IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 9 August 2022, accepted 21 August 2022, date of publication 25 August 2022, date of current version 31 August 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3201525

APPLIED RESEARCH

SEC-BADAEC: An Efficient ECC With No Vacancy for Strong Memory Protection

YUSEOK SONG[1], SANGJAE PARK[1], MICHAEL B. SULLIVAN[2], (Member, IEEE), AND JUNGRAE KIM[3], (Senior Member, IEEE)

[1]Department of Electrical and Computer Engineering, Sungkyunkwan University, Suwon 16419, South Korea
[2]NVIDIA, Santa Clara, CA 95051, USA
[3]Department of Semiconductor Systems Engineering, Sungkyunkwan University, Suwon 16419, South Korea

Corresponding author: Jungrae Kim (dale40@skku.edu)

This work was supported in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) Grant by the Korean Government through the Ministry of Science and ICT (MSIT) (Intelligent in-memory error-correction device for high-reliability memory, 50% Artificial Intelligence Semiconductor based Multi Device Wireless Charger SoC Development, 30%) under Grant 2021-0-00863 and Grant 2021-0-00479, and in part by the National Research and Development Program through the National Research Foundation of Korea (NRF) by the MSIT (20%) under Grant 2020M3H2A1076786. The EDA tool was supported by the IC Design Education Center (IDEC), South Korea.

ABSTRACT Shrinking process technology and rising memory densities have made memories increasingly vulnerable to errors. Accordingly, DRAM vendors have introduced *On-die Error Correction Code (O-ECC)* to protect data against the growing number of errors. Current O-ECC provides weak *Single Error Correction (SEC)*, but future memories will require stronger protection as error rates rise. This paper proposes a novel ECC, called *Single Error Correction–Byte-Aligned Double Adjacent Error Correction (SEC-BADAEC)*, and its construction algorithm to improve memory reliability. SEC-BADAEC requires the same redundancy as SEC O-ECC, but it can also correct some frequent 2-bit error patterns. Our evaluation shows SEC-BADAEC can improve memory reliability by 23.5% and system-level reliability by 29.8% with negligible overheads.

INDEX TERMS Reliability, ECC, on-die ECC, DRAM, SEC, BADAEC.

I. INTRODUCTION

Dynamic Random Access Memory (DRAM) is the de-facto standard for main memory for its low cost and high density. Large-scale systems deploy tens of thousands of DRAM chips to provide tens of terabytes of memory or more. Meanwhile, memory devices are more vulnerable to errors than logic devices, which have increased sources of error masking [1], [2], [3]. A large number of chips and greater vulnerability have made DRAM one of the primary sources of failures—for instance, DRAM errors are the second most dominant source of hardware failures in [4]. The importance of DRAM failures has been reproduced by many other field studies and radiation-testing experiments [5], [6], [7], [8], [9].

System companies typically use Rank-level ECC (R-ECC) to protect DRAM against errors. R-ECC stores redundant information on extra chips in a DRAM DIMM (Dual In-line Memory Module) to correct errors. A class of strong R-ECCs, called Chipkill-correct, can restore data even if a chip in the rank fails completely, providing the required level of DRAM reliability to current systems [10].

However, with the continued process and voltage scaling, individual DRAM chips are becoming more vulnerable to errors. DRAM manufacturers have introduced *On-die ECC (O-ECC)* to protect DRAM against growing errors [11], [12], [13]. O-ECC adds extra cells and embeds encoding/decoding circuitry inside a chip to store redundancy and correct errors internally. O-ECC has not been standardized, yet most DRAM companies utilize 8-bit redundancy over 128-bit data to provide *Single-Error Correction (SEC)*. With SEC O-ECC, single-bit errors within an O-ECC block are corrected inside a DRAM chip, and a faulty chip can operate externally error-free to improve manufacturing yield. O-ECC also improves reliability by correcting errors from inherent faults and new faults during operation.

The associate editor coordinating the review of this manuscript and approving it for publication was Cristian Zambelli.

Although SEC O-ECC is efficient against random single-bit errors, it is insufficient to correct more severe errors. Field measurements show that multi-bit errors are already relatively common, accounting for 1−5% of the total errors [5], [6], [7], [8], [14]. In addition, shrinking process technology will make future DRAM chips more vulnerable to multi-bit errors. The critical charge to upset a DRAM cell keeps decreasing with cell capacitance reduction, potentially causing more *Multi-Bit Upsets (MBUs)* [15]. The smaller supply voltage and higher parasitic capacitance also make bit-lines more susceptible to crosstalk noise [16].

Maintaining the current reliability level with future DRAMs requires stronger O-ECC against multi-bit errors. However, increasing the ECC overheads is not desirable for the cost-sensitive DRAM vendors. Current SEC O-ECC utilizes 6.25% extra cells (8-bit redundancy over 128-bit data) and 6.9% overall chip area overhead [17]. Adding one more bit to the redundancy (8-bit to 9-bit) significantly increases the extra cell ratio to 12.5% and the overall chip overhead to 12.1% [17] because DRAM internal structures are organized as a multiple of 8 (Figure 1) [17], [18], [19], [20].

This paper proposes a novel ECC, called *Single Error Correcting - Byte-Aligned Double Adjacent Error Correcting (SEC-BADAEC)* to improve memory reliability with negligible costs. SEC-BADAEC utilizes the same ECC configuration as SEC O-ECC (i.e., 8-bit redundancy over 128-bit data) and provides stronger correction by correcting frequent multi-bit errors, as well. The SEC O-ECC is a shortened code and utilizes only 136 out of its 8-bit syndrome values to correct single-bit errors and leaves 119 non-zero syndrome values for detection only. SEC-BADAEC utilizes the unused 119 syndromes to correct 119 frequent multi-bit errors and leaves no vacancy for correction. By fully exploiting the syndrome space, SEC-BADAEC can provide stronger correction without increasing redundancy.

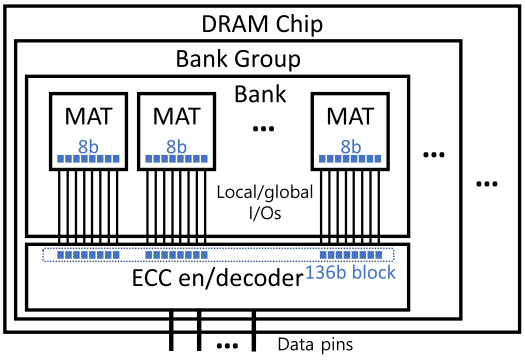


FIGURE 1. Overview of DRAM internal organization. DRAM collects 8 bits from individual MATs to build a 136-bit block and applies SEC O-ECC.

The 119 syndromes are used to correct *Byte-Aligned Double Adjacent Errors (BADAEs)*. *Double Adjacent Errors (DAEs)* are two-bit errors at adjacent positions, and BADAEs are DAEs that do not cross a byte boundary (Figure 2). There are 119 BADAEs on a 136-bit block (7 DAEs per byte and

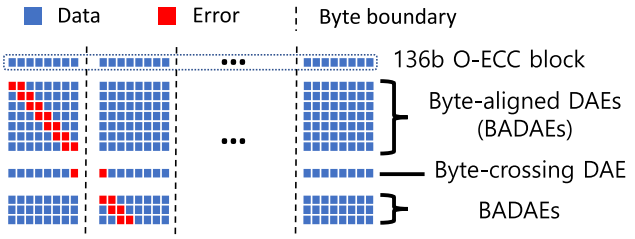


FIGURE 2. Double Adjacent Errors (DAEs) and Byte-Aligned DAEs (BADAEs) in DRAM. Byte-crossing DAEs are rare due to DRAM internal organization.

