A Survey of Storage Class Memory: Principles, Problems, and Possibilities

Aditya K Kamath, *Student Member, IEEE,* Leslie Monis, A Tarun Karthik, and Basavaraj Talawar, *Member, IEEE*

Abstract—Storage Class Memory (SCM) is a class of memory technology which has recently become viable for use. Their name arises from the fact that they exhibit non-volatility of data, similar to secondary storage while also having latencies comparable to primary memory and byte-addressibility. In this area, Phase Change Memory (PCM), Spin-Transfer-Torque Random Access Memory (STT-RAM), and Resistive RAM (ReRAM) have emerged as the major contenders for commercial and industrial use. In this paper, we describe how these memory types function. We then discuss the ongoing research being done in these fields, highlighting a few of the major works that have been undertaken.

Index Terms—storage class memory, classification, non-volatile memory

1 Introduction

Non-Volatile Memory (NVM) is a special class of memory that exhibits persistence, similar to that of secondary memory, while providing access speeds at least two magnitudes faster. NVM can potentially replace or augment any of the current existing memory layers, like that of the cache or primary memory. Storage Class Memory (SCM) is a subset of NVM whereby the device exhibits the feature of data persistence, while offering performance comparable to or better than that of primary memory along with byte-addressibility [1], [2], [3].

The current technology used for designing caches and primary memory face many problems that SCM could potentially solve. Static RAM (SRAM), the technology typically used in caches, suffers from low density making it more and more difficult to pack together in order to meet the increasing demands of speed [4]. Dynamic RAM (DRAM) used in primary memory has a better density, but suffers from slower access times, and requires constant power to refresh memory [5].

Three emerging SCM technologies are Phase Change Memory (PCM), Spin-Transfer-Torque Random Access Memory (STT-RAM), and Resistive Random Access Memory (ReRAM). As Table 1 and Table 2 indicate, their main advantages are that they exhibit low leakage power dissipation, since they do not need to be refreshed constantly. They also have the ability to be packed close together leading to higher data storage for the same given volume. This is since the cell size of STT-RAM and ReRAM, considered as possible replacements for SRAM, are nearly one-tenth the size of SRAM. Another important feature that these memory devices exhibit is their ability to function as Multi-Level Cells (MLCs) [6], this is where a single memory cell is capable of storing more than one bit. As the resistance of the memory element can be varied based on current supplied for SCM technology, MLC is achieved by assigning different resistance levels different bit values.

While these all seem like promising reasons to adopt SCM technology, they also face some severe drawbacks. PCM and STT-RAM both suffer from high write latencies [7], nearly ten times that of DRAM for PCM, and ten times SRAM for STT-RAM, and high write energy. PCM and ReRAM also suffer from a limit on write endurance before a hard error occurs. Unlike soft errors, a hard error is when a memory element gets permanently stuck at a certain value and cannot be changed [8]. These are major issues that needs to be tackled before SCM can be widely adopted.

In this paper, we give a brief overview of how the different types of SCM function. We then identify the key areas of focus, classifying the research being done in these fields.

Section 2 covers the details of how the three SCM based memories PCM, STT-RAM, and ReRAM function. Section 3 covers all the recent research trends that have been ongoing in this field. Section 4 gives a brief introduction of popular simulators oriented towards studying properties of SCM memory. Section 5 gives an overview of possibilities that may emerge in future.

2 SCM Characteristics

2.1 Phase Change Memory (PCM)

Figure 1 shows the basic view of a PCM cell, along with the requirements for programming the cell. The phase change material is a chalcogenide typically consisting of Ge-Sb-Te (GST) [9]. GST can exist in two states, amorphous and crystalline. In the amorphous state, the device exhibits higher resistance than the crystalline state. These two states can be used to store data, where high resistance is assigned a value of 0, and low resistance is assigned a 1. PCM can also be used to store multiple states as a Multi-Level Cell (MLC).

• *A. Kamath, L. Monis, A. Karthik, and B. Talawar are with the Department of Computer Science and Engineering, National Institute of Technology Karnataka, Surathkal, Karnataka, 575025.*
E-mails: akamath1997@gmail.com, lesliemonis@gmail.com, tarunkarthik999@gmail.com, basavaraj@nitk.edu.in

Manuscript received -; revised -.

This is by assigning data values to intermediate resistances. In order to SET the cell, it must reach a temperature below melting point, but above crystallization temperature. The cell needs time to change into crystalline state, which is why the pulse lasts much longer. For RESET, the cell temperature must be rapidly increased to melting point, then quickly cooled so that the material sets into amorphous state [10], [11].

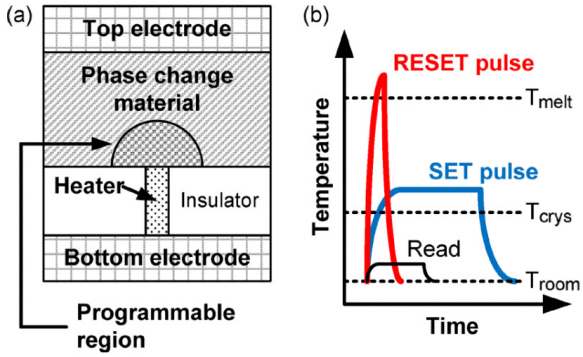


Fig. 1. (a) Cross-section of PCM cell. (b) Temperature and time required to program or read PCM cell. [12]

2.2 Spin-Transfer Torque Random Access Memory (STT-RAM)

STT-RAM represents bits of data by relying on differences in magnetic directions. There are two ferromagnetic layers in an STT-RAM separated by a dielectric. One layer is referred to as the reference layer, as it has a fixed magnetization direction. The other layer is referred to as free layer, whose magnetization direction can be controlled by passing current. Based on the relative directions of the two layers, the resistance of the magnetic tunnel junction (MTJ), will differ. In the case where the magnetization direction of the two layers are aligned, the resistance at the MTJ will be low, indicating a state of 1. If the two layers have opposing directions, the resistance becomes high, indicating a state of 0 [13]. Figure 2 shows the arrangement of the cell for the two states.

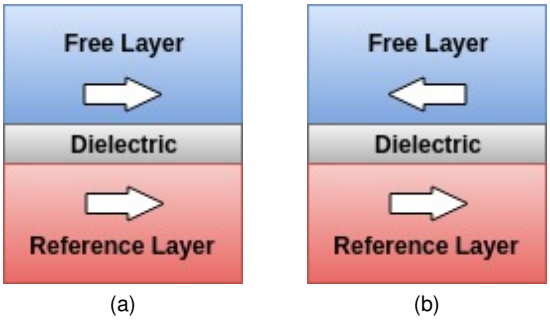


Fig. 2. (a) STT-RAM cell in SET (1) state. (b) STT-RAM cell in RESET (0) state.

2.3 Resistive Random Access Memory (ReRAM)

Resistive RAM experiences resistance changes due to electrochemical effects. The ReRAM cell consists of two metal electrodes, separated by a metal oxide layer. The behaviour of this system is dependent on the concentration of the oxygen vacancy in the metal oxide layer. By applying current to the cell, the state the cell is in can be switched. In the case of a bipolar ReRAM cell, the SET operation is undertaken when a negative bias is applied, while the RESET operation is undertaken when a positive bias is applied. An example of a ReRAM cell is a titanium oxide layer sandwiched between two platinum electrodes [14].

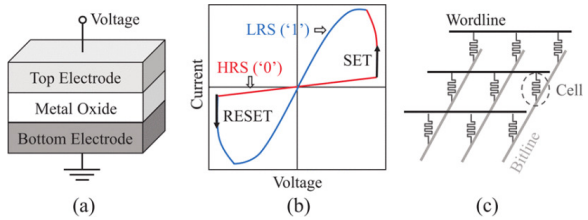


Fig. 3. (a) Structure of ReRAM cell. (b) Bias and current for write operations. [12]

3 Current Research Areas

As mentioned before, since the majority of SCM is still in the experimental stage, there is a lot of ongoing research on possible outcomes of these memory devices. Below is listed several research areas which are currently popular in this field. Table 3 lists all the areas along with the relevant papers. Some of these topics have relations where an impact in one topic may end up causing an impact in another topic. Thus, certain papers end up accomplishing multiple objectives and have been repeated multiple times in the table. Figure 4 shows some of the relations between solutions to problems that SCM faces.

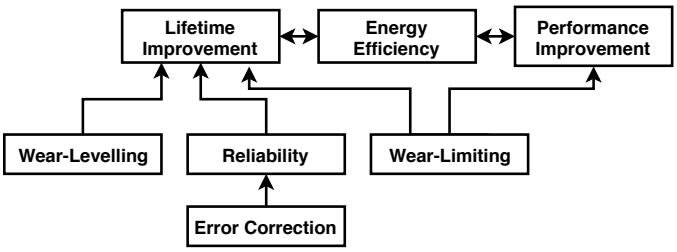


Fig. 4. Relations between different SCM issues. An arrow from one topic to another implies that changes in that field may cause impacts in the other topic.

