SELCC: Enhancing MLC Reliability and Endurance with Single-cell Error Correction Codes

Yujin Lim*, Dongwhee Kim†, and Jungrae Kim‡
*Dept. of Electrical and Computer Engineering**, *Dept. of Semiconductor and Display Engineering†*,
Dept. of Semiconductor Systems Engineering‡, *Sungkyunkwan University*
Email: *dbwls1229@skku.edu, †xyz123479@skku.edu, ‡dale40@skku.edu

***Abstract*—Conventional DRAM's limitations in volatility, high static power consumption, and scalability have led to the exploration of alternative technologies such as Phase Change Memory (PCM) and Resistive RAM (ReRAM). Storage-Class Memory (SCM) arises as a target application for these emerging technologies with non-volatility and higher capacity through Multi-Level Cells (MLCs). However, MLCs face issues of reliability and reduced endurance. To address this, our paper introduces a novel Error Correction Codes (ECC) method, "Single Eight-Level Cell Correcting" (SELCC) ECC. This technique efficiently corrects single-cell errors in 8-level cell memories using existing ECC syndromes without added redundancy. SELCC enhances memory reliability and improves 8LC memory endurance by 3.2 times, surpassing previous solutions without significant overheads.**

***Index Terms*—Storage-Class Memory, Multi-Level Cell, Error Correction Codes, Reliability, Endurance**

I. Introduction

Dynamic Random Access Memory (DRAM) has long been the predominant type of main memory in computing devices, owing to its fast speed and well-established manufacturing process. However, despite these advantages, DRAM comes with inherent limitations; volatility (data is lost when power is turned off), static power consumption (continuous refresh operations are required), and limited scalability (process shrinkage struggles around the 10nm mark). In response to these challenges, alternative memory technologies, such as Phase Change Memory (PCM) and Resistive RAM (ReRAM), have either emerged or are currently under intensive research. While completely replacing DRAM might not be feasible due to economic factors, these technologies offer attractive options for contemporary and future computer systems.

Storage-Class Memory (SCM) stands as the most promising use case for these alternative memories [1]. With the non-volatile nature of these memories, SCM can bridge the gap between fast-but-volatile DRAM-based main memory and slower-but-non-volatile storage. Moreover, these memories can provide higher density than DRAM by storing multiple bits of information per cell (i.e., Multi-Level Cell, or MLC) and using more refined process technologies. Therefore, placing SCM just beneath the main memory in the memory hierarchy can boost the system performance with its high capacity, despite its relatively longer latency.

However, utilizing these emerging memories as SCM poses several challenges, particularly in terms of reliability and endurance. Altering a cell's phase or resistance during writes induces mechanical stress, which limits the number of write cycles a cell can endure before wearing out. The IRDS has projected that PCM-based 3D XPoint and ReRAM can endure only 10^9 write cycles even in 2034 [2]. This number is considerably less than DRAM, which can sustain more than 10^{15} write cycles [3]. While traditional storage systems can use techniques such as wear-leveling to evenly distribute writes among cells, the stringent latency requirements of SCM make incorporating an address translation layer challenging, potentially further compromising its lifespan due to uneven write patterns.

Reliability in emerging memories is an another critical issue. In PCM, for instance, drifts in a cell's resistance can compromise data integrity over time [4]. Furthermore, write operations might inadvertently change the resistance of adjacent cells, especially as cell density increases [1]. The error rate escalates with MLCs, where the resistance range is subdivided to accommodate multiple bits within a single cell [5], [6].

To mitigate these endurance and reliability issues, Error Correction Codes (ECC) are extensively employed in SCMs. By leveraging additional cells for redundancy, ECC provides a mechanism to recover data from physical defects such as wear-out, resistance drift, and write disturbance. In traditional Single-Level Cell (SLC) memories, these defects typically result in single-bit errors, and the conventional Single Error Correcting - Double Error Detecting (SEC-DED) ECC could suffice to enhance both endurance and reliability. However, with MLCs, the subdivided resistance levels elevate the likelihood of errors, and defects at the cell level can impact multiple bits simultaneously. This means that MLC memories necessitate ECC with a stronger correction capability compared to SLC memories.