17 bytes per block). DAEs that cross a byte boundary are rare [9], [10], [14] because DRAM internally fetches 8-bit data from each cell array, called MAT, for routing and area efficiencies [17], [18], [19], [20]. Therefore, a severe MBU fault often generates errors in adjacent bit positions without crossing a byte boundary. A field measurement shows that 84 out of 85 multi-bit errors are confined to a single byte, and 26.3% of double errors are BADAEs [14]. The 26.3% ratio is significantly higher than expected from a random distribution of double errors (5.6%). By mapping the unused syndromes to the frequent BADAEs, SEC-BADAEC can provide efficient and strong protection for O-ECC.

SEC-BADAEC utilizes a novel code construction algorithm to map BADAEs into the syndromes that are not used by SEC ECC. It utilizes a divide-and-conquer approach based on the cyclic property of the Galois field to construct SEC-BADAEC codes quickly.

Our evaluation based on field data shows that SEC-BADAEC improves DRAM-level reliability by 23.5% and system-level reliability by 29.8% over SEC O-ECC. Despite its aggressive correction, SEC-BADAEC reduces system-level undetectable errors by 35.2%. Moreover, the hardware overhead is minimal (a few thousand logic gates and an additional 3-input OR gate delay).

The major contributions of this paper are as follows:

- We propose a novel class of ECC codes, called SEC-BADAEC, to strongly and efficiently protect DRAM with O-ECC. SEC-BADAEC limits its target DAEs to byte boundaries to correct most DAEs using the same redundancy as the current SEC.
- We propose a novel algorithm to construct SEC-BADAEC codes. It uses a systematic search based on the cyclic property and cosets of Galois Field to find SEC-BADAEC codes in a short time despite the long code length (136-bit).
- We implement an encoder and decoder for the constructed SEC-BADAEC code, showing that hardware overheads are minimal. The encoder has almost the same latency, area, and power as one for SEC, and a nominal increase in latency.

The rest of this paper is organized as follows: Section II provides the background to facilitate the understanding of SEC-BADAEC. Section III analyzes some of the most related

work in literature. Section IV provides the motivation behind SEC-BADAEC, and Section V provides an overview and details of SEC-BADAEC and its construction algorithm. Section VI evaluates SEC-BADAEC, demonstrating its superior correction capability and minimal hardware overheads. Section VII discusses other applications and scalability of SEC-BADAEC. Section VIII concludes the paper.

II. BACKGROUND

This section reviews the concepts and theories that are fundamental to understanding SEC-BADAEC. Brief introductions to DRAM, DRAM ECC, linear block codes, and Galois fields are presented.

A. DRAM ORGANIZATION

The high density of DRAM comes from its simple cell structure. A DRAM cell is composed of one transistor and one capacitor. A group of cells that share the same control signal over the access transistor is called a *row*, whereas a group of cells sharing the same bit-line is called a *column*. A bank is a two-dimensional cell array, and a DRAM has multiple banks to overlap accesses and hide long access latencies.

A deeper look at the DRAM bank structure can provide more details. Each bank is subdivided into 512 × 512-cell *MATs*. Each MAT has 512 local Bit-Line Sense-Amplifiers (BLSA). On a row activation, a BLSA senses a subtle change in the bit-line voltage, amplifies it to full VDD or VSS, and holds the voltage level. During a read/write, Column-Select Lines (CSLs) select a group of bits from BLSAs to transfer.

Outside a MAT, the selected data is transferred from/to a MAT via local and global I/Os. Modern DRAMs transfer 8 bits from a single MAT to reduce area overheads [17], [18], [19], [20]. A bank group gathers data from 17 MATs (one byte each) to build a 136-bit block and applies O-ECC [18], [20]. Due to the byte-aligned organization, a local fault within a MAT can only corrupt bits within its byte data, regardless of its severity. The byte alignment of MAT-local faults has been borne out in a large-scale measurement of field errors [14] and recent radiation tests of DRAM [9].

B. DRAM ECC

System companies have long used ECC to protect data from DRAM errors. They utilize *Rank-level ECC (R-ECC)* based on extra chips added to DRAM modules. On a 64-bit data interface, 8-bit redundancy is required for SEC-DED (Single Error Correction - Double Error Detection), leading to the standard 72-pin ECC-DIMM with 12.5% extra chips. Some systems provide SEC-DED protection using the 12.5% redundancy, while high-reliability systems utilize the same redundancy to provide *Chipkill-correct* ECC, which can restore data even if a chip completely fails [10], [21], [22], [23].

As DRAM raw error rates increase with processing scaling, DRAM vendors have introduced *On-die ECC (O-ECC)* since LPDDR4, DDR5, and HBM2E [12], [18], [24]. These DRAMs embed redundant cells and ECC circuits into each memory die and can correct errors internally. The internal correction allows a faulty device to operate externally fault-free and improves yields and reliability. The amount of redundancy is not standardized, but most companies utilize a multiple of 8 bits of redundancy (Section II-A). The eight extra bits can be used 1) to correct single-bit errors on 128-bit data (6.25% redundancy) or 2) to correct single-bit errors and detect double-bit errors on 64-bit data (12.5% redundancy). The extra detection capability is not very useful for O-ECC because commodity DRAMs do not have a real-time reporting mechanism for error detection [13]. On the contrary, DRAM vendors are sensitive to costs and utilize the SEC code with lesser redundancy and area overheads [17].

Errors uncorrected by O-ECC reach R-ECC, which can correct and/or detect severe errors using Chipkill corrections. Note that SRAM ECC requires safe detection because SRAMs have single-level ECC protection. Meanwhile, DRAMs have two-level protection, and poor error detection in O-ECC is acceptable owing to the safe detection of R-ECC [10]. Combined with O-ECC and R-ECC, the system can ensure data correctness unless a block has multiple chips with multi-bit errors.

C. LINEAR BLOCK CODE

DRAM ECCs utilize *linear block codes* (e.g., Hsiao [25] or Reed-Solomon [26]) to minimize decoding latency. On a memory write, a linear block code generates redundancy using linear combinations of the data; the valid pair of data and redundancy is called a *codeword*. An (n, k) linear block code generates a n-bit codeword (denoted as c) from k-bit data (denoted as m) and can be fully described by a $k \times n$ *Generator Matrix* (denoted as G) (Eq. 1).

$$c = m \times G \tag{1}$$

On a memory read, the received word (denoted as c_e) may have an error (denoted as e) (Eq. 2). An invalid pair of data and redundancy due to an error is called a *non-codeword*.

$$c_e = c + e \tag{2}$$

A linear block code decodes a word by multiplying it with a transposed *Parity Check Matrix* (denoted as H^T) (Eq. 3).

$$s = c_e H^T \tag{3}$$

The H-matrix has a dimension of $(n-k) \times n$ to have $(n-k)$-bit output in Eq. 3. The output is called a *syndrome* (denoted as s). If the G and H satisfy Eq. 4, the syndrome follows Eq. 5 and it must be zero for all valid codewords (i.e., e = 0). A non-codeword has a non-zero syndrome that is determined by the error only, regardless of the protected data.

$$GH^T = 0 \tag{4}$$

$$s = (c + e)H^T = mGH^T + eH^T = eH^T \tag{5}$$

An ECC can identify a class of errors if its H-matrix generates a unique non-zero syndrome for each constituent error. After identification, ECC can restore the data by flipping the

bit at the affected error position (or all erroneous bits, in the case of multi-bit correction). An H-matrix for SEC shall have unique non-zero column values. With a single-bit error (i.e., e is a one-hot vector), the calculated syndrome is the column value at the 1's position from Eq. 5. If each column is unique, the ECC can identify the error and correct it by flipping the bit at the column position.