3.1 Lifetime Improvement

While PCM has many advantages over DRAM memory, including non-volatility, low standby power, and high density, a major cause of concern is the endurance [10]. PCM and ReRAM technologies have a write endurance that are magnitudes less than those of memory technologies currently in use. Due to this, the lifetime of these memories is severely hindered.

TABLE 1
Comparison of Device Properties of Memory Technologies [7]

	Cell size (F^2)	Access Granularity	Read Latency	Write Latency	Erase Latency	Endurance	Standby Power
HDD	N/A	512B	5 ms	5 ms	N/A	$\geq 10^{15}$	1 W
SLC Flash	4 - 6	4KB	$25\mu s$	$500\mu s$	2 ms	$10^4 - 10^5$	0
DRAM	6 - 10	64B	50 ns	50 ns	N/A	$\geq 10^{15}$	Refresh Power
PCM	4 - 12	64B	50 ns	500 ns	N/A	$10^8 - 10^9$	0
STT-RAM	6 - 50	64B	10 ns	50 ns	N/A	$\geq 10^{15}$	0
ReRAM	4 - 10	64B	10 ns	50 ns	N/A	10^{11}	0

TABLE 2
Comparison of Cache Memory Technologies [15]

	SRAM	DRAM	STT-RAM	PCM
Cell Size (F^2)	120 - 200	4 - 6	6 - 50	4 - 12
Multi-level cell	No	No	Yes	Yes
Read speed	Very fast	Slow	Fast	Slow
Write speed	Very fast	Slow	Slow	Very slow
Read energy	Low	Medium	Low	Medium
Write energy	Low	Medium	High	High
Leakage	High	Medium	Low	Low
Throughput	Very high	Medium	High	Low
Write Endurance	10^{16}	10^{16}	$\geq 10^{12}$	$10^8 - 10^9$
Soft Error	Low	High	No	No

3.1.1 Wear-leveling

These are techniques which attempt to distribute writes evenly over all cells, by continuously trying to change the cell that write operations take place. Wear-leveling techniques already exist [16], [17] and are used for NAND Flash based SSDs. These involve creating a logical to physical mapping of addresses, storing the number of writes a line experiences in a table and using that data to periodically change the mapping. A shortcoming of this is that it requires a high overhead for the tables, and increases latencies due to reading and resolving of mappings. Moinuddin K. Qureshi et al. [18] proposed an alternative system which uses address space randomization to provide a low overhead wear-leveling approach.

Caches are optimised in order to maximise the temporal locality of the data in order to improve performance. This can lead to a disproportionate amount of writes being directed to certain cache lines. Endurance problems can thus arise when these caches consist of SCM. When the lifetime of the cache is estimated, the assumption is that writes are distributed evenly across the cache. Due to the disproportionate writes, certain cache lines may start to fail much before the estimated lifetime. Wear-leveling techniques are thus required in order to avoid this situation. Two simple types of wear leveling schemes can arise for caches. The first is intra-set wear-leveling, where an attempt is made to distribute writes evenly within a cache set. The second scheme is inter-set wear-leveling, where an attempt is made to avoid one cache set from receving more writes than the other sets. In this way the writes are distributed across the entire cache. For the most part, both of these schemes can operate independently, allowing for one inter-set and one intra-set wear-leveling methodology to coexist in the same cache.

One inter-set wear-leveling scheme involves set remapping [19] which has been proposed to try to tackle this problem. In this proposal, a register is maintained, and after a certain amount of time the value of the register is changed. The register is used to determine which set to write to. This method requires that tags maintain a set index, increasing the memory overhead. Jue Wang et al. [20] tried to introduce inter-set and intra-set wear-leveling using two methods. For inter-set leveling, the number of writes are measured, and after a threshold is reached, two sets are swapped. Data that was present in the sets before the swap are then invalidated. Since only two sets are swapped at a time, the performance does not take a severe toll. For intra-set leveling, the number of write hits is kept track of using a global counter. When the counter saturates, the cache line last written is flushed from the cache. This can cause a decrease in performance, and does not guarantee that the cache line that was flushed was actually frequently used.

Another proposal [21] breaks a cache set into multiple modules. The amount of writes each module receives is noted, and if the variation reaches a certain threshold, data in the most written module is moved to the least written module, and the most written module is temporarily disabled. In this way intra-set wear-leveling is achieved. A limitation is that this requires complicated computational circuitry when calculating the variation. Similar to this, Sukarn Agarwal et al. [22] proposed partitioning the cache into multiple windows. For each window, the number of writes received is noted. At regular intervals, the window with the most writes is set to read-only, and the counter for that window is reset achieveing intra-set wear-leveling. Sparsh Mittal et al. [23] proposed a intra-set wear-leveling mechanism which required maintaining a counter for each cache line. Comparisons are made between the counters of the same set, and when the difference between the largest and smallest value crosses a threshold, their data is swapped, and their counters are reset. The larger the associativity of the cache, the better the performance.

3.1.2 Wear-limiting

These are techniques which attempt to reduce the overall number of writes required for functioning. Data Comparison Write [24] schemes achieve this by first issuing a read on a cell, and then only writing to the cell if the written value differs from the stored value. Flip-N-Write [25] is another technique which builds on this, to reduce the number of writes. This is done by allocating an extra bit per block. When a write is issued to a block, the number of bit write operations needed is checked. If it is greater than half the size of the block, the extra bit is set, and each bit of the

data is complemented, then stored. Otherwise a normal data comparison write takes place. This method guarantees that a block write will require at maximum, bit writes equal to half the size of the block.

While wear-leveling and wear-limiting techniques help reduce the impact of writes, they do not have any effect on the limit on the number of writes that a cell can experience. An analytical model [26] found that there was a tradeoff between endurance and write speed. Lunkai Zhang et al. [27] built on this, proposing a wear-limiting methodology to reduce the wear that a single write operation causes by performing slow writes. Three different mechanisms have been discussed, in order to slow writes without adversely affecting the performance. Results show that this could potentially result in the lifetime doubling.

3.2 Error-Correction

Recent research [28], [29] has shown that a common cause of system crashing is due to memory failures. While many methods have arisen that attempt to reduce the effect of writes on cells and thus increase the lifetime of SCM, hard errors may still occur in the memory. By improving the error correction capabilities of SCM, the lifetime of the memory will improve, as the system will be able to last even after errors start occurring. Due to this, some procedures have to be developed in order to overcome these errors when they arise. Error-Correcting Codes (ECCs) [30], [31] have existed to mitigate effects of soft errors that occur in DRAM and SRAM. However, these approaches cannot be applied directly for SCM, as they have a high memory overhead. New methods for error mitigation have to be explored.

Error-Correcting Pointers (ECPs) [8] were suggested as an alternative for ECCs in SCM. An ECP is a pointer to a failed cell, which also stores the correct value for that cell. The suggestion given was to provide 6 ECPs for every 512-bit memory block. This enables each 512 bit block to recover from at most 6 hard errors. One limitation is that uniform allocation is not optimal. Certain rows may be more likely to experience hard failures, while others may be less likely. Once a 512 bit block fails beyond recovery, the entire page has to be discarded. As we can see from Figure 5 the vast majority of rows do not even reach four errors when the failure point of the page is reached. To counter this, proposals [32], [33] have been made to improve usage of ECPs, at the cost of additional latency and hardware. Other methods [34], [35] rely on the operating system for providing support.

Use of Error-Correcting Strings (ECSs) [36] is another proposal, in which variable-length offsets are used instead of fixed-length pointers. This allows for larger tolerance of hard errors until failure occurs. This is combined with a page-level error correction, which provides ECSs to blocks on demand. To reduce the wastage of pages caused by failures, Mohammad Khavari Tavana et al. [37] suggest utilizing Aegis [38] alongside ECP for error correction, and suggesting a method of block cooperation where the unused metadata space of one block is shared with other nearby blocks in order to reduce the chance of failure of the entire page.