Numerous researches have proposed ECC schemes for MLC memories. Some suggest applying binary BCH codes, which can correct multi-bit errors independently [7]. However, their stronger correction capabilities come with the trade-off of increased redundancy (e.g., 21.9% redundancy for Double Error Correcting (DEC) over 64-bit data vs. 12.5% redundancy

This work was supported in part by the Technology Innovation Program (RS-2023-00237136, Development of CXL/DDR5-based memory subsystem for AI accelerators, 40%) funded by the Ministry of Trade, Industry & Energy(MOTIE, Korea)(1415187686) and in part by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2021-0-00863, Intelligent in-memory error-correction device for high-reliability memory, 30%) and in part by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.RS-2023-00228970, Development of Flexible SW·HW Conjunctive Solution for On-edge Self-supervised Learning, 30%).

979-8-3503-4859-0/DATE24/© 2024 EDAA

for SEC-DED). Others have explored Gray coding of resistance levels within a cell, ensuring that a drift error between adjacent levels only affect a single bit. While this approach can enhance the initial product reliability, its efficacy diminishes over the product's lifespan, especially as worn-out MLCs start exhibiting multi-bit errors.

This paper introduces a novel ECC solution tailored for MLC memories with 8-Level Cells (8LCs), commonly referred to as Triple-Level Cells (TLCs). Each 8LC embeds 3 bits of information by leveraging eight distinct levels. Our approach, named the *Single Eight-Level Cell Correcting (SELCC)* ECC, utilizes the same redundancy configuration as the traditional SEC-DED, which consists of a 64-bit data plus 8-bit redundancy. Notably, SELCC is capable of correcting all single-cell errors (i.e., up to 3-bit errors in a cell), whether arising from a slight drift in resistance or a complete cell wear-out.

To boost error correction without introducing extra redundancy, SELCC leverages the unused syndromes in SEC-DED. SEC-DED with 8-bit redundancy uses only 72 of its 255 nonzero syndromes for single-bit error correction, leaving 119 syndromes unused for correction. SELCC repurposes these unused syndromes and maps all single-cell error patterns (168 patterns from 7 patterns per cell and 24 cells in 72-bit ECC word) to distinct non-zero syndromes. These unique syndromes facilitate precise error identification and subsequent restoration.

With its ability to correct single-cell errors, SELCC can significantly enhance both the reliability and endurance of 8LC memories. While previous reliability solutions with similar redundancy could only address errors of limited magnitudes, SELCC can recover data even when a cell faces an unlimited resistance drift. Additionally, it can prolong the lifetime of MLC memories without compromising performance, a notable advantage over prior endurance solutions based on redirection, which might degrade performance.

The main contributions of this paper are as follows:

- We propose SELCC, a strong and efficient ECC scheme specifically designed for 8LC memories.
- We detail the construction process of SELCC, leveraging previously unused syndromes to correct all single 8LC errors.
- Through our evaluation of SELCC, we demonstrate this method can significantly enhance both the reliability and endurance of 8LC memories, surpassing prior solutions with same redundancy and comparable overheads.

II. Background

This section provides background knowledge for understanding SELCC by describing MLC memory errors, using PCM as an example, and ECC.

A. MLC Memory Errors

MLC errors originate from numerous physical phenomena. For clarity, we categorize these errors based on their magnitude: limited or unlimited.

1) Limited Magnitude: Errors in this category exhibit a maximum magnitude that is restricted, and does not span the full range of a cell's potential resistance. A representative example is errors from resistance drifts associated with Gray coding. As time progresses, the resistance of a PCM cell might rise due to the structural relaxation process [6]. Such a drift, albeit gradual, can shift a cell's resistance enough to move it to a higher level. With Gray coding, a binary numerical system in which two successive values differ by only one bit, a single-level shift results in just a single-bit error.

2) Unlimited Magnitude: Errors in this category can induce radical shifts in the cell's state, rendering it far from its original value. A notable example is errors from PCM wear-outs. Each time a PCM cell undergoes a phase transition, it experiences mechanical stress from heating and cooling. Continuous exposure to this stress gradually weakens the structural integrity of the cell material and can eventually result in a loss of the cell's connection to the electrode. When a cell wears out, its resistance transitions to the maximum level, regardless of the original level, causing errors of potentially unlimited magnitude within that cell.

B. Error Correction Codes

ECC can restore data from errors using redundancy. In linear block codes, redundant bits are algorithmically generated from the data using a *Generator Matrix* (G-matrix). A valid pair of $(n-r)$-bit data and r-bit redundancy is called an n-bit *codeword*. An error can convert a codeword into an invalid pair, known as a *non-codeword*. During decoding, ECC multiplies the word by the *Parity-Check Matrix* (a.k.a. H-matrix), and the result is termed as a *syndrome*. The G-matrix and H-matrix are algorithmically coupled, ensuring all codewords result in zero syndromes, whereas non-codewords yield non-zero syndromes.

