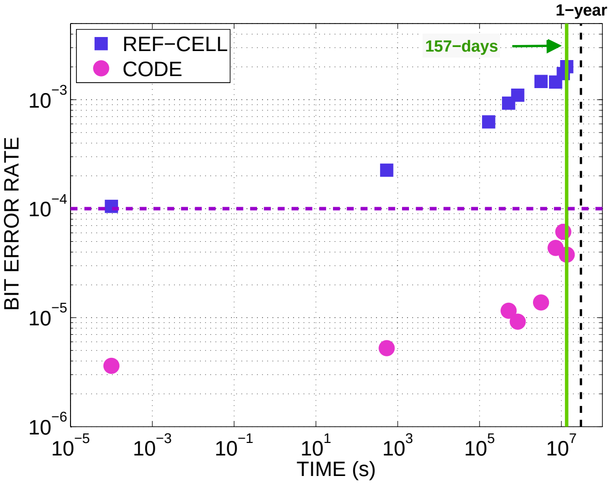


Fig. 11. Bit-error-rate of "reference-cell" and the proposed coding methods on a PCM array of 200 kcells. Measurements have been performed starting at 40 μs after programming and for extended periods of time.

V. Conclusions

Multilevel storage is essential for the competitiveness of PCM technology. A family of novel iterative programming techniques has been presented, which achieve multiple tight resistance distributions in a small number of iterations and are robust to cell variability. In particular an analog programming scheme promises significant savings in latency, power and area over its digital counterparts. Despite achieving tight resistance distributions at the end of the programming operation, noise and resistance drift quickly lead to read errors due to level shifting and distribution broadening. We have presented two strategies for dealing with the problem of drift. First, we have shown that a collection of novel metrics of the cell state in MLC PCM are largely robust to drift, in contrast to the resistance metric. Finally, a new modulation coding technique was presented that offers enhanced reliability for MLC PCM over extended periods of time.

Acknowledgments

We acknowledge the support of the entire PCM team at IBM, in particular the PCM team at the IBM T.J. Watson Research Center. We especially thank U. Egger at the IBM Zurich Research Laboratory and Prof. Th. Antonakopoulos and his group at the University of Patras, Greece, for their contributions in building the characterization setup used in parts of this work.

References

[1] R. F. Freitas and W. W. Wilcke, "Storage-class memory: The next storage system technology," *IBM J. Res. Dev.*, vol. 52, no. 4/5, pp. 439–447, 2008.

[2] 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.

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

[4] F. Bedeschi *et al.*, "A bipolar-selected phase change memory featuring multi-level cell storage," *IEEE J. Solid-State Circ.*, vol. 44, no. 1, pp. 217–227, 2009.

[5] I. Karpov and S. Kostylev, "SET to RESET programming in phase change memories," *IEEE Elect. Dev. Lett.*, vol. 27, pp. 808–810, Oct. 2006.

[6] N. Papandreou, H. Pozidis, A. Pantazi, A. Sebastian, M. Breitwisch, C. Lam and E. Eleftheriou, "Programming Algorithms for Multilevel Phase-Change Memory," in *Proc. ISCAS*, 2011, pp. 329–332.

[7] A. Itri and D. Ielmini and A. Lacaita and A. Pirovano and F. Pellizzer, and R. Bez, "Analysis of phase-transformation dynamics and estimation of amorphous-chalcogenide fraction in phase-change memories," in *Proc. IEEE IRPS*, 2004, pp. 209–215.

[8] A. Pantazi, A. Sebastian, N. Papandreou, M. Breitwisch, C. Lam, H. Pozidis, and E. Eleftheriou, “Multilevel Phase Change Memory Modeling and Experimental Characterization,” in *Proc. EPCOS*, 2009, pp. 34–41.

[9] N. Papandreou, A. Pantazi, A. Sebastian, E. Eleftheriou, M. Breitwisch, C. Lam, and H. Pozidis, “Estimation of amorphous fraction in multilevel phase change memory cells,” *Solid-State Electronics*, vol. 54, no. 9, pp. 991–996, 2010.

[10] G. F. Close, U. Frey, J. Morrish, R. Jordan, S. Lewis, T. Maffitt, M. Breitwisch, C. Hagleitner, C. Lam and E. Eleftheriou, “A 512Mb Phase-Change Memory (PCM) in 90nm CMOS achieving 2b/cell,” in *Proc. Symposium VLSI Circuits*, 2011.

[11] M. Breitwisch *et al.*, “Novel lithography-independent pore phase change memory,” in *Proc. Symp. VLSI Tech.*, 2007, pp. 100–101.

[12] G. F. Close, U. Frey, M. Breitwisch, H. L. Lung, C. Lam, C. Hagleitner, and E. Eleftheriou, “Device, circuit and system-level analysis of noise in multi-bit phase-change memory,” in *Proc. IEDM Tech. Dig.*, 2010, pp. 660–663.

[13] D. Ielmini and Y. Zhang, “Analytical model for subthreshold conduction and threshold switching in chalcogenide-based memory devices,” *J. Appl. Phys.*, vol. 102, no. 5, p. 054517, 2007.

[14] A. Sebastian, N. Papandreou, A. Pantazi, H. Pozidis, and E. Eleftheriou, “Non-resistance-based cell-state metric for phase-change memory,” *Journal of Applied Physics*, June 2011 (submitted).

[15] N. Papandreou, H. Pozidis, T. Mittelholzer, G. Close, M. Breitwisch, C. Lam and E. Eleftheriou, “Drift-tolerant Multilevel Phase-Change Memory,” in *Proc. IMW*, 2011, pp. 147–150.

[16] Y.N. Hwang, C.Y. Um, J.H. Lee, C.G. Wei, H.R. Oh , G.T. Jeong, H.S. Jeong , C.H. Kim , C.H. Chung, “MLC PRAM with SLC write-speed and robust read scheme,” in *Proc. Symposium VLSI Circuits*, 2010, pp. 201–202.