If a word has multiple errors, the output syndrome is a sum of the individual syndromes (Eq. 6).

$$\begin{aligned} c_e &= c + e_1 + \ldots + e_m \\ s &= c_e H^T = e_1 H^T + \ldots + e_m H^T \end{aligned} \tag{6}$$

For DEC, an H-matrix shall have unique non-zero column values, and a sum of two distinct columns shall not match another column or sum. Such an error has e as a two-hot vector, and the syndrome is a sum of the two columns in the H-matrix. Unless the sum matches a column value (aliased to a 1-bit error) or another sum (aliased to another 2-bit error), the error can be identified and corrected by flipping bits at the column positions.

A *systematic code* embeds input data in the original form in the codeword. Systems prefer systematic codes for monitoring and other purposes (e.g., a faster hash lookup). Systematic codes have H and G matrixes as:

$$G = [I_k : P], H = [P^T : I_{(n-k)}], \tag{7}$$

where I_x indicates $x \times x$ identity matrix, and P indicates $k \times r$ sized parity matrix.

D. GALOIS FIELD

Each column in the H-matrix can be regarded as an element in a *Galois Field*. In mathematics, a field is a set on which addition, subtraction, multiplication, and division are defined (e.g., real numbers), and GF(q) is a field whose number of elements is finite to q. GF(2) has two elements, 0 and 1, and its addition and multiplications are defined as XOR and AND, respectively. For a positive integer k, there exists a GF(2^k). Elements of GF(2^k) can be represented as polynomials of degree strictly less than k over GF(2). Addition in GF(2^k) can be done by adding coefficients of two polynomials in GF(2). Multiplication can be done by multiplying the polynomials modulo an irreducible polynomial, called a generator polynomial.

A Galois Field has at least one α (called a primitive element) that can represent all non-zero elements as $1, \alpha^1, \alpha^2, \ldots \alpha^{q-2}$ (*cyclic property* [27]). These representations are called *multiplicative*, and multiplication in GF can also be done by adding the powers of these representations. To stay closed within the finite number of elements, α^{q-1} must be 1 [27], and multiplication leads to arithmetic addition modulo $q-1$ of their powers. SEC-BADAEC utilizes GF(256), which implies that α^{255} is 1.

III. PRIOR WORK

ECC has a trade-off between protection capabilities (e.g., detection and correction) and overheads (e.g., redundancy ratio and block size), leading to various ECCs for different purposes. This section presents the most related state-of-the-art ECCs for DRAM, classified by their protection capabilities. Table 1 presents a summary of the classes.

TABLE 1. Comparison of existing ECC classes and SEC-BADAEC.

Class	Error Protection			Memory ECC configs.		
	SE	DAE	DE	n	k	Redundancy
SEC	C	U	U	71	64	10.9%
				136	128	6.25%
SEC-DED	C	D	D	72	64	12.5%
				137	128	7.03%
DEC	C	C	C	78	64	21.9%
				144	128	12.5%
SEC-DED-DAEC	C	C	D	72	64	12.5%
				137	128	7.03%
SEC-BADAEC	C	AC	U	136	128	6.25%

SE: Single Error DAE: Double Adjacent Error DE: Double Error
(A)C: (Almost) Correctable D: Detectable U: Undetectable

A. SEC CODES

A SEC (Single Error Correction) ECC was first proposed by Hamming [28]. A Hamming code with r-bit redundancy can correct single errors on $(2^r - 1)$-bit blocks by filling its H-matrix with unique nonzero columns. With the full block size (e.g., 255-bit codeword), Hamming code is a perfect code by satisfying the Hamming bound [29]. However, memories are organized to have a power-of-2 data width and require a *shortening* of the H-matrix to match the combined data and redundancy size. After shortening, Hamming codes are applicable to DRAM O-ECC as (71, 64) or (136, 128) SEC ECCs.

B. SEC-DED CODES

SEC-DED (Single Error Correction–Double Error Detection) ECCs add double-error detection capability to SEC. Hamming extended his SEC code by adding an extra parity bit [28]. Hsiao modified the Hamming code to use odd-weight columns only in the H-matrix and reduce the number of ones in the matrix [25]. When generating parity bits on encoding or generating syndrome bits on decoding, an $\lceil log_2(x) \rceil$ depth XOR gate tree is needed, where x is the maximum weight (number of ones) of any row in the parity-check matrix. Therefore, minimizing the number of ones or doing weight balancing lowers hardware latency and complexity. Finally, the Hsiao code also reduces the miscorrection probability of triple bit errors and the detection probability of quadruple bit errors, relative to Hamming SEC-DED.

SEC-DED requires one more redundant bit than SEC, leading to (72, 64) and (137, 128) codes. Once an error is detected-but-uncorrectable by ECC, the system may rely on a higher-level recovery mechanism (e.g., software-based checkpoint and restart [30]) to correct the error. Although strong error detection is important to SRAM ECCs and DRAM R-ECCs, error detection has limited importance in O-ECC because commodity DRAMs do not have a real-time reporting mechanism for detected errors and use detection information primarily for fault diagnosis.

C. DEC CODES

DEC (Double Error Correction) ECC can correct any two-bit errors. A DEC code is commonly constructed from BCH (Bose-Chaudhuri-Hocquenghem) code. BCH codes provide multiple random bit error correction based on polynomial over Galois fields. But, BCH DEC requires twice the redundancy of SEC (e.g., 16-bit redundancy for 128-bit data DEC) and more complex decoding [31].

D. BURST ERROR CORRECTION CODES

A high-energy particle strike (e.g., proton) can upset multiple nearby memory cells, a symptom known as *Multiple-Bit Upset (MBU)*. An MBU corrupts multiple neighboring bits within an ECC block, and there is rich literature on correcting such errors using less redundancy than random ones. An *n*-bit adjacent error has *n* consecutive errors (e.g., 111 for 3-bit adjacent error), whereas an n-burst error has errors confined to *n* consecutive positions but not necessarily consecutive (e.g., 101 and 111 for 3-burst error).

References [32] and [33] proposed OLS codes that can correct adjacent errors with reduced overheads. Reference [34] proposed XGYL code that corrects X global errors (i.e., random errors) and Y-bit local errors (i.e., burst errors). Reference [35] proposed low-overhead burst ECC using identity matrix based on hamming code. Reference [36] proposed 3BEC-QAEC (3-bit burst error correction - quadruple adjacent error correction) code by extending 3BEC codes to correct 4-bit adjacent errors. However, these codes require additional parity bits than the baseline (e.g., random correction only) to correct burst errors.

A few studies add adjacent error correction capability without extra parities. SEC-DED-DAEC (SEC-DED-Double Adjacent Error Correction) code adds DAE correction to SEC-DED [37], [38], [39]. They utilize unused syndromes in SEC-DED to correct the frequent adjacent errors without additional bits. Dutta and Touba [37] proposed SEC-DED-DAEC codes by extending Hsiao SEC-DED codes [25]. Dutta requires its H-matrix to have unique and non-zero columns, no linear dependency involving three or fewer columns, and no linear dependency among two sums of adjacent columns. It uses a greedy search to find such a matrix. However, the computation time for the greedy search grows exponentially with code length, so the paper provided codes for up to (72, 64) blocks. Ming *et al.* [38] reduced the miscorrection probability of SEC-DED-DAEC by using high-weight column values. Naele and Sachdev [39] proposed SEC-DED-DAEC-xAED (x-bit Adjacent Error Detection) codes, which can correct DAEs and detect triple adjacent errors using the same redundancy as SEC-DED. It provides a systematic algorithm to build an H-matrix to find a SEC-DED-DAEC-TAED code quickly and detect longer adjacent errors with extra parity bits.