The decoder utilizes the syndrome to correct errors. For instance, Hamming codes [8] employ an H-matrix where every individual single error pattern produces a unique non-zero syndrome. This distinctness allows for both error detection and correction. With an r-bit redundancy, there are $2^r - 1$ non-zero syndromes, allowing for the correction of up to $2^r - 1$ error patterns.

III. Prior Work

A. Reliability Enhancement Techniques

ECC is the predominant strategy for enhancing memory reliability. Many systems lean towards simpler bit-level or symbol-level correction schemes to satisfy their strict latency constraints. This section analyzes some of the most pertinent schemes.

1) Single Error Correcting - Double Error Detecting Codes: In the 1950s, Richard Hamming pioneered the Single Error Correcting (SEC) codes and the SEC-DED codes [8]. SEC codes, when equipped with 7-bit redundancy, are capable of identifying 127 unique error patterns. This allows for the correction of a Single Error (SE) across both 120-bit data and 7-bit redundancy. However, memories typically have a power-of-2 access granularity and these codes frequently undergo *shortening* to protect 64-bit data with 7-bit redundancy.

With SEC codes, Double Errors (DEs) can produce a syndrome identical to that of SEs. In that case, the DE is miscorrected as an SE, amplifying the error magnitude from two bits to three. To mitigate this, Hamming introduced an additional parity bit, ensuring that each DE generates a syndrome distinct from those of SEs. Although this addition does not provide the capability to correct DEs as certain DEs share identical syndromes, the derived SEC-DED codes can now reliably detect all DEs without any miscorrections. In memories, 8-bit redundancy is required to protect 64-bit data, leading to the conventional 72-bit ECC memory configuration.

2) Single Error Correcting - Double Adjacent Error Correcting (SEC-DAEC) Codes: SEC-DAEC codes are designed to correct both SEs and Double Adjacent Errors (DAEs). These codes can be particularly useful in situations where adjacent bit errors are prevalent. For instance, SEC-DAEC codes can perform particularly well in 4LC memories, which store 2 bits per cell. When these cells malfunction, they tend to exhibit SEs and DAEs, making SEC-DAEC codes an apt choice for error correction. However, they cannot correct MLC errors with more bits per cell. SEC-DAEC codes can be constructed by ensuring distinct syndromes for SEs and DAEs. [9] proposed efficient SEC-DAEC codes by utilizing the unused syndromes of 72-bit SEC-DED codes.

3) Single Error Correcting - Double Adjacent Error Correcting - Triple Adjacent Error Correcting (SEC-DAEC-TAEC) Codes: SEC-DAEC-TAEC codes are designed to correct SEs, DAEs, and Triple Adjacent Errors (TAEs). These codes can cover most of the single 8LC cell error patterns: SEs (001, 010, 100), DEs (011, 110), and TAE (111). However, they do not provide full coverage, as they overlook Double Non-adjacent Error (DNE) (101) in a cell.

4) IP-DAEC: Several studies have analyzed error patterns in MLC memories and designed ECCs to address those specific cases [10]–[12]. This approach can enhance the error correction capability of codes without adding extra parity bits or complicating the algorithms. IP-DAEC [12] exploits the observation that errors with a magnitude of 3 or less always affect one of the lowest and the second lowest bits in the 8LCs. The MSB bit does not flip on its own, which indicates that the MSB bit can only flip if there is an error in one of the two lower bits. IP-DAEC generates an Interleaved Parity (IP) bit by XORing every MSB bit across the cells and covers the remaining bits using SEC-DAEC codes. The IP bits assist in identifying errors on the MSB bits if an erroneous cell is located by the SEC-DAEC codes. Therefore, errors with a magnitude of 3 or less in a single cell can always be corrected through a combination of IP and SEC-DAEC codes. However, this mechanism may fall short in rectifying errors of unlimited magnitude that exceed a magnitude of 3.

Table I compares the correction capabilities of prior works with that of SELCC, which underscores the need for stronger cell-level ECC for 8LC memories.

TABLE I: Comparison of ECC schemes with equal redundancy ratios in correcting single 8LC error patterns.