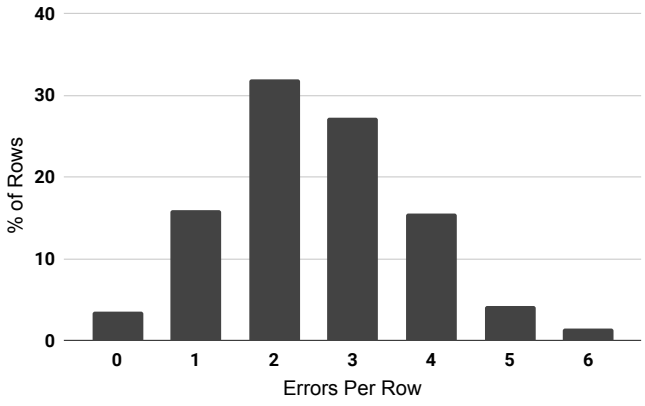


Fig. 5. Distribution of number of errors in a row when a page reaches failure point. [36]

3.3 Performance and Parallelism

The previously mentioned technique, Flip-N-Write [25], was a major milestone in the improvement of write performance. It provided a strict upper-bound on the time that would be required to complete a write operation on a block of data. It also reduced the worst-case number of write operations in half. Recent works are looking at further improving the write performance of SCM by writing multiple cells in parallel. For clarity, a *bit write* refers to a single bit whose value is changing, while a *write operation* refers to a single write issued to the controller which may consist of multiple bit writes.

Many techniques try to improve write parallelism by taking advantage of the asymmetries that exist in PCM. Writing a 0 (RESET) in PCM takes more current, but requires less time compared to writing a 1 (SET). Moinuddin K. Qureshi et al. [39] proposed a PreSET mechanism where all the cells of a PCM main memory line are preemptively SET when the respective cache line becomes dirty. In this way, when a writeback request is sent to main memory, only the faster RESET operations need to take place. This mechanism incurs a large lifetime penalty, as many unnecessary SET operations are required. Another method referred to as two-stage-write [40] exploits the write asymmetries by breaking a write operation into two stages. In the first stage, bits that are 0 are written at an accelerated rate. This is followed by the second stage, where bits that are 1 are written into in parallel to the extent that the power constraints allow. While this method improves write performance, it requires significant write asymmetries to give viable improvements. In addition, due to the separate stages, extra control circuitry is required. Zheng Li et al. [41] made an approach where the number of 1s and 0s to be written is measured. Following this, a schedule is created keeping in mind the asymmetries of power and time. In this way, concurrent bit writes are arranged without violating the power requirements. An alternative proposal called MaxPB [42] involves estimating the power requirements of each write operation that needs to be undertaken. Based on this, and the power budget given, the write operations are packaged together in order to obtain the minimum number of write operations required.

The write operations in a single package are then executed in parallel, allowing for an improvement in speed. Figure 7 compares the time taken for a few of the previously mentioned schemes.

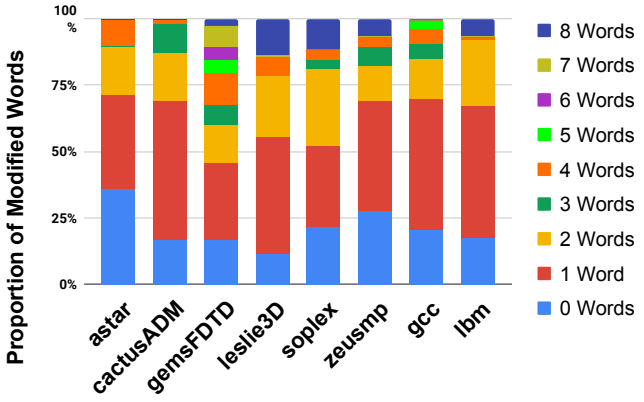


Fig. 6. Percentage of cache lines writes with given number of modified words, assuming a cache line is 64 bytes and a word is 8 bytes. [43]

An issue faced by writes on PCM memory is that write operations are handled at a cache line granularity. However, the entire cache line may not have been modified, or the write may in fact be a Silent Store [44]. Figure 6 shows the percentage of cache lines with the number of modified words for a subset of the SPEC CPU 2006 [45] benchmark suite. 0 words modified means that the write was a Silent Store [44], while 8 means that the entire cache line was modified. It can be clearly seen that in the majority of cases, only a small part of the cache line has actually been modified. Another problem arising is that while the write is being handled, the unmodified chips within a rank will remain idle. A proposal [46] was made to coalesce writes that effect the same rank into a single write operation. In this way, multiple write operations that target the same rank can be sent as a single operation, reducing the latencies and energy requirements that the multiple operations would have consumed. Similarily, MaxPB addressed this issue in order to reduce the time that write operations took. Since read operations are on the critical path, writes that are issued are put into a buffer until a certain threshold is reached. Once this threshold is reached, write operations that are required are undertaken until a lower threshold is reached. In this way, reads and writes occur in bursts. PCM also has a smaller write bandwidth than DRAM [47], [48]. Due to this, while write operations are undertaken, read operations are forced to wait, increasing their overall latency. One proposal [49] made to address this, was to cancel or pause writes when read operations were necessary. Mohammad Arjomand et al. [43] proposed several mechanisms involving utilizing error correction codes and rotating data mappings to allow for parallel read and write operations for chips that are currently being written into.

3.4 Multi-Level Cell

The resistance range of the cells in PCM are fairly large, due to the resistance gap between the amorphous and crystalline states. By manipulating the temperature and duration of a write operation to PCM, it is found that resistance values intermediate to the two states can be achieved. This allows for the possibility of a single cell to hold multiple bits by using intermediate resistance values for different states. These cells are thus referred to as Multi-Level Cells (MLCs). By using MLC instead of the previous Single-Level Cell (SLC), the density of PCM is greatly increased. In addition, the extra storage that is gained can help in tolerating failures.

However, MLCs comes with their own set of problems. Due to smaller gaps in resistance between bit levels, the write operation is more complicated and must be more precise. It is performed by applying multiple iterations of Program and Verify (P&V) [50], [51]. The operation takes place because the exact parameters (current/temperature) with which to issue the write to achieve a certain resistance varies between cells. Thus, an estimation is made and a write operation is undertaken. The value is then read, and if the resistance is not within the desired range, the process is repeated. This results in latencies almost 4 to 8 times higher than in SLC [49]. In addition the write energy consumption is also increased. MLCs also experience problems from resistance drift [52], [53]. Resistance drift is a phenomenon where the resistance of PCM cells increase with time. In SLC, since the resistance gap was so large this issue did not cause any problems, but in MLC the gaps are smaller and resistance drift can cause an unwanted change in state of the cell. This causes soft errors to pop up in PCM cells, and reduces the reliability of the device.

To resolve issues with reliability, the concept of guardbands are used [50]. In this, a band of resistance is kept between consecutive states. When a P&V operation takes place, it is ensured that the final value does not lie within the guardband range. The size of the guardband determines the amount of resistance drift that can be endured. A larger guardband means that data is retained longer as a larger resistance drift can be endured, at the cost of a larger number of iterations of P&V. A smaller guardband means that writes are less costly in terms of latency and energy, but the data will not be retained as long. There is thus a clear latency/retention time tradeoff involved [54]. It has been shown that the number of P&V iterations undertaken have no effect on the endurance of the cell [55], because RESET pulses determine endurance more than SET. To take advantage of all of these functionalities, Mingzhe Zhang et al. [56] proposed that two types of write operations exist. One, a long latency, high retention write, and the second a low latency, low retention write. They proposed a region retention monitor, to guess the required type of write when write operations are issued. Based on this guess, the appropriate write operation takes place. In this way, latencies can be improved for data that does not require high retention time.

Another method [57] proposed to overcome the reliability problems was to reduce the number of possible states that each cell of MLC has from 4 to 3, combining the two states with small resistance gap into a single state. Saeed Rashidi et al. [58] combined this with utilising the extra storage reserved for hard error correction to further alleviate the issue of reliability. This extra storage typically remains under-utilized for a long time, until hard errors start to

	Block 0 (B0)	Block 1 (B1)	Block 2 (B2)	Block 3 (B3)	Block 4 (B4)	Block 5 (B5)	Block 6 (B6)	Block 7 (B7)
Bit Changes (Inverted)	3 (3)	10 (6)	1 (1)	2 (2)	13 (3)	3 (3)	8 (8)	14 (2)