SEC-BADAEC is related to SEC-DED-DAEC in that it utilizes the unused syndromes to correct burst errors. On the other hand, SEC-BADAEC restricts target DAEs to fit into the unused syndrome of SEC instead of using codes with more redundancy (i.e., SEC-DED). The number of DAEs is large (={block size} − 1) and does not fit into the unused syndromes of SEC. As a result, the prior studies resorted to SEC-DED with more unused syndromes to map all DAEs. However, the extra 1-bit parity for DED can cause significant overheads in DRAM because DRAMs are organized with 8-bit granularity [17]. Instead, SEC-BADAEC takes an alternative approach of limiting target DAEs so that their numbers match the number of unutilized syndromes of SEC. By targeting byte-aligned DAEs only, it can provide similar correction coverage as DAEC using the same ECC configuration as SEC. As a result, SEC-BADAEC can provide a practical solution for DRAM O-ECC without changing the current organization.

IV. MOTIVATION

Despite the rich literature on ECC, there are few works optimized for O-ECC, and DRAM vendors are using aged and sub-optimal SEC ECC. This section provides the motivations that led to the development of SEC-BADAEC.

A. SHORTENED CODE

A (136, 128) ECC has 8-bit redundancy and can map 255 error patterns to its 255 non-zero syndrome values. However, there are only 136 single-error cases in the 136-bit word, leaving 119 non-zero syndromes. The current SEC O-ECC does not use the remaining 119 syndromes for correction but for detecting some multi-bit errors whose syndromes happen to match the unused value. Detected errors, however, has limited benefits in reliability because O-ECC uses the information primarily for diagnosis instead of preventing systems from failure. The DDR5 standard only utilizes error detection information as diagnosis counters on a few most problematic rows [13]. SEC-BADAEC improves reliability by mapping these unused syndromes to frequent multi-bit errors and correcting them.

B. DRAM ERRORS

DRAM errors are generated for various causes, including a defective circuit, weak cell, high-energy particle strike, and silicon wear-out [5], [14], [40], [41], [42], [43], [44], [45], [46] Most DRAM errors are single-bit and are correctable by SEC O-ECC. However, some faults (e.g., a wordline fault, a multiple-bit-upset from high-energy neutrons) generate multi-bit errors inside DRAM. These errors exceed the correction capability of O-ECC and are exposed outside the chip, breaking DRAM-level reliability and threatening R-ECC and the overall system reliability. The patterns and rates of multi-bit errors vary across DRAMs and systems, but a few large-scale measurements [5], [6], [7], [8], [14], [40], [47] reveal secrets under the hood.

Among the field measurements, we use ones with LPDDR [14] and HBM2 [9] to analyze errors for O-ECC. We especially focus on the LPDDR study, because it presents detailed error patterns of bigger blocks within a chip.

Other studies with commodity DRAMs (i.e., DDRx) provide limited insights into O-ECC due to their narrow chip width. Systems with commodity DRAMs group many DRAM chips to build a channel (e.g., 18 4-pin DDR4 chips for a 72-pin ECC DIMM), so that their word gathers a small amount of data (e.g., 4-bit) from many chips. Error patterns in such systems do not show per-chip error patterns beyond the small block. Meanwhile, LPDDR and HBM2 build a channel from a single chip, and their error patterns can provide per-chip error patterns at a larger granularity (e.g., 32-bit).

TABLE 2. Multi-bit errors from a large-scale LPDDR study [14].

Number of corrupted bits	Byte-aligned	Consecutive	Occurrences	Expected Value	Corrupted value
2	Yes	Yes	1	0x000016bb	0x000016b8
2	Yes	Yes	2	0x000003c1	0x000003c2
2	Yes	Yes	7	0xffffffff	0xffffff3ff
2	Yes	Yes	10	0xffffffff	0xffffff9ff
2	Yes	No	2	0xffffffff	0xffffeeff
2	Yes	No	4	0xffffffff	0xffff7dff
2	Yes	No	4	0xffffffff	0xffffff5ff
2	Yes	No	10	0xffffffff	0xffff77ff
2	Yes	No	36	0xffffffff	0xffff7bff
3	Yes	Yes	1	0xffffffff	0xffffff1ff
3	Yes	No	1	0xffffffff	0xffff75ff
4	Yes	Yes	1	0x00002957	0x00002958
4	Yes	No	1	0x00000461	0x00006e61
4	Yes	No	1	0x000071b2	0x00007100
5	Yes	No	1	0x000002e4	0x00000215
6	Yes	No	1	0x00006ab4	0x00006a5a
8	Yes	Yes	1	0xffffffff	0xffffff00
9	No	No	1	0x00000058	0xe6006358

Table 2 shows the raw DRAM errors collected from 945 SoC nodes with LPDDR over a year [14]. If a fault generates multiple incorrect values for many iterations, it is counted as a single independent error. After filtering out a faulty node, they observed 55,000 independent memory errors. Among the 55,000 errors, 85 errors are multi-bit. 84 out of the 85 multi-bit errors are confined to a single byte. This aligns with other studies that most DRAM errors are limited to a single byte due to the internal structure [17], [18], [19], [20]. Out of the 84 byte-aligned errors, 76 are two-bit errors. Among the two-bit errors, 20 have errors in adjacent positions (i.e., DAEs). In a 32-bit word, there are ${}_{32}C_2$ cases of two-bit errors and 28 cases of BADAEs. If all errors are random, BADAEs will be 5.6%(= 28/496) of the DEs, yet the measurement shows 26.3% of DEs are BADAEs, suggesting BADAEs are more frequent than other DEs. At a block level, a 136-bit O-ECC block has 9180 (=${}_{136}C_2$) DEs and 119 BADAEs (1.3% of DEs), but the 26.3% ratio between BADAEs and DEs may still apply, if the block has only one erroneous 32-bit word. SEC-BADAEC focuses on the BADAEs, which take 26.3% of two-bit errors in the study, and corrects them using the unused syndromes.

Sullivan *et al.* [9] exposed NVIDIA GPUs to a high-energy neutron beam to measure soft errors in HBM2. The results show that multi-bit soft errors are relatively common in HBM2, and that 86.7% of multi-bit errors are constrained within a byte. The number of corrupted bits in each multi-bit error is relatively large in the measurement, meaning that the errors are likely due to logic faults internal to the DRAM, and not direct cell strikes. Neutron beam testing provides accelerated testing of soft errors, but it does not cover errors from other sources (e.g., variable retention time, transistor wear-out).

V. PROPOSED CODE

O-ECC has a strict constraint of (136, 128) block size to meet layout design rules and minimize DRAM die size. We propose a novel class of ECC, called Single Error Correction–Byte Aligned Double Adjacent Error Correction (SEC-BADAEC), to fully utilize the code space for aggressive correction. We first provide an overview of SEC-BADAEC, followed by its properties and an efficient algorithm to construct such a code.

A. (136, 128) SEC-BADAEC

A (136, 128) code has 255 non-zero syndromes from its 8-bit redundancy. Among the syndromes, SEC uses 136 for single error correction and leaves 119 unused for correction. This number exactly matches the number of Byte-Aligned Double Adjacent Errors (BADAEs) in 136-bit words (7 DAEs per byte $\times$ 17 bytes per word). Our goal is to map these 119 BADAEs to the 119 unused syndromes to leave no vacancy in the syndromes and increase correction coverage to all single errors and BADAEs.