Scheme	Redundancy ratio	Possible error patterns on a single cell			
		SE 	DAE 	DNE 	TAE 
SEC-DED	12.5%	O	X	X	X
SEC-DAEC	12.5%	O	O	X	X
SEC-DAEC-TAEC	12.5%	O	O	X	O
IP-DAEC	12.5%	X	O	O	O
SELCC (proposed)	12.5%	O	O	O	O

B. Endurance Enhancement Techniques

Academia and industry have proposed various methods to prevent and repair memory wear-outs. Wear-leveling, a common prevention mechanism in storage devices, distributes write operations evenly across cells through address translation. This technique extends the lifespan of a device by preventing a few frequently updated cells from dominating its wear and tear. However, the inclusion of this additional translation layer can increase latency, making it less suitable for SCM.

Another line of research focuses on repair strategies. Detecting cell wear-outs is relatively simple by performing a verifying read after a write. Several solutions, including FREE-p [13], ECP [4], and SAFER [14], repair devices by remapping the faulty cell to a spare one. These schemes require extra storage for pointers and spares, yet the system still requires ECC with separate redundancy since they cannot address errors that arise after writes (e.g., resistance drift). Moreover, accessing a faulty region requires additional access to the spare, slowing down performance as cells wear out over time. In contrast, SELCC presents a holistic solution addressing both reliability and endurance concerns, eliminating the need for separate redundancy and access.

IV. SELCC ECC

We propose a novel ECC scheme, SELCC, tailored to correct every single-cell error in 8LC memories. Despite its enhanced correction capability, SELCC does not require additional redundancy beyond what the conventional SEC-DED requires.

SELCC achieves this efficiency by leveraging unused syndromes of shortened codes and error boundaries. Conventional SEC-DED codes employ only 72 out of the possible 255 nonzero syndromes to correct all SEs across a 72-bit codeword span. SEC-DAEC codes further harness another 71 from the remaining 183 syndromes to correct 71 DAEs. Similarly, SEC-DAEC-TAEC utilizes another 70 from the residual 112 syndromes, enabling the correction of an additional 70 TAEs. This approach based on error adjacency, however, faces a limitation as the leftover syndromes are insufficient to correct DNEs.

Instead, SELCC focuses on error boundaries to reduce the number of targeted error patterns. In memory systems, it is rare for adjacent errors to cross a cell boundary, making them prime candidates for exclusion from the targeted error pool. Within a 72-bit word, there exist 24 8LC cells, and each of these cells can manifest 7 distinct error patterns, ranging from 000 through 111. Cumulatively, this leads to a collective count of 168 errors, all neatly contained within cell boundaries. This count aptly fits within the constraint of 8-bit redundancy.

The challenge then lies in devising a suitable mapping for these 168 errors onto unique syndromes, setting the stage for effective error correction. The subsequent subsections delve

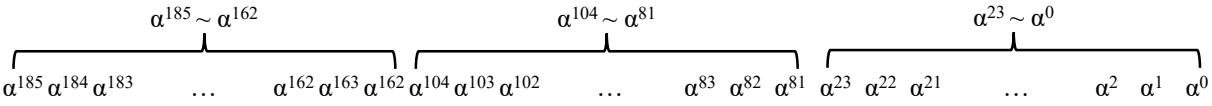

Fig. 1: An example of (72,64) SELCC H-matrix in multiplicative form.

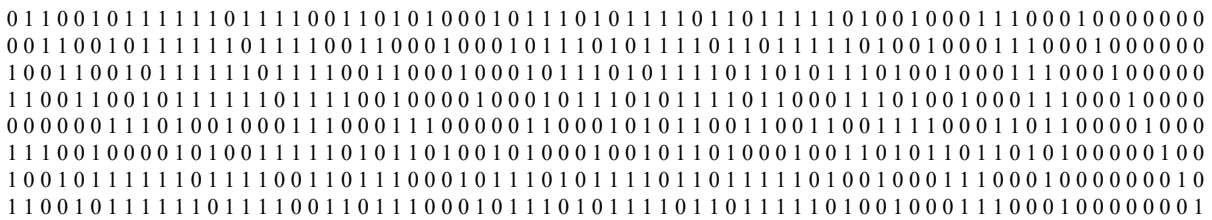

Fig. 2: An example of (72,64) SELCC H-matrix in binary form.

into this in detail, outlining the required properties for such codes, introducing a construction algorithm to find these codes, and describing their hardware implementation.

A. Code Properties

In an H-matrix, each column represents the syndrome generated when an SE arises at the corresponding bit position. Likewise, the syndrome resulting from a multi-bit error is the sum (XOR) of the columns from the corresponding bit positions. Consequently, the H-matrix of SELCC should adhere to the following properties to correct all single-cell errors.

