Circuit and System-Level Aspects of Phase Change Memory

Haralampos Pozidis, *Senior Member, IEEE,* Nikolaos Papandreou, *Senior Member, IEEE,* and Milos Stanisavljevic, *Member, IEEE*

***Abstract*—Phase Change Memory (PCM) is a new nonvolatile memory technology that promises to disrupt current big data applications and even create entirely new ones. This is because it can serve as both fast storage and large memory, which is out of reach for incumbent memory technologies. This tutorial focuses on recent technological developments and system-level concepts for the realization of PCM-based systems. We also discuss new circuits and architectures that enable novel applications of PCM in big data analytics and hardware for artificial intelligence.**

***Index Terms*—Phase change memory, memory hierarchy, persistent memory, big data analytics, in-memory computing.**

I. Introduction

BIG advances in solid-state memories are historically being made approximately every three decades and have profound effects on the entire computing industry when they happen. Starting from Dynamic Random Access Memory (DRAM) in the 50's, and flash memory in the 80's, we are now experiencing a new era in semiconductor memory evolution, the era of Phase Change Memory (PCM) and other Nonvolatile Memory (NVM) technologies.

DRAM has been the catalyst in creating the modern von Neumann machine, the ubiquitous computer. With its unprecedented speed for reading and writing bits, and its extreme durability, it became the perfect companion to the central processing unit for exchanging data at high speed. Flash memory appeared thirty years later to satisfy another need, namely nonvolatile storage of vast amounts of data in non-moving media. Flash enabled the mobile revolution in consumer electronics that changed our lives, but also disrupted the datacenter and enterprise computing space more recently.

Despite their proliferation, however, DRAM and flash sometimes fall short when it comes to modern, time-critical, data-hungry applications, either due to limited capacity (DRAM) or slower access time (flash). Notable examples are in-memory databases, Artificial Intelligence (AI) and large graph mining. Such applications can significantly benefit from larger memory or faster storage, breaking the bottleneck of swapping data back and forth between flash and DRAM main memory.

A form of PCM recently appeared at the market to fulfill that new need and to enable new applications. Finally, it appears that the holy grail of large memory and fast storage may be within reach [1]. If this is the case, then one may wonder what is preventing us from installing PCM in every server across all datacenters in the world. The answer is cost, as it always has been for memory technologies. Unless the cost per bit of PCM drops to a level comparable to that of flash, the former will likely not become mainstream.

Admittedly, there are two main ways to reduce the cost per bit in a memory technology. One is Multilevel-Cell (MLC) capability, i.e., storing multiple bits per memory cell, and the other is three-dimensional stacking of the memory cells. Both methodologies come with significant challenges in PCM, caused by the physics of phase change materials. These challenges typically limit the endurance, scalability or data retention of PCM. In Section II of this tutorial we discuss some of the key technology challenges associated with single-level-cell storage, multilevel-cell storage and vertical cell stacking in phase change memory. We review recent circuit-level and coding techniques that have been proposed to address these challenges.

In Section III of the tutorial we discuss various proposals for enabling PCM at the system level. Section IV discusses applications of PCM. A particularly promising emerging application is in-memory computing [2], which has the potential to disrupt the current von Neumann computing paradigm. We review the main characteristics of this application and discuss why PCM is well suited to build circuitry and systems to realize it, as well as provide insights into recent prototypes. We also review other key applications that are disrupted by PCM, namely, large in-memory databases, large-graph analytics and High-Performance Computing (HPC).

We specifically focus on circuitry that has been realized in prototypes to mitigate reliability issues of PCM. Furthermore, we give specific examples of subsystems and system-level demonstrators that have been built around prototype PCM chips to enable first-of-a-kind applications. Finally, we describe how these circuits and systems exploit various PCM properties and how they tackle different challenges imposed by PCM technology.

II. Technology Aspects

In this section we present an overview of technology aspects of PCM focusing on device reliability. Developments regarding PCM device physics, cell design and materials are well documented in various comprehensive studies in the literature [3]–[6]. Here we focus on the developments towards the realization of commercial products and adoption of PCM in memory systems. We therefore choose to review recent advancements in the areas of power consumption, endurance, MLC capability and density scaling.

H. Pozidis, N. Papandreou, and M. Stanisavljevic are with IBM Research Europe, Säumerstrasse 4, 8803 Rüschlikon, Switzerland, (e-mail: hap@zurich.ibm.com; npo@zurich.ibm.com; ysm@zurich.ibm.com).

A. RESET current

Reduction of the PCM RESET current is important for reducing the overall power consumption. The material compound typically used in PCM devices is $Ge_2Sb_2Te_5$ (GST) [7]. One approach to reduce the RESET current is the use of dopants, such as SiO_2- or C-, in an attempt to increase the efficiency of Joule heating in the device [8]–[10]. At the same time, however, the use of dopants typically results in a degradation of the SET performance, which is attributed to an increase in the resistivity of the SET state [9], [10]. Other ways to reduce the RESET current are by scaling the memory cell to smaller dimensions, or by confining the metal heater [11]–[13]. Lowering the RESET current is also beneficial for reducing the thermal disturb effects in the array [14]. Due to thermal disturb, the neighbors (victim cells) of an aggressor cell that is being programmed to the RESET state may be unintentionally affected. This type of disturb has a direct impact on the device reliability and can cause an increase of the Raw Bit-Error Rate (RBER) in the array. Obviously, thermal disturb becomes more critical with scaling, therefore, in addition to the reduction of the RESET current, advances in materials and dielectrics also play an important role [15], [16].