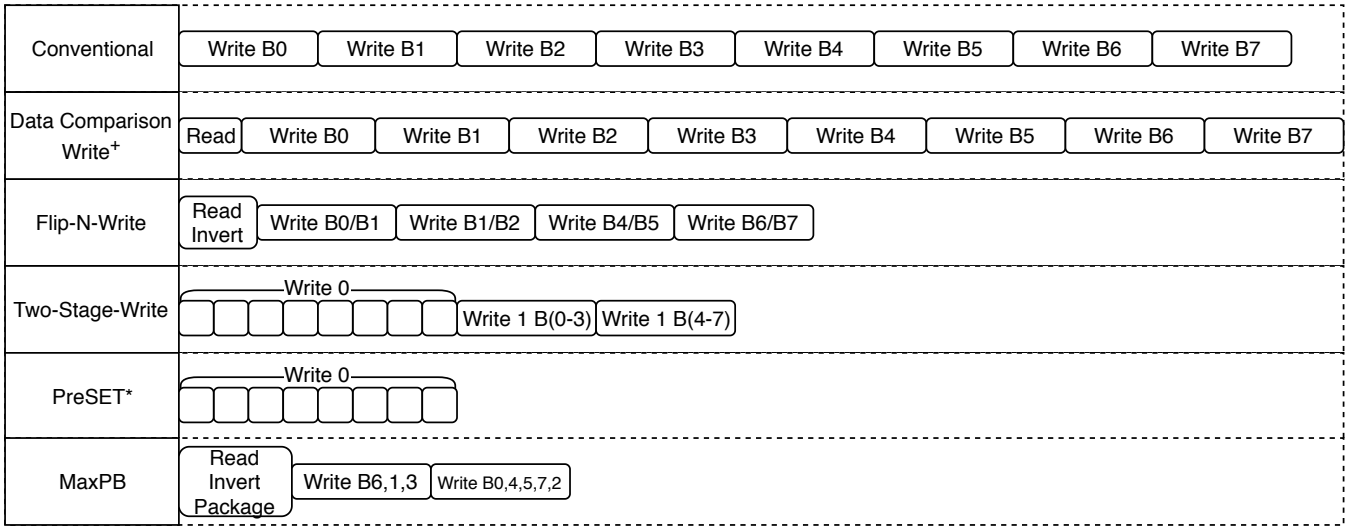


Fig. 7. Rough comparison of time taken for Conventional, Data Comparison Write [24], Flip-N-Write [25], Two-Stage-Write [40], PreSET [39], and MaxPB [42] write schemes, assuming blocks of 16 bits are written at a time. ($^+$) While Data Comparison Write has the overall highest latency, it saves both energy and endurance over the Conventional scheme, by first reading the cell. (*) While PreSET has the overall least latency, it reduces the lifetime of the memory, due to writing SET regardless of whether it's required.

occur. M-Metric [59] alongside this extra error-correction metadata is used with a tri-level PCM to improve the latencies and energy consumption as well as increase the overall IPC.

Compression techniques have been proposed as a method of reducing the impact on energy, lifetime, and latencies of MLC SCMs. A few techniques [60], [61] leverage compression to try to reduce the number of individual write operations required. However, these methods have a high overhead in terms of memory and computation. Another method is CompEx [62] coding, which attempts to provide a low overhead solution to classical methods of compression by combing statistical compression with expansion coding. Building on this, CompEx++ [63] integrates custom expansion codes and variable compressibility. These methods reduce the total energy consumption of the memory system by almost half, along with providing improvements in IPC and bandwidth.

Along with the other issues that MLC cells face, another key issue is the high read latency. In most of the common architectures [64], [65], [66], reading the values of an MLC cell works in an iterative fashion, where the values of the MSB are detected, as the resistance values differ by a large amount. Following this, the further bits are detected, with smaller consecutive resistance gaps. This method resembles a binary search across the possible resistance values of the states. A solution [67] presented to reduce the required iterations for reads is by striping multiple lines and grouping them on a single array. This method improves the read latency and IPC, at a small cost of lifetime.

3.5 Accelerators

Research has shown that PCM [68], STT-RAM [69], [70], and ReRAM [71], [72] are capable of performing operations and computations in addition to their capability of storing data. This method of having a device store data as well as perform computations allows for building in-memory accelerators which can quickly compute basic functions without requiring CPU intervention. Out of these ReRAM exhibits a crossbar array structure that is beneficial for in-memory processing of matrixvector multiplication. Due to this, many accelerators exploiting the structure of ReRAM have arisen [73], [74], [75].

Ping Chi et al. [76] used the structure in order to accelerate neural network computations. Ali Shafiee et al. [77] focussed on applying ReRAM for Convolutional Neural Network (CNN) applications. However, both of these methodologies did not work well with training and weight updations for CNNs. In an attempt to rectify this, Linghao Song et al. [78] built on these works, taking advantage of intra-layer parallelism to create an architecture that greatly boosts speed during both the training and inference phase of CNNs.

Besides focusing on neural network computations, Mahdi Nazm Bojnordi et al. [79] explored the usage of ReRAM in developing an accelerator for Boltzmann machine, resulting in a large improvement in performance along with reduced energy consumption when compared with a multicore implementation. Linghao Song et al. [80] proposed a structure to speed graph processing of graph algorithms that can be expressed as sparse matrix vector multiplication. S. Mittal compiled a survey [81] which explains neural network and processing in-memory accelerators that utilize ReRAM in more detail.

TABLE 3
Classification of Research Work

Field of Work	Papers
Lifetime Improvement	[18], [19], [20], [21], [22], [23], [27], [60], [62], [63], [82], [83], [84], [85]
Error-Correction	[8], [32], [33], [34], [35], [36], [37], [38], [86], [87], [88]
Performance and Parallelism	[39], [40], [41], [42], [43], [46], [49], [89], [90], [91], [92], [93]
Multi-Level Cells	[49], [50], [51], [52], [53], [54], [56], [57], [58], [65], [66], [67], [94], [95], [96], [97], [98], [99]
Accelerators	[68], [69], [70], [71], [72], [73], [74], [75], [76], [77], [78], [79], [80], [100], [101], [102], [103], [104], [105], [106], [107], [108], [109], [110], [111], [112], [113], [114]
SCM Type	**Papers**
ReRAM	
STT-RAM	
PCM	

4 Simulators

While several options for system simulation exist, the most popular simulators for SCM are GEM5 [115], NVSim [116] and NVMain [117], [118].

GEM5 is a simulator commonly used in computer architecture and system research. It can simulate system-level and processor microarchitecture. Provisions exist for full-system capabilities of Alpha, ARM, SPARC, and x86 simulations. The CPU model being simulated can be interchanged between a simple, an inorder, or a full-scale out-of-order CPU. For out-of-order CPU, traces can be run to obtain detailed results on the performance of the memory system.

NVSim is a circuit-level modelling tool, which provides estimations for the performance, energy, and area values for a given design specification. It supports the commonly used NVM memories, such as PCM, STT-RAM, ReRAM, as well as NAND based Flash memory. The intent of the tool is to help in creating optimized designs for the metrics mentioned, before fabricating the physical chip. NVSim is based off of the existing analytical model, CACTI [119], [120].

NVMain is a main memory simulator targeted for use with NVM-based memories. NVMain provides a cycle-accurate simulator which can estimate energy consumption at the system level. NVMain supports main memories of DRAM, PCM, STT-RAM, and ReRAM, as well as hybrid versions of these memories. NVMain has the option to be used in conjunction with GEM5 in order to evaluate full system simulation.

5 Future Outlook

Due to the many shortcomings of a pure SCM-based memory system, the majority of research now deals with hybrid memory. This gives the benefit of scalability, while allowing the shortcomings of SCM to be offset by a DRAM/SRAM component. With the release of Intel's 3D-XPoint memory [121] the use of this memory has gone from being hypothetical to practical, with claims that it performs 1000 times faster than NAND SSDs, with 1000 times the endurance. Instead of replacing existing layers of memory as was previously expected, current NVM memory is trying to bridge the gap between long latency NAND-based secondary memory, and primary memory, providing a large capacity of storage in doing so. In this way, NVM is seeking to change the memory hierarchy from a set of discrete memory layers to a spectrum of different memory possibilities.

Another aspect of SCM being actively looked at is its unique capability of supporting MLC. As mentioned previously, the use of MLC comes a new range of problems that have to be overcome before widespread use becomes practical. The International Technology Roadmap for Semiconductors (ITRS) [122], [123] predicts that densities of 3-bit and 4-bit MLCs may become possible in the near future. This increase in density may further exacerbate the existing problems that are being experienced.

References

[1] G. W. Burr, B. N. Kurdi, J. C. Scott, C. H. Lam, K. Gopalakrishnan, and R. S. Shenoy, "Overview of candidate device technologies for storage-class memory," *IBM Journal of Research and Development*, vol. 52, no. 4.5, pp. 449–464, July 2008.

[2] F. Hameed, C. Menard, and J. Castrillon, "Efficient stt-ram last-level-cache architecture to replace dram cache," in *Proceedings of the International Symposium on Memory Systems*, ser. MEMSYS '17. New York, NY, USA: ACM, 2017, pp. 141–151. [Online]. Available: http://doi.acm.org/10.1145/3132402.3132414

[3] J. B. Kotra, M. Arjomand, D. Guttman, M. T. Kandemir, and C. R. Das, "Re-nuca: A practical nuca architecture for reram based last-level caches," in *2016 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, May 2016, pp. 576–585.

[4] S. Mittal, "A survey of architectural techniques for improving cache power efficiency," *Sustainable Computing: Informatics and Systems*, vol. 4, no. 1, pp. 33 – 43, 2014.