1) Every column must be nonzero.
2) Each column must be unique.
3) The sum of two or three columns within any cell boundaries must be nonzero and unique.

The first and second properties guarantee that syndromes arising from any single error are distinct. In the third criterion, each trio of columns represents a cell, and the property ensures that syndromes from double or triple errors within a cell are unique.

B. Code Construction

We employ GF(2^8) arithmetic to derive an H-matrix that satisfies the properties. GF(2^8), a Galois Field with 256 elements, has a primitive element symbolized as α. This field is closed under multiplication and addition, and the element values range from 0, α^0 to α^{254}. We treat every 8-bit column within the H-matrix as an element of GF(2^8).

The construction initiates by populating the rightmost 8 columns of the H-matrix with descending powers of α, from α^7 down to α^0. This ensures the codes are systematic, meaning the data portion of the codeword is a reflection of the original input data, and it simultaneously aligns with 8LC boundaries.

As a consequence, double or triple errors from the rightmost cell yield syndromes of $\alpha^0+\alpha^1$ (1st DAE), $\alpha^1+\alpha^2$ (2nd DAE), $\alpha^0+\alpha^2$ (DNE), and $\alpha^0+\alpha^1+\alpha^2$ (TAE). In a similar manner, the next rightmost cell produces syndromes of $\alpha^3+\alpha^4 = \alpha^3(\alpha^0+\alpha^1)$, $\alpha^3(\alpha^1+\alpha^2)$, $\alpha^3(\alpha^0+\alpha^2)$, and $\alpha^3(\alpha^0+\alpha^1+\alpha^2)$. Generalizing this, assigning descending powers of α to n cells, ranging from α^{3n-1} to α^0, will lead to syndromes of $\{\alpha^{n-1}, \dots, \alpha^0\}(\alpha^0+\alpha^1)$ for the 1st DAEs, $\{\alpha^{n-1}, \dots, \alpha^0\}(\alpha^1+\alpha^2)$ for the 2nd DAEs, and so forth.

The next step is to allocate these sequences, ensuring they do not intertwine, thereby preserving the uniqueness of syndrome values. After iterating over the 16 generator polynomials of GF(2^8) and considering different lengths of allocation, we identified several H-matrices that meet all SELCC properties. An illustrative H-matrix, for instance, employs the generator polynomial 0x11D with a cell-run-length of 8, as visualized in Fig. 1 (in multiplicative) and Fig. 2 (in binary).

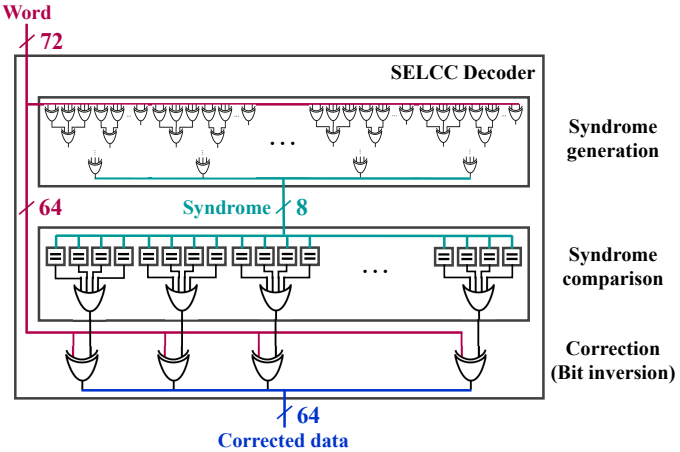

Fig. 3: An implementation of (72,64) SELCC decoder.

C. Hardware Implementation

The encoding process of SELCC shares that of the conventional SEC-DED. By multiplying a 64-bit data vector with a (64×72) G-matrix, a (1×72) codeword is produced. In the binary domain, addition and multiplication operations translate to XOR and AND operations, respectively. As a result, each redundancy bit is constructed via an XOR gate tree, with contributions determined by the G-matrix [15].

Decoding is slightly more intricate, as shown in Fig. 3. The decoder multiplies a 72-bit word with the (72×8) transposed H-matrix to yield a (1×8) syndrome, using XOR gate trees. This syndrome is then distributed to the output data bits, which compare the syndrome against its correctable syndromes. An output bit in SEC-DED has only one correctable syndrome (SE at the current position, P), whereas one in SELCC has up to 4 correctable syndromes; one SE at P, two DEs at (P, one of the two other positions within the cell), and one TAE (all positions within the cell). If the syndrome matches one of these correctable syndromes, the final stage flips the output bit using an XOR gate, thereby correcting the error.