B. Endurance

Reducing the RESET current is not only desirable from a power efficiency aspect, but it also affects favorably the cycling properties of the device. High endurance is a key factor for the adoption of PCM in hybrid memory architectures. Proper engineering of the bottom electrode has been successfully demonstrated to provide significant reduction of programming power, enabling endurance of more than 10^9 cycles [17]. Moreover, novel (Sb-rich) confined PCM cell designs with a thin metallic liner have been shown to significantly improve the endurance by preventing void formation during the write process, thus delivering endurance of more than 10^{12} cycles [18]. Material engineering can also play a key role in enhancing the endurance, in addition to improving other device properties such as the SET speed and the retention at high temperatures [19]. The latter is a critical property for the adoption of PCM in IoT and automotive applications. Typically, materials that offer higher crystallization temperatures result in a better thermal stability and therefore longer data retention [7]. In [20], a C-doped GST material was introduced to improve the thermal stability and cycling endurance required for embedded applications.

C. Multilevel-cell capability

Thanks to its inherent property of multilevel resistance programmability, PCM is a perfect candidate for realizing MLC storage [21]. This key property is exploited in memory and storage applications to increase the capacity and reduce the cost per bit, and for in-memory computing applications to realize storage of analog weights and computations [22], [23]. Accurate and fast resistance level programming is desirable for two reasons, first in order to achieve precise and tight level distributions, and second in order to reduce the overall program time of the iterative write-and-verify scheme [24].

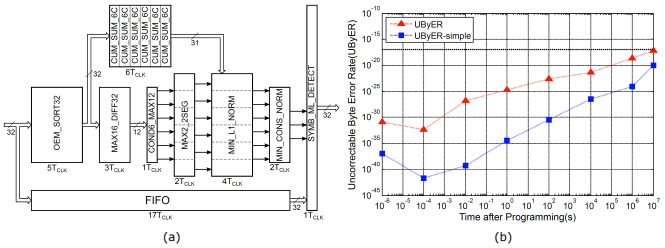

Fig. 1. (a) Block diagram of DID algorithm. The algorithm estimates variable read thresholds based on ordered statistics and clustering of the soft read signals providing reliable level detection for MLC PCM. (b) Using an outer Reed-Solomon ECC, the combined performance translates to uncorrectable byte-error rate of 10^{-17} after 4 months of retention. Adapted from [37].

An important challenge towards reliable MLC PCM is the variations of the resistance levels with time (typically referred to as drift) and temperature, both inherent to the PCM material properties [25], [26]. Such resistance variations have implications for both storage and computing applications as they can affect the RBER of the media and the accuracy of deep neural network inference, respectively, [27], [28]. In particular, the variability of drift, accross cells and resistance values at the array level, can significantly affect the reliability of the device and effectively pose a limitation to the number of levels that can be stored in a memory cell. A number of methods have been proposed to combat the resistance variations starting from the device- and circuit-level, with efficient cell designs [29]–[31] and read sensing schemes [32]–[34], and extending to the chip- and system-level with novel signal processing and coding techniques [35]–[37].

In particular for memory applications, where the access granularity is small, the task of fast level detection from a limited number of cells is challenging. Fig. 1 shows a novel high-speed Drift-Invariant Detection (DID) algorithm that enables reliable multilevel detection, presented in [37]. The DID algorithm performs adaptive level detection using a block of 32 cells for MLC (2 bits-per-cell) PCM and is agnostic to the drift dynamics. The algorithm can track and adapt to the changes of the resistance levels due to temperature variations as well. A key requirement for extended memory applications (where PCM co-exists with DRAM) is to keep the signal processing and other media management functions at low latency. The DID circuit design exhibits a latency of 90ns, which makes it suitable for high throughput on-chip implementations. By employing several of the aforementioned techniques, a demonstration of reliable TLC (3 bits-per-cell) PCM was presented for the first time in [38].

D. Scaling

Cost effectiveness is a key requirement for the widespread adoption of PCM in memory and storage applications. Scalability of PCM has been proven both in single devices at sizes in the order of several nm [39]–[42], and in prototype and commercial chips [13], [43], e.g., a 20nm 8Gb PCM chip was presented in [44]. Cross-point PCM is a recent solution that can further improve the cost effectiveness as it not only realizes the ideal $4F^2$ cell area density, but also

enables 3-D stackable structures [45], [46]. A key design point for cross-point PCM is the efficiency of the selector in terms of ON/OFF current ratio and sneak current, as well as threshold-voltage adjustability and hold characteristics [6], [47]. Avoiding large overshoot currents during read operations is critical for the reliability in the array level [6]. This is a type of read disturb that is different from pulse-induced crystallization effects caused by repeated read pulses, which also deteriorates the array reliability [48].

III. System-Level Aspects

In the past decade or so a large collection of papers presented various approaches to address the main reliability issues of PCM at the system level. They can be broadly classified in methods to improve performance [49], [50], correct soft/hard errors [51]–[53], address endurance limitations through write reduction [49], [50], [54]–[57], or wear leveling [56]–[59], or to effectively use MLC capability [60].

A. Performance Improvement

The authors of [49] propose a narrow buffer organization for PCM devices to minimize NVM writes, and multiple buffer rows that exploit data locality to coalesce writes to mitigate PCM's long write latency. A hybrid memory organization is proposed in [50] consisting of a large PCM memory and a smaller DRAM acting as page cache for the PCM. The DRAM cache helps performance by storing frequently accessed pages, while PCM holds the pages that are less often accessed.

B. Error Correction

A few papers have addressed the issue of stuck-at faults in PCM, that is, the fact that PCM cells tend to fail after repeated write cycling and this failure is permanent, leaving the cell in a state that cannot be altered. In [51] the concept of Error Correcting Pointers (ECP) is proposed, which encodes and stores the addresses of failed cells and allocates additional cells to replace them in a memory block. In contrast to conventional ECC, in which redundant cells need to be re-written every time the block data is updated, ECP does not need extra writes except when a new cell failure occurs.