Although it is desirable to increase the correction coverage further (e.g., SEC-DED-DAEC or DEC), it can incur significant overheads with O-ECC. For example, SEC-DED-DAEC on 128-bit data requires 9-bit redundancy. However, the redundancy increases to 16 bits (12.5%) in DRAM because DRAMs require a multiple-of-8 word size. A DRAM vendor reported the overall chip area (including cells, peripherals, and I/O) increases by 6.9% for (136, 128) SEC and 12.1% for (137, 128) SEC-DED [17]. Decreasing the block size to 64-bit data can implement SEC-DED-DAEC with 8-bit redundancy, yet the redundancy ratio remains at 12.5% due to the smaller block size. Meanwhile, DAEs across a byte boundary are rare and did not appear in the large-scale field study [14] or neutron beam-testing results [9]. SEC-BADAEC can keep the redundancy at 6.25% but can provide the same level of correction as SEC-DED-DAEC by focusing on BADAEs.

B. CODE PROPERTY

Our proposed SEC-BADAEC code can correct all single errors and all BADAEs. In a linear block code, it means the H-matrix of SEC-BADAEC should have the following properties:

1) All columns are non-zero
2) All columns are unique.
3) The sum of ith and $(i+1)$th columns are unique, if $(i \bmod 8) \neq 7$.

The first and the second properties are for single-bit error correction. With a single-bit error (one-hot error location vector), the resulting syndrome is the column value at the error position. Therefore, each column value must be non-zero (to distinguish from no error) and unique (to distinguish from

other errors). The third property is for BADAE correction. A syndrome of multiple errors is the sum of syndromes of individual errors (Eq. 6). Double adjacent errors at position i and $(i+1)$, where i is not at the end of a byte, have its syndrome as the sum of ith and $(i+1)$ columns. By ensuring that the sum is unique, the ECC can identify and correct BADAEs.

C. CODE CONSTRUCTION

Although the number of non-zero syndromes matches the number of single errors and BADAEs, it is challenging to find such a code if it exists. One may try a greedy approach to find a solution. A greedy search finds a new H-matrix column based on previously-selected columns [36], [37], [38]. However, its search space grows exponentially with block sizes and is primarily for codes with short sizes (e.g., $\leq$ 72 bits). If we apply the greedy approach to SEC-BADAEC by excluding a new column value and its XOR with the previous column from column candidates, there are $\sim 10^{270}$ ($= 255 \times 254 \times 252 \times 250 \times \ldots$) permutation cases. Meanwhile, we expected there are a small number of solutions satisfying the 100% syndrome utilization for correction. We implemented the search in the C language and ran it for a week to cover $\sim 10^{10}$ cases, yet we could not find a solution.

$$H = \left[\left[\begin{array}{c} S_0 \\ (8x8) \end{array}\right]\left[\begin{array}{c} S_1 \\ (8x8) \end{array}\right]\cdots\left[\begin{array}{c} S_{15} \\ (8x8) \end{array}\right]\left[\begin{array}{c} \\ S_{16} \end{array}\right]\right]$$

FIGURE 3. 8 × 136 H-matrix is divided into 17 8 × 8 submatrices (S_i). The last submatrix is the identity matrix.

Instead, we propose a systematic algorithm based on a divide-and-conquer approach. It first divides the 8 × 136 H-matrix at a byte granularity into 17 8 × 8 submatrices (Figure. 3). The ith submatrix is denoted as S_i, and we use the 8 × 8 identity matrix (i.e., I_8) as the last submatrix (S_{16}) to make the code systematic. The identity matrix has unique columns and sums of adjacent columns to correct single errors and DAEs within the byte.

	8 columns in S_{16}								7 sums of adjacent columns in S_{16}						
Vector representation	$\begin{bmatrix}1\\0\\0\\0\\0\\0\\0\\0\end{bmatrix}$	$\begin{bmatrix}0\\1\\0\\0\\0\\0\\0\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\\1\\0\\0\\0\\0\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\\0\\1\\0\\0\\0\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\\0\\0\\1\\0\\0\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\\0\\0\\0\\1\\0\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\\0\\0\\0\\0\\1\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\\0\\0\\0\\0\\0\\1\end{bmatrix}$	$\begin{bmatrix}1\\1\\0\\0\\0\\0\\0\\0\end{bmatrix}$	$\begin{bmatrix}0\\1\\1\\0\\0\\0\\0\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\\1\\1\\0\\0\\0\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\\0\\1\\1\\0\\0\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\\0\\0\\1\\1\\0\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\\0\\0\\0\\1\\1\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\\0\\0\\0\\0\\1\\1\end{bmatrix}$
Multiplicative representation	α^7	α^6	α^5	α^4	α^3	α^2	α^1	α^0	α^{x+6}	α^{x+5}	α^{x+4}	α^{x+3}	α^{x+2}	α^{x+1}	α^x

FIGURE 4. Conversion of columns and the sum of adjacent columns in S_{16} into a multiplicative representation of GF(256).

Then we convert columns and column sums of S_{16} into elements of GF(256) (Fig. 4). The columns are represented as $\alpha^7, \alpha^6, \alpha^5, \ldots$, and α^0 in the GF multiplicative representation, and the adjacent sums are represented as α^{6+x}, α^{5+x}, α^{4+x}, $\ldots$, α^x, where α^x is a multiplicative representation of the rightmost sum (i.e., $\alpha^1 + \alpha^0$). Then, we group the 255 non-zero elements in GF(256) into 15 cosets, each of which has 17 elements. The ith coset is $\{\alpha^i, \alpha^{15+i}, \alpha^{30+i}, \ldots, \alpha^{240+i}\}$. Then we search for a primitive polynomial which maps $\alpha^1 + \alpha^0$ to an element in $Coset_8$ (i.e., (x mod 15) = 8). GF(256) has 16 irreducible polynomials, and we found two of them meet the requirement: 0x14D and 0x165. If we use 0x14D as the generator polynomial, x equals to 23, and the adjacent sums in the identify matrix will be $\{\alpha^{29}, \alpha^{28}, \alpha^{27}, \ldots, \alpha^{23}\}$. Please note that all the values and sums are from different cosets: columns from $Coset_7$, $Coset_6$, $\ldots$, $Coset_0$ and sums from $Coset_{14}$, $Coset_{13}$, $\ldots$, $Coset_8$.

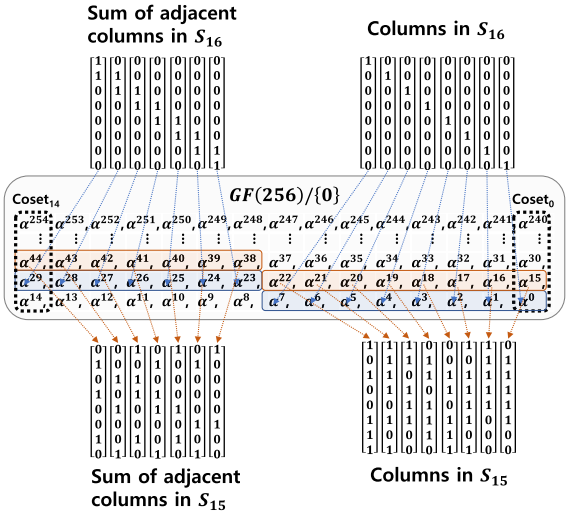
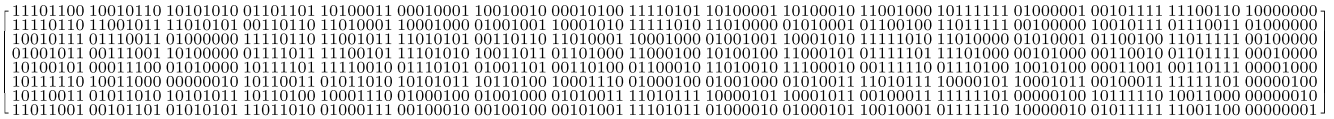


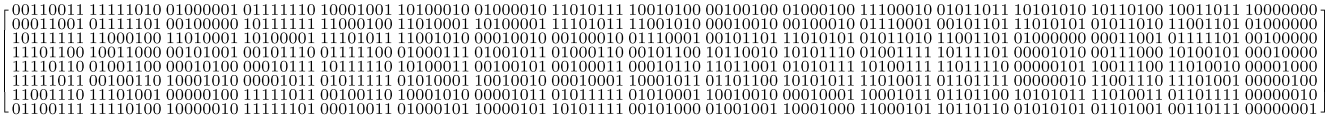
FIGURE 5. Derivation flow of S_{15} from S_{16}. Each column is multiplied by α^{15} to be the next element in each coset. Multiplied columns make the sums multiplied by the same value so that sums have the next element in their cosets.