In comparison to SEC-DED, each output bit in SELCC mandates three additional comparators and a 4-input OR gate

TABLE II: Correction capabilities against single 8LC errors (higher is better) with existing ECC schemes and SELCC. SDC probabilities (lower is better) are noted in parentheses.

Error scenario	SEC-DED	SEC-DAEC	SEC-DAEC-TAEC	IP-DAEC	SELCC (proposed)
SE	100% (0%)			67.97% (0%)	100% (0%)
DE	0% (0%)	66.69% (16.69%)	66.63% (29.21%)	100% (0%)	100% (0%)
TAE	0% (0%)	0% (66.62%)	100% (0%)		100% (0%)

to aggregate comparison results. By parallelizing comparisons, the OR gate becomes the primary source of the latency increment. Section V-C presents a detailed evaluation regarding latency, area, and power consumption.

V. Evaluation

A. Reliability

To quantify the correction and detection capability of an ECC scheme, we employed a Monte-Carlo simulation. The simulation injects errors at random positions within an ECC word with 64-bit data. These errors may manifest as a single error within a cell (labeled SE), double errors within a cell (labeled DE, covering both DAEs and DNEs), or a triple adjacent error within a cell (labeled TAE). The simulation introduces up to 2 such errors to measure the correction capability for single-cell errors and the detection capability for double-cell errors.

Following error injection, ECC decoding is applied to determine if the error is a *Correctable Error (CE)*, a *Detectable-but-Uncorrectable Error (DUE)*, or if it results in *Silent Data Corruption (SDC)*. For comparison, we select four state-of-the-art ECC schemes: SEC-DED [16], SEC-DAEC [9], SEC-DAEC-TAEC [17], and IP-DAEC [12]. All these schemes have the same redundancy configuration as SELCC in 8LC memories (i.e., 64-bit data + 8-bit redundancy). For our simulation, we sourced the H-matrices from the respective cited papers. Each error scenario underwent 1 million simulation iterations.

Table II presents a comparison of the correction capabilities against single-cell errors. All the evaluated schemes successfully correct SEs — with the exception of IP-DAEC which managed to correct $2/3$ of such errors. When examining DEs, the SEC-DED scheme exhibits 0% correction capability. In contrast, both SEC-DAEC and SEC-DAEC-TAEC demonstrate a $2/3$ correction ratio (addressing errors of 011 and 110), but both failed to correct the 101 error pattern. IP-DAEC shows a 100% correction capability for DEs, despite its weaker protection against SEs. In terms of TAEs, only the SEC-DAEC-TAEC and IP-DAEC schemes are capable of correcting these highly severe errors. Meanwhile, SELCC can correct any single-cell errors, regardless of the severity, using the same redundancy.

Table III compares the detection capabilities against double cell errors. A prevalent theme in ECC design is the inevitable trade-off between correction and detection capabilities. As more syndromes are allocated for correction, the likelihood that syndromes associated with severe errors match one of these correction syndromes increases. This can diminish an ECC's ability to detect certain error patterns. This concept is clearly illustrated in the table; SEC-DED with 72 correction syndromes and SEC-DAEC with 143 correction syndromes generally demonstrate lower SDC probabilities (i.e., better detection). Similarly, SEC-DAEC-TAEC, with 213 correction syndromes, shows higher SDC probabilities than SELCC, which utilizes 168 correction syndromes. A noteworthy outlier is an IP-DAEC. Despite dedicating only 76 syndromes to correction, it exhibits the second-highest SDC ratios. This irregularity stems from its two-tiered protection strategy, which allots 1-bit redundancy for the MSB parity and 7-bit redundancy for (kind of) SEC-DAEC.

TABLE III: SDC probabilities against double 8LC errors (lower is better) with existing ECC schemes and SELCC.

Error scenario	SEC-DED	SEC-DAEC	SEC-DAEC-TAEC	IP-DAEC	SELCC (proposed)
SE+SE	0%	54.19%	83.91%	81.32%	65.48%
SE+DE	50.55%	54.38%	83.27%	71.88%	65.50%
SE+TAE	0%	55.63%	83.67%	71.74%	66.85%
DE+DE	0.97%	53.63%	83.57%	58.38%	66.40%
DE+TAE	71.98%	56.15%	82.40%	58.17%	62.74%
TAE+TAE	12.69%	55.45%	81.52%	57.32%	61.56%

Although it is not shown in the table, both SEC-DAEC and SEC-DAEC-TAEC can correct a small portion of two-cell errors, of which these cells are nearby and exhibit adjacent bit errors. For example, SE+SE errors in adjacent bit positions account for 0.9% of such errors, which these schemes can correct. Likewise, SEC-DAEC-TAEC can address 0.9% of SE+DE errors spanning adjacent cell borders. Nevertheless, this marginal error correction capability for double-cell errors does not overshadow their weak correction capability against the more frequent single-cell errors [18].