An alternative method for tackling permanent cell failures in PCM is proposed in [52]. The Stuck-At Fault Error Recovery (SAFER) technique continues to use stuck-at cells to store data, as their value is still readable. SAFER dynamically partitions data blocks so that each partition has at most one failed bit, and thus the data can be recovered by single bit error correction techniques.

One weakness of both the ECP and SAFER approaches is that they are not designed to handle soft errors, which are known to hamper PCM operation. The Fine-grained Remapping with ECC and Embedded-Pointers (FREE-p) method of [53] addresses both hard and soft errors in PCM, as well as device failures. Fine-gained Remapping (FR) utilizes the still-functional cells of worn-out memory blocks to store remapping information. FR is also integrated with ECC to detect and correct both permanent and soft errors. The FREE-p method can also be augmented to support correction of device failures.

C. Endurance Boosting

A number of system-level techniques have been proposed to increase the endurance of PCM to better serve hybrid-memory applications. They can be broadly categorized in write-reduction methods and wear-leveling methods.

In the simplest form of write reduction, the Data Comparison Write (DCW) scheme of [54] employs a read-before-write operation to first determine whether new bits differ from previously stored bits in the same cells. A new bit is only written if it differs from the prior bit in the same cell. In [49] the authors propose partial writes, i.e., tracking data modifications and only writing modified cache lines or words to the PCM. The paper of [55] proposes Flip-N-Write (FNW), which also relies on a read-modify-write operation. On a write operation FNW writes either the new data word as is or the "flipped" value of it, depending on which results in less bit flips compared with the originally stored data word. FNW introduces an extra bit for each data word to indicate whether the data word had been flipped or not. In all cases, the extra read operation for every write is justified in terms of both endurance and performance because PCM writes consume much more energy and are much slower than PCM reads.

System-level methods to boost the endurance of PCM are proposed in [50] and [56]. The authors of [50] propose a hybrid memory, where a large PCM memory is augmented with a small DRAM that acts as a page cache for the PCM memory. The page cache helps endurance by reducing the number of writes to PCM with write combining and coalescing. At the cache line level, only the lines modified in a page are written to PCM. To avoid unbalanced damage from writes, cache lines are rotated on a page. In [56] a novel cache replacement policy is applied to reduce writebacks from DRAM to PCM also in a hybrid memory configuration. Read-write-read and page partitioning techniques are used to remove unnecessary writes and also detect potential write failures, similar to the DCW and partial write methods above.

In a later approach, [59] proposes a hybrid DRAM/PCM memory system design that is robust across a wide range of workloads. A memory controller is introduced that implements a page placement policy called Rank-based Page Placement (RaPP). The policy ranks pages according to access frequency and write intensity, migrating top-ranked pages to DRAM.

In addition to write reduction, the endurance of PCM can be improved at the device level by distributing writes across the entire cell array equally, a technique known as wear leveling. In [57] Row-Level rotation (RL) and Segment Swapping (SS) are introduced for wear leveling. RL equalizes wear at the row level by rotating cache lines, whereas SS swaps two segments, the one currently being written and the one that is least-frequently-written.

The authors of [58] introduce the Start-Gap (SG) scheme, where an algebraic mapping between logical and physical address is the key concept. SG is shown to be very effective in prolonging the lifetime of PCM, while incurring minimal storage overhead. Furthermore, SG can be regulated to limit the extra writes caused by wear leveling to less than 1% of total writes. Address space randomization is proposed to

reduce the likelihood that spatially-correlated, heavily-written memory lines limit the system lifetime.

D. Using MLC Capability

Somewhat complementary to the above architectural approaches is the topic of MLC functionality in PCM. Leveraging the capability of PCM cells to be programmed either in single-bit (SLC) or multi-bit per cell (MLC) mode, [60] introduces Morphable Memory System (MMS), an adaptive method that can dynamically partition the memory into high-density pages and low-latency pages, corresponding to pages programmed in MLC and SLC mode, respectively. The work exploits the fact that typical applications do not use all the available memory capacity and proposes a cost-effective runtime mechanism to determine the best partition between high-density region and low-latency region. It also provides an interface for the operating system to handle dynamically varying memory capacity.

This mechanism is very similar to hybrid SLC/MLC controllers in flash memory systems [61] that have received increased attention recently. In fact, most modern enterprise-grade Solid State Drives and all-flash-arrays employ such hybrid controllers today to jointly offer high performance and endurance (provided by SLC flash) and high storage capacity (offered by TLC or QLC flash, depending on product).

IV. Applications

In this section we present an overview of promising potential PCM applications. The first part reviews more conventional memory applications, whereas in the second part we discuss novel applications in hardware for AI.

A. Memory Applications

The Intel Optane DC Persistent Memory (OPMM) is the first commercially-available NVM with higher density and lower cost than DRAM. It is available in memory DIMM form factor with up to 512GB capacity, 8x larger than the densest DRAM-based DIMM available today. This allows the design of affordable systems with up to 6TB of randomly accessible memory in a single server.

An expanding number of applications from HPC to data analytics, databases and cloud computing demand higher memory capacity to respond to the needs of workloads with increasing data sets. These applications, at least in principle, stand to benefit from the availability of large-capacity memory. A number of studies emerged recently studying the applicability of OPMM in various big-data workloads [62]–[65].