[5] ——, "A survey of architectural techniques for dram power management," *Int. J. High Perform. Syst. Archit.*, vol. 4, no. 2, pp. 110–119, Dec. 2012. [Online]. Available: http://dx.doi.org/10.1504/IJHPSA.2012.050990

[6] S. Mittal, R. Wang, and J. Vetter, "Destiny: A comprehensive tool with 3d and multi-level cell memory modeling capability," *Journal of Low Power Electronics and Applications*, vol. 7, no. 3, 2017. [Online]. Available: http://www.mdpi.com/2079-9268/7/3/23

[7] S. Mittal and J. S. Vetter, "A survey of software techniques for using non-volatile memories for storage and main memory systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 5, pp. 1537–1550, May 2016.

[8] S. Schechter, G. H. Loh, K. Strauss, and D. Burger, "Use ecp, not ecc, for hard failures in resistive memories," *SIGARCH Comput. Archit. News*, vol. 38, no. 3, pp. 141–152, Jun. 2010. [Online]. Available: http://doi.acm.org/10.1145/1816038.1815980

[9] S. Hudgens and B. Johnson, "Overview of phase-change chalcogenide nonvolatile memory technology," *MRS Bulletin*, vol. 29, no. 11, p. 829832, 2004.

[10] S. Raoux, G. W. Burr, M. J. Breitwisch, C. T. Rettner, Y. . Chen, R. M. Shelby, M. Salinga, D. Krebs, S. . Chen, H. . Lung, and C. H. Lam, "Phase-change random access memory: A scalable technology," *IBM Journal of Research and Development*, vol. 52, no. 4.5, pp. 465–479, July 2008.

[11] A. Pirovano, A. L. Lacaita, A. Benvenuti, F. Pellizzer, S. Hudgens, and R. Bez, "Scaling analysis of phase-change memory technology," in *IEEE International Electron Devices Meeting 2003*, Dec 2003, pp. 29.6.1–29.6.4.

[12] H. . P. Wong, S. Raoux, S. Kim, J. Liang, J. P. Reifenberg, B. Rajendran, M. Asheghi, and K. E. Goodson, "Phase change memory," *Proceedings of the IEEE*, vol. 98, no. 12, pp. 2201–2227, Dec 2010.

[13] A. Nigam, C. W. Smullen, V. Mohan, E. Chen, S. Gurumurthi, and M. R. Stan, "Delivering on the promise of universal memory for spin-transfer torque ram (stt-ram)," in *IEEE/ACM International Symposium on Low Power Electronics and Design*, Aug 2011, pp. 121–126.

[14] J. J. Yang, M. D. Pickett, X. Li, D. A. A. Ohlberg, D. R. Stewart, and R. S. Williams, "Memristive switching mechanism for metal/oxide/metal nanodevices," *Nature Nanotechnology*, vol. 3, pp. 429 – 433, 2008. [Online]. Available: https://doi.org/10.1038/nnano.2008.160

[15] S. Mittal, J. S. Vetter, and D. Li, "A survey of architectural approaches for managing embedded dram and non-volatile on-chip caches," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 6, pp. 1524–1537, June 2015.

[16] A. Ban, "Wear leveling of static areas in flash memory," May 4 2004, uS Patent 6,732,221.

[17] T. Kgil, D. Roberts, and T. Mudge, "Improving nand flash based disk caches," in *ACM SIGARCH Computer Architecture News*, vol. 36, no. 3. IEEE Computer Society, 2008, pp. 327–338.

[18] M. K. Qureshi, J. Karidis, M. Franceschini, V. Srinivasan, L. Lastras, and B. Abali, "Enhancing lifetime and security of pcm-based main memory with start-gap wear leveling," in *Proceedings of the 42Nd Annual IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO 42. New York, NY, USA: ACM, 2009, pp. 14–23. [Online]. Available: http://doi.acm.org/10.1145/1669112.1669117

[19] Y. Chen, W.-F. Wong, H. Li, C.-K. Koh, Y. Zhang, and W. Wen, "On-chip caches built on multilevel spin-transfer torque ram cells and its optimizations," *J. Emerg. Technol. Comput. Syst.*, vol. 9, no. 2, pp. 16:1–16:22, May 2013. [Online]. Available: http://doi.acm.org/10.1145/2463585.2463592

[20] J. Wang, X. Dong, Y. Xie, and N. P. Jouppi, "i2wap: Improving non-volatile cache lifetime by reducing inter- and intra-set write variations," in *2013 IEEE 19th International Symposium on High Performance Computer Architecture (HPCA)*, Feb 2013, pp. 234–245.

[21] S. Mittal, J. S. Vetter, and D. Li, "Writesmoothing: Improving lifetime of non-volatile caches using intra-set wear-leveling," in *Proceedings of the 24th Edition of the Great Lakes Symposium on VLSI*, ser. GLSVLSI '14. New York, NY, USA: ACM, 2014, pp. 139–144. [Online]. Available: http://doi.acm.org/10.1145/2591513.2591525

[22] S. Agarwal and H. K. Kapoor, "Improving the lifetime of non-volatile cache by write restriction," *IEEE Transactions on Computers*, pp. 1–1, 2019.

[23] S. Mittal and J. S. Vetter, "Equalwrites: Reducing intra-set write variations for enhancing lifetime of non-volatile caches," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, no. 1, pp. 103–114, Jan 2016.

[24] P. Zhou, B. Zhao, J. Yang, and Y. Zhang, "A durable and energy efficient main memory using phase change memory technology," *SIGARCH Comput. Archit. News*, vol. 37, no. 3, pp. 14–23, Jun. 2009. [Online]. Available: http://doi.acm.org/10.1145/1555815.1555759

[25] S. Cho and H. Lee, "Flip-n-write: A simple deterministic technique to improve pram write performance, energy and endurance," in *2009 42nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, Dec 2009, pp. 347–357.

[26] D. B. Strukov, "Endurance-write-speed tradeoffs in nonvolatile memories," *Applied Physics A*, vol. 122, no. 4, p. 302, Mar 2016. [Online]. Available: https://doi.org/10.1007/s00339-016-9841-0

[27] L. Zhang, B. Neely, D. Franklin, D. Strukov, Y. Xie, and F. T. Chong, "Mellow writes: Extending lifetime in resistive memories through selective slow write backs," in *Proceedings of the 43rd International Symposium on Computer Architecture*, ser. ISCA '16. Piscataway, NJ, USA: IEEE Press, 2016, pp. 519–531. [Online]. Available: https://doi.org/10.1109/ISCA.2016.52

[28] V. Sridharan, N. DeBardeleben, S. Blanchard, K. B. Ferreira, J. Stearley, J. Shalf, and S. Gurumurthi, "Memory errors in modern systems: The good, the bad, and the ugly," *SIGARCH Comput. Archit. News*, vol. 43, no. 1, pp. 297–310, Mar. 2015. [Online]. Available: http://doi.acm.org/10.1145/2786763.2694348

[29] J. Meza, Q. Wu, S. Kumar, and O. Mutlu, "Revisiting memory errors in large-scale production data centers: Analysis and modeling of new trends from the field," in *2015 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, June 2015, pp. 415–426.

[30] C. L. Chen and M. Y. Hsiao, "Error-correcting codes for semiconductor memory applications: A state-of-the-art review," *IBM Journal of Research and Development*, vol. 28, no. 2, pp. 124–134, March 1984.

[31] D. Rossi, N. Timoncini, M. Spica, and C. Metra, "Error correcting code analysis for cache memory high reliability and performance," in *2011 Design, Automation Test in Europe*, March 2011, pp. 1–6.

[32] M. K. Qureshi, "Pay-as-you-go: Low-overhead hard-error correction for phase change memories," in *2011 44th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, Dec 2011, pp. 318–328.

[33] R. Azevedo, J. D. Davis, K. Strauss, P. Gopalan, M. Manasse, and S. Yekhanin, "Zombie memory: Extending memory lifetime by reviving dead blocks," *SIGARCH Comput. Archit. News*, vol. 41, no. 3, pp. 452–463, Jun. 2013. [Online]. Available: http://doi.acm.org/10.1145/2508148.2485961

[34] D. H. Yoon, N. Muralimanohar, J. Chang, P. Ranganathan, N. P. Jouppi, and M. Erez, "Free-p: Protecting non-volatile memory against both hard and soft errors," in *2011 IEEE 17th International Symposium on High Performance Computer Architecture*, Feb 2011, pp. 466–477.