B. Endurance

In assessing endurance enhancement, we assume that each MLC can withstand 10^9 write cycles on average [2]. However, this average is nuanced by two distinct variations: *cell variation*, where each cell can tolerate a different number of write cycles due to intrinsic process variations, and *usage variation*, where each cell experiences a different number of write cycles from the software.

Cell variation is characterized using a *Coefficient Of Variation (COV)*. In statistics, a COV is calculated as the standard deviation (σ) divided by the mean (μ). [3] reported that cell variations in PCM follow normal distributions and used COVs ranging between 10% and 30%. For our evaluation, we take a midpoint value, setting the COV at 20%.

To achieve an operational reliability where 99.99% of 72-bit words have no defective cells, each of the 24 8LCs should possess a wear-out likelihood of less than 0.0004%. Interpreting this probability within a normal distribution context indicates a deviation of -4.45σ from the mean. Given our COV of 20%, the wear-out threshold stands at approximately 1.1×10^8 write cycles. Surpassing this threshold increases the risk of cell wear-out, which subsequently raises the chances of a faulty word over the acceptable 0.01%.

SELCC can tolerate higher probabilities of cell wear-outs by allowing up to one worn-out cell per word. Through binomial

TABLE IV: A comparison of hardware overheads.

		SEC-DED	SEC-DAEC	SEC-DAEC-TAEC	IP-DAEC	SELCC
Encoder	Latency (ns)	0.40	0.40	0.40	0.40	0.40
	Area (um^2)	115	135	103	95	147
	Power (mW)	0.21	0.25	0.18	0.18	0.29
Decoder	Latency (ns)	0.43	0.49	0.48	0.46	0.52
	Area (um^2)	1020	1841	2589	1684	2704
	Power (mW)	2.28	6.63	10.68	6.71	10.60

distribution calculations, SELCC can withstand a cell failure rate of up to 0.06%. This corresponds to a -3.24σ shift from the mean in normal distribution. With the aforementioned COV of 20%, this flexibility enables each cell to handle up to approximately 3.52×10^8 writes. In comparison to a system with zero wear-out tolerance, this represents a significant $3.2\times$ increase in endurance.

Moreover, considering variations in usage patterns can also lead to further improvements in endurance. An uneven distribution of writes, with some cells experiencing excessive writes, can considerably shorten memory lifetime. SELCC could mitigate this degradation by accommodating some worn-out cells. Nevertheless, given that usage variation fluctuates significantly between different applications and memory configurations, our evaluation opts not to factor this in.

C. Hardware Overheads

Despite the stronger correction capability, SELCC incurs similar hardware overheads to existing ECCs. To assess latency, area, and power consumption overheads, we implemented SystemVerilog models for SELCC encoder and decoder. We also implemented models for the state-of-the-art schemes to demonstrate that SELCC does not increase the overheads significantly. We synthesized the models using Synopsys Design Compiler and UMC 28nm SVT cells. The target clock frequency is set to 1.5GHz, with a 40% margin for clock uncertainty and wire delays. This leaves a 0.40ns budget for the encoding/decoding. Table IV presents the results.

1) Latency: Every encoder meets the timing constraint, while all decoders fail and reveal their critical delays. SELCC shows a slight $0.09ns$ increment in the critical path compared to SEC-DED due to its aggregating OR gate. Still, the decoder can operate at an impressive frequency of 1.5GHz, even after factoring in the 40% uncertainty. By leveraging advancements in current and upcoming process technologies, this latency increase can be further mitigated, ensuring SELCC's competitiveness in SCMs.

2) Area: SELCC displays a higher increase in area compared to latency. This can be ascribed to the fact that the area directly reflects the additional comparators needed for syndrome matching, whereas latency can often hide most of the overheads through parallelization. However, the overall die size of modern processors are substantial in comparison to this area overhead. For instance, SELCC accounts for a negligible $< 0.003\%$ of advanced CPUs which span hundreds of mm^2 in silicon footprint. This estimation, based on 28nm process, will further shrink in newer and future process nodes.