In [62] the authors compare the performance of shared-memory graph analytics frameworks on OPMM with a state-of-the-art distributed graph analytics system. They show that the same graph algorithms running on a 48-core server with 6TB of OPMM are competitive in completion time when running in a large cluster of 256 machines with a total of 12,288 cores and 49TB of DRAM. Similar findings are reported in [64], where a hybrid memory system with a large OPMM and small DRAM as cache is evaluated for large graph applications. Interestingly, [64] finds that using OPMM on a single socket may be more efficient than accessing DRAM across two sockets in a server.

The work of [63] evaluates OPMM on HPC applications, mainly stencil codes and matrix operations. They use OPMM as an address space extender for the main memory in HPC systems. They find that using OPMM alone hampers the performance of memory-bound HPC applications due to higher access latencies and lower memory bandwidth. However, using DRAM as a cache for the OPMM maintains the performance of HPC applications observed on DRAM-only memory systems, while also increasing the memory capacity of the system.

Large in-memory database systems are also set to benefit significantly from large NVM. In [65] the performance of OPMM is evaluated on analytical database workloads. It is shown that using a hybrid system with DRAM and OPMM can allow running much larger workloads with minimal degradation in performance compared with a purely DRAM-based, in-memory system when the query intermediates fit in DRAM.

Finally, a word is in order on the power consumption of OPMM. DRAM consumes static power even when not accessing data, because it needs to refresh its content periodically. As it was shown in a comprehensive study in [64], across different workloads and OPMM/DRAM configurations, OPMM significantly reduces the dynamic memory power compared to DRAM. It also achieves higher or similar power efficiency (in GB/s per W) compared to DRAM, except for write-only workloads. As OPMM has higher write power than DRAM, it is important to isolate writes from OPMM via a DRAM write buffer, as discussed in section III-C.

B. Applications in AI Hardware

One of the most exciting application areas for PCM technology is in realizing hardware for AI. In general AI computing hardware faces a severe efficiency problem since in data-centric computing most of the energy is consumed in transferring data to and from the memory instead of during computation [66]. In-memory computing (also known as computational memory) is a promising application where computational tasks are performed in the memory itself [67]–[71], exploiting key properties of the PCM technology such as binary storage capability, MLC capability and accumulative behavior arising from the crystallization dynamics [72]. We review recent in-memory applications such as logical operations, matrix-vector multiplication and computing with accumulative behavior that leverage the aforementioned properties of PCM. Additionally, we discuss the important concept of mixed-precision in-memory computing. Finally, we provide an overview of techniques and circuits as well as architectures used for in-memory computing.

1) In-memory Computing Applications: Typical basic logic operations like NOR have been demonstrated in memristive logic crossbar architectures [73]–[75]. These can be easily extended to PCM devices. A complete set of logic functions including NOR, NAND and NOT gates, each utilizing a single PCM device, has been demonstrated using the physics of crystallization [76] and melting [77]. By designing the read

circuitry for PCM to be able to compute the bitwise logic of two or more memory rows using custom sense amplifiers, bulk bitwise operations can be efficiently realized inside a memory chip [78].

Matrix-vector multiplication can be typically performed by mapping matrix weights linearly to the conductance values of PCM devices organized in a crossbar configuration and applying amplitudes or durations of read voltages to the crossbar along the rows. The result of the computation is proportional to the resulting current measured along the columns of the array [28], [69]. However, the precision is ultimately limited by the conductance variations arising from inherent PCM characteristics such as drift, 1/f noise and resistance changes due to ambient temperature variations [25], [79]. One of the most promising applications of in-memory matrix-vector multiplication is deep learning inference [23], [80]–[82]. The main limitations are variability in programming and ratio between SET and RESET conductance. Signal processing tasks such as compressed sensing and recovery, particularly in the context of image compression, could also utilize in-memory matrix-vector multiplications with PCM [28], [83].

In [84], [85] the authors perform the basic arithmetic operations of addition, multiplication, division and subtraction, with simultaneous storage of the result, leveraging the accumulation property of PCM. Note, however, that the crystallization dynamics of PCM exhibit sizeable intra- and inter-device variability, which may adversely affect the accumulation process accuracy [25]. Efficient factorization using PCM cells is a technique that could pave the way for massively parallelized computations [85], [86]. The accumulative property of PCM is also used to demonstrate unsupervised learning of temporal correlations between binary random processes [67].

2) Mixed-precision In-memory Computing: The idea behind mixed-precision in-memory computing is to use a low-precision computational memory unit to obtain an approximate solution in the part of the task that has high computational load, but exactness is not essential, and a high-precision processing unit to realize the part of the task that has low computational load [87]. One prime application of mixed-precision computing is for solving systems of linear equations. In an experimental demonstration of this concept using model covariance matrices, the linear system could be solved with high accuracy after performing a sufficient amount of iterations. However, problems tackled in this work were of relatively small scale because of the limited size and precision of the used hardware [87]. The mixed-precision in-memory computing concept is particularly well-suited for training Deep Neural Networks (DNNs). Recent Deep Learning (DL) research enables training of Quantized Neural Networks (QNNs) with extremely low precision (e.g., 1-bit) weights and activations at run-time. The computational memory unit is used to store the synaptic weights, forward and backward propagation passes are performed with low precision, while the gradients are accumulated in high precision (Fig. 2(a)) [88], [89]. Experiments on a two-layered neural network with PCM devices used for storing weights achieves 97.73% test accuracy on the task of classifying handwritten digits (based on the MNIST dataset), within 0.6% of the

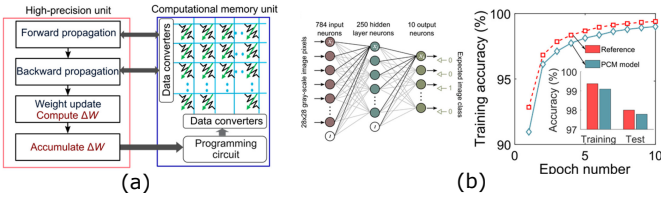

Fig. 2. (a) Block diagram illustration of the mixed-precision architecture for training deep neural networks. (b) PCM-based weights are shown to achieve comparable training accuracies as full precision training. Adapted from [89].

software baseline (Fig. 2(b)) [89], [90]. By exploiting the crossbar topology, it is also possible to estimate the gradients to perform the resulting synaptic weight update in-place in O(1) complexity [91]–[93].