The next step is extending S_{16} to other submatrices. During the derivation, a syndrome value used in one submatrix (for either a single error or DAE) shall not be used by another submatrix. To ensure the uniqueness of syndromes, we use GF multiplications and utilize their cyclic property. We multiply S_{16} by α^{15} to generate S_{15}. The columns will be $\{\alpha^{22}, \alpha^{21}, \alpha^{20}, \ldots, \alpha^{15}\}$, which are the next elements in each coset (Fig. 5). Similarly, the adjacent column sums ($\{\alpha^{44}, \alpha^{43}, \alpha^{42}, \ldots, \alpha^{38}\}$) are the next elements in each coset. They are distinct from column values and column sums used in the previous S_{16}, and their errors can be corrected.

We repeat this step ($\times\alpha^{15}$) to generate all other submatrices. If the power after the multiplication exceeds 255, it wraps around and goes to the first element in the coset because $\alpha^{255} = 1$ (i.e., cyclic). No column or sum will overlap because different bit offsets within a byte result in elements in different cosets, and different byte location causes values in different elements within the coset. After generating all submatrices, we convert the resulting H-matrix into a binary vector representation. Fig. 6 presents the full H-matrices for 0x14D and 0x165 generator polynomials.



(a) The H-matrix with generator polynomial = 0x14D



(b) The H-matrix with generator polynomial = 0x165

FIGURE 6. H-matrices of (136, 128) SEC-BADAEC.

VI. EVALUATION

We evaluate the DRAM-level reliability, system-level reliability, latency, area, and power impacts of SEC-BADAEC and compare them against the current SEC O-ECC. The results show that SEC-BADAEC can reduce uncorrectable errors in DRAM by 23.5% and improve system-level reliability by 29.8% with negligible hardware overheads.

A. DRAM-LEVEL RELIABILITY

We evaluate DRAM-level reliability using the error patterns from a large-scale field measurement [14]. We count the number of uncorrectable errors in the profiled errors (Table 2). SEC cannot correct multi-bit errors and therefore it suffers from 85 uncorrectable errors. SEC-BADAEC can correct all single errors and BADAEs to reduce the uncorrectable count by 20. This translates to a 23.5% reduction in the uncorrectable error probability from a DRAM chip.

The aggressive correction of SEC-BADAEC leaves no syndrome for detection-only, yet the DRAM standard utilizes detection information primarily for failure analysis using error counters. If needed, SEC-BADAEC can provide the same error count by incrementing the counter with BADAE corrections, which would be detected errors with SEC.

B. SYSTEM-LEVEL RELIABILITY

System-level reliability is measured by estimating uncorrectable error and undetectable error probabilities in a system. We run Monte-Carlo simulations to inject errors into a DRAM channel and apply O-ECC and Chipkill-correct R-ECC to find out whether the error is correctable (CE), detectable-but-uncorrectable (DUE), or left-uncorrected (silent data corruption, or SDC).

1) EVALUATION ENVIRONMENT

The target system uses 18 DDR4 chips (4 data pins each) to build a channel, typical for high-reliability systems. Each DRAM chip uses 8-bit redundancy over 128-bit data to implement SEC or SEC-BADAEC O-ECC. Each chip transfers 32-bit data per access using 4 data pins and burst length of 8, so that the 128-bit data from an O-ECC block spreads over four memory accesses. For R-ECC, we use AMD Chipkill-correct [21] for its efficiency. It builds an 8-bit symbol from 2 beats of per-chip data. A memory access has four R-ECC words from the burst length of 8. If those words have different corrected chip positions, AMD Chipkill discards the corrections and reports a DUE to minimize miscorrections (the conservative correction [10]).

We randomly inject multi-bit errors into two chips in the channel. We focus on multi-bit errors because single-bit errors are always correctable by O-ECC. We model multi-bit errors as one of three types: BADAE, DE, and Chipkill. A BADAE has two adjacent bit errors at a random starting position but does not cross a byte boundary, and a DE has two bit errors at random positions. BADAE patterns are excluded from DEs. A Chipkill corrupts each bit with 50% flipping probability. We inject the errors into two chips because single-chip errors are always correctable by R-ECC and three-chip errors are extremely rare. As a result, there are 6 scenarios of multi-bit/two-chip errors: BADAE+BADAE, BADAE+DE, BADAE+Chipkill, DE+DE, DE+Chipkill, and Chipkill+Chipkill. For each scenario, we randomly inject errors and apply O-ECC and R-ECC to find out the error is correctable or detectable. We perform 1 billion Monte-Carlo simulations for each measurement.

2) EVALUATION RESULTS

Table 3 presents the CE/DUE/SDC probabilities of each error scenario with SEC and SEC-BADAEC. SEC + AMD Chipkill-correct can correct some 2-chip/multi-bit errors. The 128-bit data from an O-ECC block spread over 4 memory accesses and sixteen R-ECC words. If the errors after O-ECC belong to different R-ECC words (e.g., chip 0 has double errors in the 1st and 4th R-ECC words, while chip 1 has double errors in the 5th and 8th R-ECC words), they can be corrected by the R-ECC. Note that AMD chipkill-correct utilizes the conservative correction to minimize SDCs. As a result, corrections on different chips within an access lead to a DUE, while corrections on different chips across memory accesses lead to a CE.

SEC-BADAEC + R-ECC can correct multi-bit/two-chip errors, if one of them is a BADAE. For more severe errors, however, SEC-BADAEC can sometimes increase the error severity through BADAE miscorrection. For DEs, for example, SEC-BADAEC increases the error severity from

TABLE 3. Comparison of system-level reliability against fault scenarios. The system utilizes both O-ECC and R-ECC (AMD Chipkill-correct).

Error Scenarios		ECC	SEC +	SEC-BADAEC +
Error Types	Weight	Result	AMD Chipkill	AMD Chipkill
BADAE + BADAE	5.54%	CE	43.12%	CE 100%
		DUE	56.22%	
		SDC	0.66%	
BADAE + DE	31.00%	CE	38.26%	
		DUE	60.51%	
		SDC	1.24%	
BADAE + Chipkill	4.98%	CE	2.52%	
		DUE	94.51%	
		SDC	2.97%	
DE + DE	43.40%	CE	22.52%	12.16%
		DUE	76.18%	86.10%
		SDC	1.30%	1.74%
DE + Chipkill	13.95%	CE	0.14%	0%
		DUE	99.26%	99.74%
		SDC	0.60%	0.26%
Chipkill + Chipkill	1.12%	CE	0%	0%
		DUE	100%	100%
		SDC	0%	0%
Overall	100%	CE	24.17%	46.80%
		DUE	74.61%	52.40%
		SDC	1.22%	0.79%

2-bit to 3-bit (53.3%, if the syndrome matches one of the 136 single-error syndromes) or to 4-bit (46.7%, if the syndrome matches one of the 119 BADAE syndromes). Meanwhile, SEC increases the error severity from 2-bit to 3-bit (53.3%, if the syndrome matches a single-error one) or keeps the severity constant (46.7%, if the syndrome has no match). The increased error severity leads to fewer CEs and more SDCs in the DE+DE scenario. In the scenario, errors are rare, so that R-ECC can correct some scattered errors if all of its R-ECC words have errors from a single chip. Meanwhile, with SEC-BADAEC, the chances of having all R-ECC words having up to one-chip error decreases with a bigger number of error bits, leading to fewer CEs and more SDCs. On the contrary, the increased severity actually provides better detection in the Chipkill+DE scenario. In the scenario, almost all R-ECC words already have errors from the chipkill fault, and adding one or two-bit errors from the miscorrection does not degrade the CE ratio significantly. Instead, more errors make it easier to be detected by R-ECC. For the Chipkill+Chipkill scenario, both show similar reliability levels because of the strong detection capability of AMD Chipkill-correct.