[35] E. Ipek, J. Condit, E. B. Nightingale, D. Burger, and T. Moscibroda, "Dynamically replicated memory: Building reliable systems from nanoscale resistive memories," *SIGPLAN Not.*, vol. 45, no. 3, pp. 3–14, Mar. 2010. [Online]. Available: http://doi.acm.org/10.1145/1735971.1736023

[36] S. Swami, P. M. Palangappa, and K. Mohanram, "Ecs: Error-correcting strings for lifetime improvements in nonvolatile memories," *ACM Trans. Archit. Code Optim.*, vol. 14, no. 4, pp. 40:1–40:29, Dec. 2017. [Online]. Available: http://doi.acm.org/10.1145/3151083

[37] M. K. Tavana, A. K. Ziabari, and D. Kaeli, "Block cooperation: Advancing lifetime of resistive memories by increasing utilization of error correcting codes," *ACM Trans. Archit. Code Optim.*, vol. 15, no. 3, pp. 36:1–36:26, Aug. 2018. [Online]. Available: http://doi.acm.org/10.1145/3243906

[38] J. Fan, S. Jiang, J. Shu, Y. Zhang, and W. Zhen, "Aegis: Partitioning data block for efficient recovery of stuck-at-faults in phase

change memory," in *2013 46th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, Dec 2013, pp. 433–444.

[39] M. K. Qureshi, M. M. Franceschini, A. Jagmohan, and L. A. Lastras, "Preset: Improving performance of phase change memories by exploiting asymmetry in write times," *SIGARCH Comput. Archit. News*, vol. 40, no. 3, pp. 380–391, Jun. 2012. [Online]. Available: http://doi.acm.org/10.1145/2366231.2337203

[40] J. Yue and Y. Zhu, "Accelerating write by exploiting pcm asymmetries," in *2013 IEEE 19th International Symposium on High Performance Computer Architecture (HPCA)*, Feb 2013, pp. 282–293.

[41] Z. Li, F. Wang, D. Feng, Y. Hua, W. Tong, J. Liu, and X. Liu, "Tetris write: Exploring more write parallelism considering pcm asymmetries," in *2016 45th International Conference on Parallel Processing (ICPP)*, Aug 2016, pp. 159–168.

[42] Z. Li, F. Wang, D. Feng, Y. Hua, J. Liu, and W. Tong, "Maxpb: Accelerating pcm write by maximizing the power budget utilization," *ACM Trans. Archit. Code Optim.*, vol. 13, no. 4, pp. 46:1–46:26, Dec. 2016. [Online]. Available: http://doi.acm.org/10.1145/3012007

[43] M. Arjomand, M. T. Kandemir, A. Sivasubramaniam, and C. R. Das, "Boosting access parallelism to pcm-based main memory," in *2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)*, June 2016, pp. 695–706.

[44] K. M. Lepak and M. H. Lipasti, "Silent stores for free," in *Proceedings 33rd Annual IEEE/ACM International Symposium on Microarchitecture. MICRO-33 2000*, Dec 2000, pp. 22–31.

[45] C. D. Spradling, "Spec cpu2006 benchmark tools," *SIGARCH Comput. Archit. News*, vol. 35, no. 1, pp. 130–134, Mar. 2007. [Online]. Available: http://doi.acm.org/10.1145/1241601.1241625

[46] F. Xia, D. Jiang, J. Xiong, M. Chen, L. Zhang, and N. Sun, "Dwc: Dynamic write consolidation for phase change memory systems," in *Proceedings of the 28th ACM International Conference on Supercomputing*, ser. ICS '14. New York, NY, USA: ACM, 2014, pp. 211–220. [Online]. Available: http://doi.acm.org/10.1145/2597652.2597661

[47] Y. Choi, I. Song, M. Park, H. Chung, S. Chang, B. Cho, J. Kim, Y. Oh, D. Kwon, J. Sunwoo, J. Shin, Y. Rho, C. Lee, M. G. Kang, J. Lee, Y. Kwon, S. Kim, J. Kim, Y. Lee, Q. Wang, S. Cha, S. Ahn, H. Horii, J. Lee, K. Kim, H. Joo, K. Lee, Y. Lee, J. Yoo, and G. Jeong, "A 20nm 1.8v 8gb pram with 40mb/s program bandwidth," in *2012 IEEE International Solid-State Circuits Conference*, Feb 2012, pp. 46–48.

[48] M. T. Inc., "Micron ddr3 sdram part mt41k1g8sn-107," 2015. [Online]. Available: http://www.micron.com/parts/dram/ddr3-sdram/mt41k1g8sn-107

[49] M. K. Qureshi, M. M. Franceschini, and L. A. Lastras-Montao, "Improving read performance of phase change memories via write cancellation and write pausing," in *HPCA - 16 2010 The Sixteenth International Symposium on High-Performance Computer Architecture*, Jan 2010, pp. 1–11.

[50] T. Nirschl, J. B. Philipp, T. D. Happ, G. W. Burr, B. Rajendran, M. . Lee, A. Schrott, M. Yang, M. Breitwisch, C. . Chen, E. Joseph, M. Lamorey, R. Cheek, S. . Chen, S. Zaidi, S. Raoux, Y. C. Chen, Y. Zhu, R. Bergmann, H. . Lung, and C. Lam, "Write strategies for 2 and 4-bit multi-level phase-change memory," in *2007 IEEE International Electron Devices Meeting*, Dec 2007, pp. 461–464.

[51] L. Jiang, B. Zhao, Y. Zhang, J. Yang, and B. R. Childers, "Improving write operations in mlc phase change memory," in *IEEE International Symposium on High-Performance Comp Architecture*, Feb 2012, pp. 1–10.

[52] J. Li, B. Luan, and C. Lam, "Resistance drift in phase change memory," in *2012 IEEE International Reliability Physics Symposium (IRPS)*, April 2012, pp. 6C.1.1–6C.1.6.

[53] W. Zhang and T. Li, "Helmet: A resistance drift resilient architecture for multi-level cell phase change memory system," in *2011 IEEE/IFIP 41st International Conference on Dependable Systems Networks (DSN)*, June 2011, pp. 197–208.

[54] Q. Li, L. Jiang, Y. Zhang, Y. He, and C. J. Xue, "Compiler directed write-mode selection for high performance low power volatile pcm," *SIGPLAN Not.*, vol. 48, no. 5, pp. 101–110, Jun. 2013. [Online]. Available: http://doi.acm.org/10.1145/2499369.2465564

[55] K. Kim and S. J. Ahn, "Reliability investigations for manufacturable high density pram," in *2005 IEEE International Reliability Physics Symposium, 2005. Proceedings. 43rd Annual.*, April 2005, pp. 157–162.

[56] M. Zhang, L. Zhang, L. Jiang, Z. Liu, and F. T. Chong, "Balancing performance and lifetime of mlc pcm by using a region retention monitor," in *2017 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, Feb 2017, pp. 385–396.

[57] N. H. Seong, S. Yeo, and H.-H. S. Lee, "Tri-level-cell phase change memory: Toward an efficient and reliable memory system," in *Proceedings of the 40th Annual International Symposium on Computer Architecture*, ser. ISCA '13. New York, NY, USA: ACM, 2013, pp. 440–451. [Online]. Available: http://doi.acm.org/10.1145/2485922.2485960

[58] S. Rashidi, M. Jalili, and H. Sarbazi-Azad, "Improving mlc pcm performance through relaxed write and read for intermediate resistance levels," *ACM Trans. Archit. Code Optim.*, vol. 15, no. 1, pp. 12:1–12:31, Mar. 2018. [Online]. Available: http://doi.acm.org/10.1145/3177965

[59] 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*, vol. 110, no. 8, p. 084505, 2011. [Online]. Available: https://doi.org/10.1063/1.3653279

[60] D. B. Dgien, P. M. Palangappa, N. A. Hunter, J. Li, and K. Mohanram, "Compression architecture for bit-write reduction in non-volatile memory technologies," in *2014 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH)*, July 2014, pp. 51–56.

[61] S. Sardashti and D. A. Wood, "Decoupled compressed cache: Exploiting spatial locality for energy-optimized compressed caching," in *Proceedings of the 46th Annual IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO-46. New York, NY, USA: ACM, 2013, pp. 62–73. [Online]. Available: http://doi.acm.org/10.1145/2540708.2540715

[62] P. M. Palangappa and K. Mohanram, "Compex: Compression-expansion coding for energy, latency, and lifetime improvements in mlc/tlc nvm," in *2016 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, March 2016, pp. 90–101.

[63] ——, "Compex++: Compression-expansion coding for energy, latency, and lifetime improvements in mlc/tlc nvms," *ACM Trans. Archit. Code Optim.*, vol. 14, no. 1, pp. 10:1–10:30, Apr. 2017. [Online]. Available: http://doi.acm.org/10.1145/3050440

[64] G. F. Close, U. Frey, J. Morrish, R. Jordan, S. C. Lewis, T. Maffitt, M. J. BrightSky, C. Hagleitner, C. H. Lam, and E. Eleftheriou, "A 256-mcell phase-change memory chip operating at2+bit/cell," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 60, no. 6, pp. 1521–1533, June 2013.