3) Peripheral Circuits and Architectural Aspects of In-memory Computing: Fully analog peripheral circuits are sometimes implemented to avoid the complexity of digital-to-analog and Analog-to-Digital Conversions (ADCs), at the cost of less flexibility and accuracy [94], [95]. However, the preferred method for inputting digital data to PCM crossbars is pulse-width modulation, because the result of the computation, based on Ohm's law, is not affected by the non-linearity of the current–voltage characteristics of the devices. For digitizing the crossbar output, most works have employed ADCs [96] or sense amplifiers [97]. The precision of the digitization needs to be sufficient to properly resolve the analog multiply–accumulate operations, and a precision of at least four bits (including sign) has been found to be adequate for DNN inference applications [96], [97]. Because of their large area and power consumption, ADCs are usually multiplexed across multiple columns to reduce area and power consumption at the expense of increased latency. Multiplexing doesn't directly reduce energy consumption but can increase energy efficiency by reducing the operating frequency [96]. Scaling of the input and output ranges, such that the crossbar output falls within the limited dynamic range of the ADC, is critical to avoid a prohibitive loss of computational precision [98].

From an architectural point of view, a computational memory unit could have multiple in-memory computing cores connected through an on-chip network [99]. A significant research effort is currently focused on defining different hierarchical organizations that include digital processing units as well as conventional memory besides the crossbar arrays and associated peripheral circuitry [80], [100], [101].

V. Conclusions

After years of research and development, phase change memory has matured to the extend that versions of it are available on the market. Challenges still remain, in particular regarding its latency and durability, which warrant continued innovations at both the technology and system levels. However, PCM is already finding applications in big data analytics workloads as memory extension. The future holds further promise, as exciting applications such as in-memory computing have been demonstrated in prototypes. A key technological enabler is multilevel cell storage, which is expected to pave the way for broader adoption of this unique technology.

REFERENCES

[1] F. T. Hady *et al.*, "Platform storage performance with 3D XPoint technology," *Proceedings of the IEEE*, vol. 105, no. 9, pp. 1822–1833, 2017.

[2] I. Boybat *et al.*, "Neuromorphic computing with multi-memristive synapses," *Nature Communications*, vol. 9, Jun. 2018.

[3] G. W. Burr *et al.*, "Recent Progress in Phase-Change Memory Technology," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 6, no. 2, pp. 146–162, 2016.

[4] S. W. Fong *et al.*, "Phase-Change Memory—Towards a Storage-Class Memory," *IEEE Transactions on Electron Devices*, vol. 64, no. 11, pp. 4374–4385, 2017.

[5] M. Le Gallo and A. Sebastian, "An overview of phase-change memory device physics," *Journal of Physics D: Applied Physics*, vol. 53, no. 21, p. 213002, May 2020.

[6] T. Kim and S. Lee, "Evolution of Phase-Change Memory for the Storage-Class Memory and Beyond," *IEEE Transactions on Electron Devices*, vol. 67, no. 4, pp. 1394–1406, 2020.

[7] H. Y. Cheng *et al.*, "A high performance phase change memory with fast switching speed and high temperature retention by engineering the GexSbyTez phase change material," in *International Electron Devices Meeting*, 2011, pp. 3.4.1–3.4.4.

[8] Y. N. Hwang *et al.*, "Writing current reduction for high-density phase-change RAM," in *IEEE International Electron Devices Meeting*, 2003, pp. 37.1.1–37.1.4.

[9] J. H. Park *et al.*, "Reduction of RESET current in phase change memory devices by carbon doping in GeSbTe films," *Journal of Applied Physics*, vol. 117, no. 11, p. 115703, 2015.

[10] D. Ha and K. Kim, "Recent Advances in High Density Phase Change Memory (PRAM)," in *International Symposium on VLSI Technology, Systems and Applications (VLSI-TSA)*, 2007, pp. 1–4.

[11] D. H. Im *et al.*, "A unified 7.5nm dash-type confined cell for high performance PRAM device," in *IEEE International Electron Devices Meeting*, 2008, pp. 1–4.

[12] M. J. Kang *et al.*, "PRAM cell technology and characterization in 20nm node size," in *International Electron Devices Meeting*, 2011, pp. 3.1.1–3.1.4.

[13] I. S. Kim *et al.*, "High performance PRAM cell scalable to sub-20nm technology with below 4F2 cell size, extendable to DRAM applications," in *Symposium on VLSI Technology*, 2010, pp. 203–204.

[14] S. H. Lee *et al.*, "Programming disturbance and cell scaling in phase change memory: For up to 16nm based 4F2 cell," in *Symposium on VLSI Technology*, 2010, pp. 199–200.

[15] S. W. Fong *et al.*, "Dual-Layer Dielectric Stack for Thermally-Isolated Low-Power Phase-Change Memory," in *IEEE International Memory Workshop (IMW)*, 2017, pp. 1–4.

[16] S. Yoo *et al.*, "Electro-Thermal Model for Thermal Disturbance in Cross-Point Phase-Change Memory," *IEEE Transactions on Electron Devices*, vol. 67, no. 4, pp. 1454–1459, 2020.