3) Power Consumption: SELCC consumes more power than SEC-DED but is on par with one of the previous works (SEC-DAEC-TAEC). Yet, this increase takes a negligible portion in modern processors, which consume hundreds of Watts.

VI. Conclusion

We introduced a novel ECC, termed SELCC, tailored to enhance the reliability and longevity of 8LC SCMs. While MLC memories bring forth the benefits of non-volatility and higher capacity, their integration as SCMs raises concerns about reliability and endurance due to inevitable resistance drifts and cell wear-outs. Prior ECC solutions with similar overheads fell short of addressing all single-cell errors, mandating a separate mechanism to enhance SCM lifetimes. SELCC significantly improves both reliability and endurance (by 3.2 times) by correcting all single-cell errors. This enhancement is achieved without the need for additional redundancy.

References

[1] S. Rashidi, M. Jalili, and H. Sarbazi-Azad, "A Survey on PCM Lifetime Enhancement Schemes," *ACM Comput. Surv.*, vol. 52, 2019.

[2] "More Moore - International Roadmap For Devices and Systems (IRDS)." https://irds.ieee.org/editions/2022/more-moore.

[3] M. K. Qureshi, "Pay-As-You-Go: Low-overhead Hard-error Correction for Phase Change Memories," in *the 44th MICRO*, 2011.

[4] S. Schechter, G. H. Loh, K. Strauss, and D. Burger, "Use ECP, Not ECC, for Hard Failures in Resistive Memories," in *the 37th ISCA*, 2010.

[5] N. H. Seong, S. Yeo, and H.-H. S. Lee, "Tri-Level-Cell Phase Change Memory: Toward an Efficient and Reliable Memory System," in *the 40th ISCA*, 2013.

[6] W. Zhang and T. Li, "Helmet: A Resistance Drift Resilient Architecture for Multi-level Cell Phase Change Memory System," in *the 41st DSN*, 2011.

[7] R. C. Bose and D. K. Ray-Chaudhuri, "On a Class of Error Correcting Binary Group Codes," *Information and Control*, vol. 3, 1960.

[8] R. W. Hamming, "Error Detecting and Error Correcting Codes," *The Bell System Technical Journal*, 1950.

[9] S. Tripathi, J. Jana, J. Samanta, A. Anand, C. Kumar, and G. Raj, "FPGA and ASIC Implementation of SEC-DED-DAEC Codes for SRAM Applications," in *the 2nd ICCDC*, 2019.

[10] A. Das and N. A. Touba, "Efficient One-Step Decodable Limited Magnitude Error Correcting Codes for Multilevel Cell Main Memories," *IEEE Transactions on Nanotechnology*, vol. 18, 2019.

[11] A. Das, A. Sánchez-Macián, F. García-Herrero, N. A. Touba, and J. A. Maestro, "Enhanced Limited Magnitude Error Correcting Codes for Multilevel Cell Main Memories," *IEEE Transactions on Nanotechnology*, vol. 18, 2019.

[12] S. Liu, P. Reviriego, and F. Lombardi, "Codes for Limited Magnitude Error Correction in Multilevel Cell Memories," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, 2020.

[13] D. H. Yoon, N. Muralimanohar, J. Chang, P. Ranganathan, N. Jouppi, and M. Erez, "FREE-p: Protecting Non-volatile Memory against Both Hard and Soft Errors," in *the 17th HPCA*, 2011.

[14] N. H. Seong, D. H. Woo, V. Srinivasan, J. A. Rivers, and H.-H. S. Lee, "SAFER: Stuck-At-Fault Error Recovery for Memories," in *the 43rd MICRO*, 2010.

[15] Y. Song, S. Park, M. B. Sullivan, and J. Kim, "SEC-BADAEC: An Efficient ECC with No Vacancy for Strong Memory Protection," *IEEE Access*, vol. 10, 2022.

[16] J. A. Maestro, P. Reviriego, L. Xiao, S. Liu, and A. Sanchez-Macian, "Odd-weight-column SEC–DED–TAED Codes," *Electronics Letters*, vol. 52, 2016.

[17] L.-J. Saiz-Adalid, P. Reviriego, P. Gil, S. Pontarelli, and J. A. Maestro, "MCU Tolerance in SRAMs Through Low-Redundancy Triple Adjacent Error Correction," *IEEE TVLSI*, vol. 23, 2015.

[18] N. An, R. Wang, Y. Gao, H. Yang, and D. Qian, "Balancing the Lifetime and Storage Overhead on Error Correction for Phase Change Memory," *PloS one*, vol. 10, 2015.