[65] F. Bedeschi, R. Fackenthal, C. Resta, E. M. Donze, M. Jagasivamani, E. C. Buda, F. Pellizzer, D. W. Chow, A. Cabrini, G. M. A. 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 Journal of Solid-State Circuits*, vol. 44, no. 1, pp. 217–227, Jan 2009.

[66] M. K. Qureshi, M. M. Franceschini, L. A. Lastras-Montaño, and J. P. Karidis, "Morphable memory system: A robust architecture for exploiting multi-level phase change memories," *SIGARCH Comput. Archit. News*, vol. 38, no. 3, pp. 153–162, Jun. 2010. [Online]. Available: http://doi.acm.org/10.1145/1816038.1815981

[67] M. Hoseinzadeh, M. Arjomand, and H. Sarbazi-Azad, "Spcm: The striped phase change memory," *ACM Trans. Archit. Code Optim.*, vol. 12, no. 4, pp. 38:1–38:25, Nov. 2015. [Online]. Available: http://doi.acm.org/10.1145/2829951

[68] G. W. Burr, R. M. Shelby, S. Sidler, C. di Nolfo, J. Jang, I. Boybat, R. S. Shenoy, P. Narayanan, K. Virwani, E. U. Giacometti, B. N. Kurdi, and H. Hwang, "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, Nov 2015.

[69] A. F. Vincent, J. Larroque, W. S. Zhao, N. B. Romdhane, O. Bichler, C. Gamrat, J. . Klein, S. Galdin-Retailleau, and D. Querlioz, "Spin-transfer torque magnetic memory as a stochastic memristive synapse," in *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*, June 2014, pp. 1074–1077.

[70] A. F. Vincent, J. Larroque, N. Locatelli, N. Ben Romdhane, O. Bichler, C. Gamrat, W. S. Zhao, J. Klein, S. Galdin-Retailleau, and D. Querlioz, "Spin-transfer torque magnetic memory as a stochastic memristive synapse for neuromorphic systems," *IEEE*

Transactions on Biomedical Circuits and Systems, vol. 9, no. 2, pp. 166–174, April 2015.

[71] M. Hu, H. Li, Q. Wu, and G. S. Rose, "Hardware realization of bsb recall function using memristor crossbar arrays," in *Proceedings of the 49th Annual Design Automation Conference*, ser. DAC '12. New York, NY, USA: ACM, 2012, pp. 498–503. [Online]. Available: http://doi.acm.org/10.1145/2228360.2228448

[72] B. Li, Y. Shan, M. Hu, Y. Wang, Y. Chen, and H. Yang, "Memristor-based approximated computation," in *Proceedings of the 2013 International Symposium on Low Power Electronics and Design*, ser. ISLPED '13. Piscataway, NJ, USA: IEEE Press, 2013, pp. 242–247. [Online]. Available: http://dl.acm.org/citation.cfm?id=2648668.2648729

[73] M. Prezioso, F. Merrikh-Bayat, B. D. Hoskins, G. C. Adam, K. K. Likharev, and D. B. Strukov, "Training and operation of an integrated neuromorphic network based on metal-oxide memristors," *Nature*, vol. 521, no. 61, 2015. [Online]. Available: https://doi.org/10.1038/nature14441

[74] Y. Kim, Y. Zhang, and P. Li, "A reconfigurable digital neuromorphic processor with memristive synaptic crossbar for cognitive computing," *J. Emerg. Technol. Comput. Syst.*, vol. 11, no. 4, pp. 38:1–38:25, Apr. 2015. [Online]. Available: http://doi.acm.org/10.1145/2700234

[75] Z. Chen, B. Gao, Z. Zhou, P. Huang, H. Li, W. Ma, D. Zhu, L. Liu, X. Liu, J. Kang, and H. Chen, "Optimized learning scheme for grayscale image recognition in a rram based analog neuromorphic system," in *2015 IEEE International Electron Devices Meeting (IEDM)*, Dec 2015, pp. 17.7.1–17.7.4.

[76] P. Chi, S. Li, C. Xu, T. Zhang, J. Zhao, Y. Liu, Y. Wang, and Y. Xie, "Prime: A novel processing-in-memory architecture for neural network computation in reram-based main memory," in *Proceedings of the 43rd International Symposium on Computer Architecture*, ser. ISCA '16. Piscataway, NJ, USA: IEEE Press, 2016, pp. 27–39. [Online]. Available: https://doi.org/10.1109/ISCA.2016.13

[77] A. Shafiee, A. Nag, N. Muralimanohar, R. Balasubramonian, J. P. Strachan, M. Hu, R. S. Williams, and V. Srikumar, "Isaac: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars," in *Proceedings of the 43rd International Symposium on Computer Architecture*, ser. ISCA '16. Piscataway, NJ, USA: IEEE Press, 2016, pp. 14–26. [Online]. Available: https://doi.org/10.1109/ISCA.2016.12

[78] L. Song, X. Qian, H. Li, and Y. Chen, "Pipelayer: A pipelined reram-based accelerator for deep learning," in *2017 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, Feb 2017, pp. 541–552.

[79] M. N. Bojnordi and E. Ipek, "Memristive boltzmann machine: A hardware accelerator for combinatorial optimization and deep learning," in *2016 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, March 2016, pp. 1–13.

[80] L. Song, Y. Zhuo, X. Qian, H. Li, and Y. Chen, "Graphr: Accelerating graph processing using reram," in *2018 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, Feb 2018, pp. 531–543.

[81] S. Mittal, "A survey of reram-based architectures for processing-in-memory and neural networks," *Machine Learning and Knowledge Extraction*, vol. 1, no. 1, pp. 75–114, 2018. [Online]. Available: http://www.mdpi.com/2504-4990/1/1/5

[82] M. Jalili and H. Sarbazi-Azad, "Captopril: Reducing the pressure of bit flips on hot locations in non-volatile main memories," in *2016 Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2016, pp. 1116–1119.

[83] A. A. García, R. de Jong, W. Wang, and S. Diestelhorst, "Composing lifetime enhancing techniques for non-volatile main memories," in *Proceedings of the International Symposium on Memory Systems*, ser. MEMSYS '17. New York, NY, USA: ACM, 2017, pp. 363–373. [Online]. Available: http://doi.acm.org/10.1145/3132402.3132411

[84] Y. Guo, Y. Hua, and P. Zuo, "Dfpc: A dynamic frequent pattern compression scheme in nvm-based main memory," in *2018 Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2018, pp. 1622–1627.

[85] S. Agarwal and H. K. Kapoor, "Targeting inter set write variation to improve the lifetime of non-volatile cache using fellow sets," in *2017 IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC)*, Oct 2017, pp. 1–6.

[86] X. Wang, M. Mao, E. Eken, W. Wen, H. Li, and Y. Chen, "Sliding basket: An adaptive ecc scheme for runtime write failure suppression of stt-ram cache," in *2016 Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2016, pp. 762–767.

[87] M. K. Tavana, A. K. Ziabari, and D. Kaeli, "Live together or die alone: Block cooperation to extend lifetime of resistive memories," in *Design, Automation Test in Europe Conference Exhibition (DATE), 2017*, March 2017, pp. 1098–1103.

[88] M. K. Tavana, A. K. Ziabari, M. Arjomand, M. Kandemir, C. Das, and D. Kaeli, "Remap: A reliability/endurance mechanism for advancing pcm," in *Proceedings of the International Symposium on Memory Systems*, ser. MEMSYS '17. New York, NY, USA: ACM, 2017, pp. 385–398. [Online]. Available: http://doi.acm.org/10.1145/3132402.3132421

[89] H. Zhang, N. Xiao, F. Liu, and Z. Chen, "Leader: Accelerating reram-based main memory by leveraging access latency discrepancy in crossbar arrays," in *2016 Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2016, pp. 756–761.

[90] Z. Li, F. Wang, Y. Hua, W. Tong, J. Liu, Y. Chen, and D. Feng, "Exploiting more parallelism from write operations on pcm," in *2016 Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2016, pp. 768–773.

[91] N. Sayed, R. Bishnoi, F. Oboril, and M. B. Tahoori, "A cross-layer adaptive approach for performance and power optimization in stt-mram," in *2018 Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2018, pp. 791–796.

[92] P. Zhou, B. Zhao, J. Yang, and Y. Zhang, "Throughput enhancement for phase change memories," *IEEE Transactions on Computers*, vol. 63, no. 8, pp. 2080–2093, Aug 2014.

[93] Y. Li, X. Li, L. Ju, and Z. Jia, "A three-stage-write scheme with flip-bit for pcm main memory," in *The 20th Asia and South Pacific Design Automation Conference*, Jan 2015, pp. 328–333.

[94] W. Wen, M. Mao, H. Li, Y. Chen, Y. Pei, and N. Ge, "A holistic tri-region mlc stt-ram design with combined performance, energy, and reliability optimizations," in *2016 Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2016, pp. 1285–1290.