[17] J. Y. Wu *et al.*, "A low power phase change memory using thermally confined TaN/TiN bottom electrode," in *International Electron Devices Meeting*, 2011, pp. 3.2.1–3.2.4.

[18] W. Kim *et al.*, "ALD-based confined PCM with a metallic liner toward unlimited endurance," in *IEEE International Electron Devices Meeting*, 2016, pp. 4.2.1–4.2.4.

[19] H. Y. Cheng *et al.*, "Novel fast-switching and high-data retention phase-change memory based on new Ga-Sb-Ge material," in *IEEE International Electron Devices Meeting (IEDM)*, 2015, pp. 3.5.1–3.5.4.

[20] Z. T. Song *et al.*, "High Endurance Phase Change Memory Chip Implemented based on Carbon-doped Ge2Sb2Te5 in 40 nm Node for Embedded Application," in *IEEE International Electron Devices Meeting (IEDM)*, 2018, pp. 27.5.1–27.5.4.

[21] N. Papandreou *et al.*, "Multilevel phase-change memory," in *17th IEEE International Conference on Electronics, Circuits and Systems*, 2010, pp. 1017–1020.

[22] B. L. Jackson *et al.*, "Nanoscale Electronic Synapses Using Phase Change Devices," *J. Emerg. Technol. Comput. Syst.*, vol. 9, no. 2, May 2013.

[23] A. Sebastian *et al.*, "Computational memory-based inference and training of deep neural networks," in *Symposium on VLSI Technology*, 2019, pp. T168–T169.

[24] N. Papandreou *et al.*, "Programming algorithms for multilevel phase-change memory," in *IEEE International Symposium of Circuits and Systems (ISCAS)*, 2011, pp. 329–332.

[25] M. Le Gallo *et al.*, "The complete time/temperature dependence of I-V drift in PCM devices," in *IEEE International Reliability Physics Symposium (IRPS)*, 2016, pp. MY–1–1–MY–1–6.

[26] M. Stanisavljevic *et al.*, "Phase-change memory: Feasibility of reliable multilevel-cell storage and retention at elevated temperatures," in *IEEE International Reliability Physics Symposium*, 2015, pp. 5B.6.1–5B.6.6.

[27] H. Pozidis *et al.*, "A Framework for Reliability Assessment in Multilevel Phase-Change Memory," in *4th IEEE International Memory Workshop*, 2012, pp. 1–4.

[28] M. Le Gallo *et al.*, "Compressed Sensing With Approximate Message Passing Using In-Memory Computing," *IEEE Transactions on Electron Devices*, vol. 65, no. 10, pp. 4304–4312, 2018.

[29] W. Koelmans *et al.*, "Projected phase-change memory devices," *Nature Communications*, vol. 6, 2015.

[30] S. Kim *et al.*, "A phase change memory cell with metallic surfactant layer as a resistance drift stabilizer," in *IEEE International Electron Devices Meeting*, 2013, pp. 30.7.1–30.7.4.

[31] K. Ding *et al.*, "Phase-change heterostructure enables ultralow noise and drift for memory operation," *Science*, vol. 366, no. 6462, pp. 210–215, 2019.

[32] N. Papandreou *et al.*, "Drift-resilient cell-state metric for multilevel phase-change memory," in *International Electron Devices Meeting*, 2011, pp. 3.5.1–3.5.4.

[33] ——, "Exploiting the non-linear current-voltage characteristics for resistive memory readout," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2018, pp. 1–5.

[34] J. Cheon *et al.*, "Non-resistance metric based read scheme for multi-level PCRAM in 25 nm technology," in *IEEE Custom Integrated Circuits Conference (CICC)*, 2015, pp. 1–4.

[35] N. Papandreou *et al.*, "Drift-Tolerant Multilevel Phase-Change Memory," in *3rd IEEE International Memory Workshop (IMW)*, 2011, pp. 1–4.

[36] H. Pozidis *et al.*, "Phase Change Memory Reliability: A Signal Processing and Coding Perspective," *IEEE Transactions on Magnetics*, vol. 51, no. 4, pp. 1–7, 2015.

[37] M. Stanisavljevic *et al.*, "Drift-Invariant Detection for Multilevel Phase-Change Memory," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2018, pp. 1–5.

[38] ——, "Demonstration of Reliable Triple-Level-Cell (TLC) Phase-Change Memory," in *8th IEEE International Memory Workshop (IMW)*, 2016, pp. 1–4.

[39] Y. C. Chen *et al.*, "Ultra-Thin Phase-Change Bridge Memory Device Using GeSb," in *International Electron Devices Meeting*, 2006, pp. 1–4.

[40] F. Xiong *et al.*, "Low-Power Switching of Phase-Change Materials with Carbon Nanotube Electrodes," *Science*, vol. 332, no. 6029, pp. 568–570, 2011.

[41] J. Liang *et al.*, "A 1.4μA reset current phase change memory cell with integrated carbon nanotube electrodes for cross-point memory application," in *Symposium on VLSI Technology*, 2011, pp. 100–101.

[42] F. Xiong *et al.*, "Towards ultimate scaling limits of phase-change memory," in *IEEE International Electron Devices Meeting (IEDM)*, 2016, pp. 4.1.1–4.1.4.

[43] G. Servalli, "A 45nm generation Phase Change Memory technology," in *IEEE International Electron Devices Meeting*, 2009, pp. 1–4.

[44] Y. Choi *et al.*, "A 20nm 1.8V 8Gb PRAM with 40MB/s program bandwidth," in *IEEE International Solid-State Circuits Conference*, 2012, pp. 46–48.