To estimate overall CE/DUE/SDC ratios, we weight each scenario based on the field measurement data from [14]. The measurement shows 23.5% (=20/85) of multi-bit errors are BADAEs and 65.9% (=56/85) of multi-bit errors are DEs (excluding BADAEs). We regard the remaining 10.6% as Chipkills. From the probabilities of single-chip errors, we can extract probabilities of two-chip error scenarios, assuming chips are isolated and have independent errors. For example, BADAE + BADAE will take 5.54%(= 23.5% × 23.5%) of multi-bit/two-chip errors. The weighted sum indicates SEC can correct 24.2% of multi-bit/two-chip errors, while SEC-BADAEC almost doubles the correction coverage to correct 46.8% of errors. In terms of uncorrectable errors, the failure rate decreases by 29.8% (from 75.8% to 53.2%). Despite the aggressive correction, the system-level SDC rate improves down to 0.8% by correcting more error scenarios.

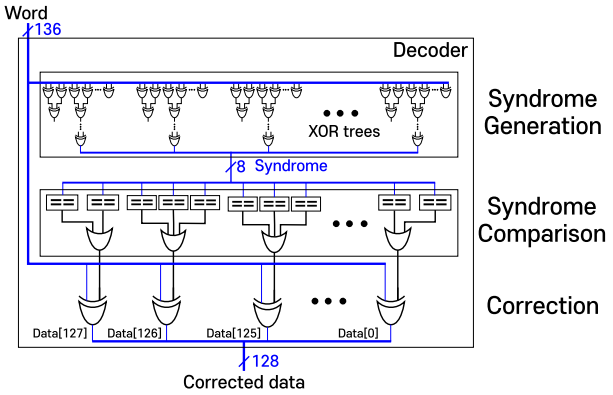


FIGURE 7. Decoder implementation used in hardware overhead evaluation. The comparators check equality to fixed syndrome values and can be optimized during a logic synthesis.

C. HARDWARE OVERHEADS

To estimate the latency, area, and power overheads of SEC-BADAEC, we built Verilog models of (136, 128) SEC and SEC-BADAEC. Fig. 7 presents an overview of the decoder designs. It generates a syndrome using XOR-trees on the input data. For SEC, each output bit position compares the syndrome against the corresponding column in the H-matrix and flips the output with a match. For SEC-BADAEC, each output position compares the syndrome against two or three values: the column value for single error correction, the sum with the previous column and the sum with the next column. The additional comparators and OR gates are the primary hardware costs of SEC-BADAEC. All the column and sums are pre-calculated, and a logic synthesis optimizes the comparison-to-constant operations. Both encoders utilize XOR-trees to generate redundancy.

We synthesized the models using Synopsys Design Compiler with UMC 28nm HVT standard cells. We could not find a public standard cell library for a DRAM process and used the logic process instead, expecting the HVT resembles DRAM processes highly optimized to minimize leakage currents. The estimated latency using the logic process is about half of one published by a DRAM vendor [17]. The target frequency is 400MHz to model DRAM internal operations. We measured the total (static+dynamic) power consumption using the default switching activity factor of 10%.

TABLE 4. Comparison of hardware overheads between SEC and SEC-BADAEC.

Metric	Encoder		Decoder	
	SEC	SEC-BADAEC	SEC	SEC-BADAEC
Latency (ns)	0.29	0.29	0.42	0.46
Area (um^2)	381	403	766	1750
Total power (mW)	0.207	0.216	0.492	1.606

Table 4 compares the hardware overheads between SEC and SEC-BADAEC. The encoders have the same latency and similar area and power overheads. This is because both

encoders have the same structure with different H-matrixes. SEC-BADAEC decoder has a longer latency than SEC because each bit position compares the syndrome against up to 3 values (one for single-bit error at the position and two for adjacent errors to the previous and the next position). Due to the additional comparisons, the area increases by $984um^2$, which corresponds to 3,347 2-input NAND gates. A DRAM chip requires 4 O-ECC circuits (one for each bank-group), and the total area increase is <0.1% of a DRAM die size ($\sim 50mm^2$) [11]. The extra power consumption due to SEC-BADAEC is 1.114 mW, because only one bank-group is accessed at a time. This translates to 0.34% of DRAM read power consumption [48].

VII. DISCUSSION

A. SCALABILITY OF SEC-BADAEC

SEC-BADAEC is based on (136, 128) SEC in the current DRAM products. However, we believe that DRAM vendors will keep using the (136, 128) code for the foreseeable future and that SEC-BADAEC can apply to future DRAMs for the reasons described below;

First, shrinking technology makes DRAMs more vulnerable to errors. However, the current SEC provides sufficiently good protection against reasonable BERs. For instance, SEC O-ECC alone can provide high yield and reliability up to 10^{-6} BER, and combining SEC O-ECC with Chipkill-correct R-ECC can tolerate up to 10^{-4} BER [17], [49], [50]. We are susceptible that a DRAM with a higher BER can be commercialized and require strong protection against random bit errors.

Second, Process scaling can increase MBUs by reducing critical charges. However, modern DRAM sub-systems have two-level protection. R-ECC operates as the final gatekeeper of reliability with Chipkill-correct, whereas O-ECC primarily targets improving the yield against inherent faults and reducing errors exposed outside the chip. Although MBUs are not correctable by O-ECC, they are still correctable by the R-ECC except very rare cases of concurrent multi-chip errors. Given that there is another protection at the rank-level, we are susceptible that DRAM vendors will strengthen O-ECC to correct MBUs at cost of more area. On the contrary, SEC-BADAEC can significantly reduce the probabilities of the rare cases with negligible overheads. Please note that SRAMs have single-level protection and require strong correction capability against MBUs.

Last but not least, DRAM vendors are likely to keep using the current access granularity (128 or 256 bit). The access granularity is optimized to cache line sizes to avoid overfetching. For example, all recent GDDR/LPDDR/HBM standards (GDDR4/5/6, LPDDR3/4/5, and HBM1/2/3) use 256-bit prefetching across all densities to match the 32B access granularity of GPU sector caches and other multimedia IPs. Because DRAM access granularity is unlikely to grow, O-ECC block size is unlikely to change. For the reasons described above, we expect DRAM will keep using (136, 128) ECC codes for the foreseeable future.

B. OTHER SEC-BADAEC USES

SEC-BADAEC corrects an error using every possible syndrome, maximizing the number of errors that can be corrected with a given redundancy. Such aggressive error correction is needed not only in DRAM but also in other storage structures as well as data transmission.

Large on-chip SRAM structures are vulnerable to soft errors, and SRAM errors have been shown to exhibit spatial locality—the linear ionizing track that is deposited following an energetic particle strike is long enough to affect multiple neighboring SRAM cells [51], [52], [53]. Furthermore, SRAM macros are often laid out with P-Well taps that may encourage multi-bit errors to be byte aligned [51]. Thus, SEC-BADAEC could potentially offer higher availability in SRAM by correcting the most frequent multi-bit soft errors.