[95] S. Seyedzadeh, A. Jones, and R. Melhem, "Enabling fine-grain restricted coset coding through word-level compression for pcm," in *2018 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, Feb 2018, pp. 350–361.

[96] H. G. Lee, S. Baek, J. Kim, and C. Nicopoulos, "A compression-based hybrid mlc/slc management technique for phase-change memory systems," in *2012 IEEE Computer Society Annual Symposium on VLSI*, Aug 2012, pp. 386–391.

[97] M. Hoseinzadeh, M. Arjomand, and H. Sarbazi-Azad, "Reducing access latency of mlc pcms through line striping," in *2014 ACM/IEEE 41st International Symposium on Computer Architecture (ISCA)*, June 2014, pp. 277–288.

[98] L. Jiang, Y. Zhang, B. R. Childers, and J. Yang, "Fpb: Fine-grained power budgeting to improve write throughput of multi-level cell phase change memory," in *Proceedings of the 2012 45th Annual IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO-45. Washington, DC, USA: IEEE Computer Society, 2012, pp. 1–12. [Online]. Available: https://doi.org/10.1109/MICRO.2012.10

[99] M. Awasthi, M. Shevgoor, K. Sudan, B. Rajendran, R. Balasubramonian, and V. Srinivasan, "Efficient scrub mechanisms for error-prone emerging memories," in *IEEE International Symposium on High-Performance Comp Architecture*, Feb 2012, pp. 1–12.

[100] S. Shirinzadeh, M. Soeken, P. Gaillardon, and R. Drechsler, "Fast logic synthesis for rram-based in-memory computing using majority-inverter graphs," in *2016 Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2016, pp. 948–953.

[101] P. Wang, S. Li, G. Sun, X. Wang, Y. Chen, H. Li, J. Cong, N. Xiao, and T. Zhang, "Rc-nvm: Enabling symmetric row and column memory accesses for in-memory databases," in *2018 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, Feb 2018, pp. 518–530.

[102] M. Imani, Y. Cheng, and T. Rosing, "Processing acceleration with resistive memory-based computation," in *Proceedings of the Second International Symposium on Memory Systems*, ser. MEMSYS '16. New York, NY, USA: ACM, 2016, pp. 208–210. [Online]. Available: http://doi.acm.org/10.1145/2989081.2989086

[103] D. Fey, M. Reichenbach, C. S{o}ll, M. Biglari, J. R{o}ber, and R. Weigel, "Using memristor technology for multi-value registers in signed-digit arithmetic circuits," in *Proceedings of the Second International Symposium on Memory Systems*, ser. MEMSYS

'16. New York, NY, USA: ACM, 2016, pp. 442–454. [Online]. Available: http://doi.acm.org/10.1145/2989081.2989124

[104] D. Bhattacharjee, R. Devadoss, and A. Chattopadhyay, "Revamp: Reram based vliw architecture for in-memory computing," in *Design, Automation Test in Europe Conference Exhibition (DATE), 2017*, March 2017, pp. 782–787.

[105] L. Chen, J. Li, Y. Chen, Q. Deng, J. Shen, X. Liang, and L. Jiang, "Accelerator-friendly neural-network training: Learning variations and defects in rram crossbar," in *Design, Automation Test in Europe Conference Exhibition (DATE), 2017*, March 2017, pp. 19–24.

[106] D. Fujiki, S. Mahlke, and R. Das, "In-memory data parallel processor," in *Proceedings of the Twenty-Third International Conference on Architectural Support for Programming Languages and Operating Systems*, ser. ASPLOS '18. New York, NY, USA: ACM, 2018, pp. 1–14. [Online]. Available: http://doi.acm.org/10.1145/3173162.3173171

[107] D. Bhattacharjee, L. Amau, and A. Chattopadhyay, "Technology-aware logic synthesis for reram based in-memory computing," in *2018 Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2018, pp. 1435–1440.

[108] T. Huang, G. Dai, Y. Wang, and H. Yang, "Hyve: Hybrid vertex-edge memory hierarchy for energy-efficient graph processing," in *2018 Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2018, pp. 973–978.

[109] B. Li, L. Song, F. Chen, X. Qian, Y. Chen, and H. H. Li, "Reram-based accelerator for deep learning," in *2018 Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2018, pp. 815–820.

[110] K. Qiu, W. Chen, Y. Xu, L. Xia, Y. Wang, and Z. Shao, "A peripheral circuit reuse structure integrated with a retimed data flow for low power rram crossbar-based cnn," in *2018 Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2018, pp. 1057–1062.

[111] M. Imani, S. Gupta, and T. Rosing, "Genpim: Generalized processing in-memory to accelerate data intensive applications," in *2018 Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2018, pp. 1155–1158.

[112] Y. Ni, W. Chen, W. Cui, Y. Zhou, and K. Qiu, "Power optimization through peripheral circuit reusing integrated with loop tiling for rram crossbar-based cnn," in *2018 Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2018, pp. 1183–1186.

[113] Y. Xiao, S. Nazarian, and P. Bogdan, "Prometheus: Processing-in-memory heterogeneous architecture design from a multi-layer network theoretic strategy," in *2018 Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2018, pp. 1387–1392.

[114] H. Zhao and J. Zhao, "Leveraging mlc stt-ram for energy-efficient cnn training," in *Proceedings of the International Symposium on Memory Systems*, ser. MEMSYS '18. New York, NY, USA: ACM, 2018, pp. 279–290. [Online]. Available: http://doi.acm.org/10.1145/3240302.3240422

[115] N. Binkert, B. Beckmann, G. Black, S. K. Reinhardt, A. Saidi, A. Basu, J. Hestness, D. R. Hower, T. Krishna, S. Sardashti, R. Sen, K. Sewell, M. Shoaib, N. Vaish, M. D. Hill, and D. A. Wood, "The gem5 simulator," *SIGARCH Comput. Archit. News*, vol. 39, no. 2, pp. 1–7, Aug. 2011. [Online]. Available: http://doi.acm.org/10.1145/2024716.2024718

[116] X. Dong, C. Xu, Y. Xie, and N. P. Jouppi, "Nvsim: A circuit-level performance, energy, and area model for emerging nonvolatile memory," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 31, no. 7, pp. 994–1007, July 2012.

[117] M. Poremba and Y. Xie, "Nvmain: An architectural-level main memory simulator for emerging non-volatile memories," in *2012 IEEE Computer Society Annual Symposium on VLSI*, Aug 2012, pp. 392–397.

[118] M. Poremba, T. Zhang, and Y. Xie, "Nvmain 2.0: A user-friendly memory simulator to model (non-)volatile memory systems," *IEEE Computer Architecture Letters*, vol. 14, no. 2, pp. 140–143, July 2015.

[119] S. J. E. Wilton and N. P. Jouppi, "Cacti: an enhanced cache access and cycle time model," *IEEE Journal of Solid-State Circuits*, vol. 31, no. 5, pp. 677–688, May 1996.

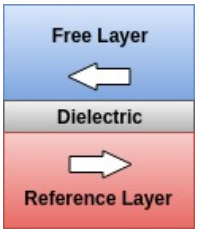
[120] S. Thoziyoor, N. Muralimanohar, J. H. Ahn, and N. P. Jouppi, "Cacti 5.1," Technical Report HPL-2008-20, HP Labs, Tech. Rep., 2008.

[121] I. Newsroom, "Intel and micron produce breakthrough memory technology, july 28, 2015."

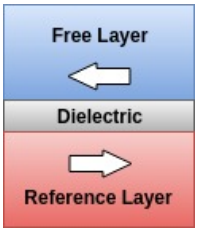
[122] A. Allan, D. Edenfeld, W. H. Joyner, Jr., A. B. Kahng, M. Rodgers, and Y. Zorian, "2001 technology roadmap for semiconductors," *Computer*, vol. 35, no. 1, pp. 42–53, Jan. 2002. [Online]. Available: https://doi.org/10.1109/2.976918

[123] W. M. Arden, "The international technology roadmap for semiconductorsperspectives and challenges for the next 15 years," *Current Opinion in Solid State and Materials Science*, vol. 6, no. 5, pp. 371 – 377, 2002. [Online]. Available: https://doi.org/10.1016/S1359-0286(02)00116-X

Aditya K. Kamath was a B. Tech student at the National Institute of Technology Karnataka, Surathkal. His research interests are in Memory Architecture and Performance Modelling. He is a student member of the IEEE.



Leslie Monis Biography text here.



A Tarun Karthik Biography text here.

Basavaraj Talawar received the PhD from the ECE department at the Indian Institute of Science, Bangalore. He is an Assistant Professor at the Computer Science and Engineering department in the National Institute of Technology Karnataka, Surathkal. His research interests are in Network-on-Chips, Simulation Acceleration and Parallelization. He is a member of the IEEE.