[45] S. Lee, "Technology scaling challenges and opportunities of memory devices," in *IEEE International Electron Devices Meeting (IEDM)*, 2016, pp. 1.1.1–1.1.8.

[46] W. C. Chien *et al.*, "Comprehensive Scaling Study on 3D Cross-Point PCM toward 1Znm Node for SCM Applications," in *Symposium on VLSI Technology*, 2019, pp. T60–T61.

[47] T. Kim *et al.*, "High-performance, cost-effective 2z nm two-deck cross-point memory integrated by self-align scheme for 128 Gb SCM," in *IEEE International Electron Devices Meeting*, 2018, pp. 37.1.1–37.1.4.

[48] N. Ciocchini and D. Ielmini, "Pulse-induced crystallization in phase-change memories under set and disturb conditions," *IEEE Transactions on Electron Devices*, vol. 62, no. 3, pp. 847–854, 2015.

[49] B. C. Lee *et al.*, "Architecting phase change memory as a scalable DRAM alternative," in *36th International Symposium on Computer Architecture*, 2009, pp. 2–13.

[50] M. K. Qureshi *et al.*, "Scalable high performance main memory system using phase-change memory technology," in *36th International Symposium on Computer Architecture*, 2009, pp. 24–33.

[51] S. Schechter *et al.*, "Use ECP, not ECC, for hard failures in resistive memories," in *37th International Symposium on Computer Architecture (ISCA)*, vol. 38, no. 3, pp. 141–152, 2010.

[52] N. H. Seong *et al.*, "SAFER: Stuck-at-fault error recovery for memories," in *43rd IEEE/ACM International Symposium on Microarchitecture*, 2010, pp. 115–124.

[53] D. H. Yoon *et al.*, "FREE-p: Protecting non-volatile memory against both hard and soft errors," in *IEEE 17th International Symposium on High Performance Computer Architecture*, 2011, pp. 466–477.

[54] B.-D. Yang *et al.*, "A low power phase-change random access memory using a data-comparison write scheme," in *IEEE International Symposium on Circuits and Systems*, 2007, pp. 3014–3017.

[55] S. Cho and H. Lee, "Flip-N-Write: A simple deterministic technique to improve PRAM write performance, energy and endurance," in *42nd IEEE/ACM International Symposium on Microarchitecture*, 2009, pp. 347–357.

[56] A. P. Ferreira *et al.*, "Increasing PCM main memory lifetime," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2010, pp. 914–919.

[57] P. Zhou *et al.*, "A durable and energy efficient main memory using phase change memory technology," in *36th International Symposium on Computer Architecture (ISCA)*, vol. 37, no. 3, pp. 14–23, 2009.

[58] M. K. Qureshi *et al.*, "Enhancing lifetime and security of PCM-based main memory with start-gap wear leveling," in *IEEE/ACM International Symposium on Microarchitecture*, 2009, pp. 14–23.

[59] L. E. Ramos *et al.*, "Page placement in hybrid memory systems," in *International Conference on Supercomputing*, 2011, pp. 85–95.

[60] M. K. Qureshi *et al.*, "Morphable memory system: A robust architecture for exploiting multi-level phase change memories," in *37th International Symposium on Computer Architecture (ISCA)*, vol. 38, no. 3, pp. 153–162, 2010.

[61] R. Stoica *et al.*, "Understanding the design trade-offs of hybrid flash controllers," in *27th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, 2019, pp. 152–164.

[62] G. Gill *et al.*, "Single machine graph analytics on massive datasets using Intel Optane DC persistent memory," in *Proceedings of the VLDB Endowment*, vol. 13, no. 8, pp. 1304–1318, 2020.

[63] O. Patil *et al.*, "Performance characterization of a DRAM-NVM hybrid memory architecture for HPC applications using Intel Optane DC Persistent Memory Modules," in *Proceedings of the International Symposium on Memory Systems*, 2019, pp. 288–303.

[64] I. B. Peng *et al.*, "System evaluation of the Intel Optane byte-addressable NVM," in *International Symposium on Memory Systems*, 2019, pp. 304–315.

[65] A. Shanbhag *et al.*, "Large-scale in-memory analytics on Intel® Optane™ DC persistent memory," in *16th International Workshop on Data Management on New Hardware*, 2020, pp. 1–8.

[66] S. Hamdioui *et al.*, "Applications of Computation-In-Memory Architectures based on Memristive Devices," in *Design, Automation Test in Europe Conference Exhibition (DATE)*, 2019, pp. 486–491.

[67] A. Sebastian *et al.*, "Temporal correlation detection using computational phase-change memory," *Nature Communications*, vol. 8, no. 1, pp. 1–10, 10 2017.

[68] M. di Ventra and Y. V. Pershin, "The parallel approach," *Nature Physics*, vol. 9, no. 4, pp. 200–202, Apr. 2013.

[69] G. W. Burr *et al.*, "Neuromorphic computing using non-volatile memory," *Advances in Physics X*, vol. 2, no. 1, pp. 89–124, Jan. 2017.

[70] M. Zidan *et al.*, "The future of electronics based on memristive systems," *Nature Electronics*, vol. 1, p. 22–29, 01 2018.

[71] D. Ielmini and H.-S. P. Wong, "In-memory computing with resistive switching devices," *Nature Electronics*, vol. 1, 06 2018.

[72] A. Sebastian *et al.*, "Crystal growth within a phase change memory cell." *Nature Communications*, vol. 5, p. 4314, Jul. 2014.

[73] J. Borghetti *et al.*, "'Memristive' switches enable 'stateful' logic operations via material implication." *Nature*, vol. 464, pp.873–876, Apr. 2010.

[74] X. Zhu *et al.*, "Performing stateful logic on memristor memory," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 60, no. 10, pp. 682–686, 2013.

[75] N. Talati *et al.*, "Logic Design Within Memristive Memories Using Memristor-Aided loGIC (MAGIC)," *IEEE Transactions on Nanotechnology*, vol. 15, pp. 635–650, 2016.

[76] M. Cassinerio *et al.*, "Logic computation in phase change materials by threshold and memory switching." *Advanced materials*, vol. 25, pp. 5975–5980, Nov. 2013.

[77] D. Loke *et al.*, "Ultrafast phase-change logic device driven by melting processes," *Proceedings of the National Academy of Sciences*, vol. 111, Sep. 2014.

[78] S. Li *et al.*, "Pinatubo: A processing-in-memory architecture for bulk bitwise operations in emerging non-volatile memories," in *53nd ACM/IEEE Design Automation Conference (DAC)*, 2016, pp. 1–6.

[79] M. Nardone *et al.*, "Possible mechanisms for 1/f noise in chalcogenide glasses: A theoretical description," *Phys. Rev. B*, vol. 79, 04 2009.

[80] A. Shafiee *et al.*, "ISAAC: A Convolutional Neural Network Accelerator with In-Situ Analog Arithmetic in Crossbars," in *43rd International Symposium on Computer Architecture (ISCA)*, 2016, pp. 14–26.

[81] M. Hu *et al.*, "Dot-product engine for neuromorphic computing: Programming 1T1M crossbar to accelerate matrix-vector multiplication," in *53rd Design Automation Conference, (DAC)*, 2016, pp. 19:1–19:6.

[82] ——, "Memristor-Based Analog Computation and Neural Network Classification with a Dot Product Engine," *Advanced Materials*, vol. 30, Jan. 2018.

[83] M. Le Gallo *et al.*, "Compressed sensing recovery using computational memory," in *IEEE International Electron Devices Meeting (IEDM)*, 2017, pp. 28.3.1–28.3.4.

[84] C. D. Wright *et al.*, "Arithmetic and biologically-inspired computing using phase-change materials." *Advanced materials*, vol. 23, pp. 3408–3413, Aug. 2011.

[85] ——, "Beyond von-Neumann Computing with Nanoscale Phase-Change Memory Devices," *Advanced Functional Materials*, vol. 23, p. 2248–2254, 05 2013.

[86] P. Hosseini *et al.*, "Accumulation-Based Computing Using Phase-Change Memories With FET Access Devices," *IEEE Electron Device Letters*, vol. 36, no. 9, pp. 975–977, 2015.

[87] M. Gallo *et al.*, "Mixed-precision in-memory computing," *Nature Electronics*, vol. 1, pp. 246–253, 04 2018.

[88] I. Hubara *et al.*, "Quantized Neural Networks: Training Neural Networks with Low Precision Weights and Activations," *J. Mach. Learn. Res.*, vol. 18, pp. 187:1–187:30, 2017.

[89] S. R. Nandakumar *et al.*, "Mixed-precision architecture based on computational memory for training deep neural networks," in *IEEE International Symposium on Circuits and Systems, (ISCAS)*, 2018, pp. 1–5.

[90] ——, "Mixed-Precision Deep Learning Based on Computational Memory," *Frontiers in Neuroscience*, vol. 14, 2020.

[91] G. W. Burr *et al.*, "Experimental Demonstration and Tolerancing of a Large-Scale Neural Network (165 000 Synapses) Using Phase-Change Memory as the Synaptic Weight Element," *IEEE Transactions on Electron Devices*, vol. 62, no. 11, pp. 3498–3507, 2015.

[92] H. Tsai *et al.*, "Recent progress in analog memory-based accelerators for deep learning," *Journal of Physics D Applied Physics*, vol. 51, no. 28, p. 283001, Jul. 2018.

[93] T. Gokmen and W. Haensch, "Algorithm for Training Neural Networks on Resistive Device Arrays," *Frontiers in Neuroscience*, vol. 14, 2020.

[94] D. Bankman *et al.*, "An Always-On 3.8 μJ/86% CIFAR-10 Mixed-Signal Binary CNN Processor With All Memory on Chip in 28-nm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 54, no. 1, pp. 158–172, 2019.

[95] S. Ambrogio *et al.*, "Equivalent-accuracy accelerated neural-network training using analogue memory," *Nature*, vol. 558, no. 7708, pp. 60–67, 2018.

[96] M. Gong *et al.*, "A 65nm thermometer-encoded time/charge-based compute-in-memory neural network accelerator at 0.735pj/MAC and 0.41pj/UPDATE," *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2020, doi: 10.1109/TCSII.2020.3027801.

[97] C. Xue *et al.*, "Embedded 1-Mb ReRAM-Based Computing-in- Memory Macro With Multibit Input and Weight for CNN-Based AI Edge Processors," *IEEE Journal of Solid-State Circuits*, vol. 55, no. 1, pp. 203–215, 2020.

[98] A. Sebastian *et al.*, "Memory devices and applications for in-memory computing." *Nature Nanotechnology*, vol. 15, pp. 529–544, Jul. 2020.

[99] M. Dazzi *et al.*, "5 Parallel Prism: A topology for pipelined implementations of convolutional neural networks using computational memory," in *NeurIPS MLSys Workshop*, Jun. 2019.

[100] A. Ankit *et al.*, "PUMA: A programmable ultra-efficient memristor-based accelerator for machine learning inference," in *24th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2019, p. 715–731.

[101] E. Eleftheriou *et al.*, "Deep learning acceleration based on in-memory computing," *IBM Journal of Research and Development*, vol. 63, no. 6, pp. 7:1–7:16, 2019.