Forward Error Correction (FEC) is often employed for high-speed transmission in technologies such as PCIe 6.0 [54]. In a FEC scheme, an aggressive error-correcting code is often combined with an error-detecting CRC. Any uncorrectable errors result in a retransmission request, and aggressive ECC serves to reduce the request rate and improve overall performance. Transmission pin errors due to crosstalk and power noise tend to be bursty in nature, such that BADAE correction could improve the efficacy of FEC.

VIII. CONCLUSION

This paper presents a novel class of On-Die ECC codes called SEC-BADAEC that are able to correct both single-bit errors as well as byte-aligned double-bit-adjacent errors. SEC-BADAEC requires no additional redundancy, and it utilizes the full 255 non-zero syndromes possible with 8b of redundancy. Thus, SEC-BADAEC is able to more aggressively correct errors than traditional On-Die ECC, improving memory reliability by 23.5% and system-level reliability by 29.8%. SEC-BADAEC offers a drop-in replacement for On-Die SEC, meaning it has negligible latency, area, and power overheads, suffers no performance degradation, and does not require changes to the DRAM interface.

REFERENCES

[1] N. George and J. Lach, "Characterization of logical masking and error propagation in combinational circuits and effects on system vulnerability," in *Proc. IEEE/IFIP 41st Int. Conf. Dependable Syst. Netw. (DSN)*, Jun. 2011, pp. 323–334.
[2] R. M. Chen, N. N. Mahatme, Z. J. Diggins, L. Wang, E. X. Zhang, Y. P. Chen, Y. N. Liu, B. Narasimham, A. F. Witulski, B. L. Bhuva, and D. M. Fleetwood, "Impact of temporal masking of flip-flop upsets on soft error rates of sequential circuits," *IEEE Trans. Nucl. Sci.*, vol. 64, no. 8, pp. 2098–2106, Aug. 2017.
[3] A. Evans and A. Lexandrescu, "System level design considerations for soft error mitigation," in *Proc. IEEE CPMT Chapter SCV Soft Error Rate Workshop*, Nov. 2016, pp. 24–27.
[4] G. Wang, L. Zhang, and W. Xu, "What can we learn from four years of data center hardware failures?" in *Proc. 47th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw. (DSN)*, Jun. 2017, pp. 25–36.
[5] V. Sridharan and D. Liberty, "A study of DRAM failures in the field," in *Proc. Int. Conf. High Perform. Comput., Netw., Storage Anal.*, Nov. 2012, pp. 1–11.

[6] V. Sridharan, J. Stearley, N. DeBardeleben, S. Blanchard, and S. Gurumurthi, "Feng Shui of supercomputer memory: Positional effects in DRAM and SRAM faults," in *Proc. Int. Conf. High Perform. Comput., Netw., Storage Anal.*, Nov. 2013, pp. 1–11.

[7] 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," in *Proc. 20th Int. Conf. Architectural Support Program. Lang. Operating Syst.*, New York, NY, USA, Mar. 2015, pp. 297–310.

[8] 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 *Proc. 45th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw.*, Jun. 2015, pp. 415–426.

[9] M. B. Sullivan, N. Saxena, M. O'Connor, D. Lee, P. Racunas, S. Hukerikar, T. Tsai, S. K. S. Hari, and S. W. Keckler, "Characterizing and mitigating soft errors in GPU DRAM," in *Proc. 54th Annu. IEEE/ACM Int. Symp. Microarchitecture*, New York, NY, USA, Oct. 2021, pp. 641–653, doi: 10.1145/3466752.3480111.

[10] J. Kim, M. Sullivan, and M. Erez, "Bamboo ECC: Strong, safe, and flexible codes for reliable computer memory," in *Proc. IEEE 21st Int. Symp. High Perform. Comput. Archit. (HPCA)*, Feb. 2015, pp. 101–112.

[11] A. Shilov, "SK Hynix details DDR5-6400," AnandTech Future, New York, NY, USA, Feb. 2019. [Online]. Available: https://www.anandtech.com/show/13999/sk-hynix-details-its-ddr56400-dram-chip

[12] R. Rooney and N. Koyle, "Micron® DDR5 SDRAM: New features," Micron Technol. Inc., Boise, ID, USA, Tech. Rep. CCM004-676576390-11390, 2019. [Online]. Available: https://www.micron.com/-/media/client/global/documents/products/white-paper/ddr5_new_features_white_paper.pdf

[13] *DDR5 SDRAM, JESD79-5A*, JEDEC Solid State Technol. Assoc., Arlington, VI, USA, Oct. 2021.

[14] L. Bautista-Gomez, F. Zyulkyarov, O. Unsal, and S. McIntosh-Smith, "Unprotected computing: A large-scale study of DRAM raw error rate on a supercomputer," in *Proc. Int. Conf. High Perform. Comput., Netw., Storage Anal.*, Nov. 2016, pp. 645–655.

[15] A. Makihara, H. Shindou, N. Nemoto, S. Kuboyama, S. Matsuda, T. Oshima, T. Hirao, H. Itoh, S. Buchner, and A. Campbell, "Analysis of single-ion multiple-bit upset in high-density DRAMs," *IEEE Trans. Nucl. Sci.*, vol. 47, no. 6, pp. 2400–2404, Dec. 2000.

[16] S. M. Seyedzadeh, D. Kline, A. K. Jones, and R. Melhem, "Mitigating bitline crosstalk noise in DRAM memories," in *Proc. Int. Symp. Memory Syst.*, New York, NY, USA, Oct. 2017, pp. 205–216, doi: 10.1145/3132402.3132410.

[17] S. Cha, O. Seongil, H. Shin, S. Hwang, K. Park, S. J. Jang, J. S. Choi, G. Y. Jin, Y. H. Son, H. Cho, J. H. Ahn, and N. S. Kim, "Defect analysis and cost-effective resilience architecture for future DRAM devices," in *Proc. IEEE Int. Symp. High Perform. Comput. Archit. (HPCA)*, Feb. 2017, pp. 61–72.

[18] T. Oh, H. Chung, J.-Y. Park, K. Lee, S.-H. Oh, S.-Y. Doo, H.-J. Kim, C.-Y. Lee, H.-R. Kim, J.-H. Lee, J.-I. Lee, K.-S. Ha, Y. Choi, Y. Cho, Y.-C. Bae, T. Jang, C. Park, K. I. Park, S.-J. Jang, and J.-S. Choi, "A 3.2 Gbps/pin 8 Gbit 1.0 V LPDDR4 SDRAM with integrated ECC engine for sub-1 V DRAM core operation," *IEEE J. Solid-State Circuits*, vol. 50, no. 1, pp. 178–190, Jan. 2015.

[19] N. Chatterjee, M. Oconnor, D. Lee, D. R. Johnson, S. W. Keckler, M. Rhu, and W. J. Dally, "Architecting an energy-efficient DRAM system for GPUs," in *Proc. IEEE Int. Symp. High Perform. Comput. Archit. (HPCA)*, Feb. 2017, pp. 73–84.

[20] K. Sangyun, Y. Seo, H. Kwon, M. Lee, and C. Sunghye, "Semiconductor memory devices and methods of operating semiconductor memory devices," U.S. Patent 16/812 850, Jan. 28, 2021.

[21] Advanced Micro Devices (AMD). (2013). *BIOS and Kernel Developer's Guide (BKDG) for AMD Family 15h Models 00h-0Fh Processors*. [Online]. Available: https://www.amd.com/system/files/TechDocs/42301_15h_Mod_00h-0Fh_BKDG.pdf

[22] X. Jian, N. DeBardeleben, S. Blanchard, V. Sridharan, and R. Kumar, "Analyzing reliability of memory sub-systems with double-chipkill detect/correct," in *Proc. IEEE 19th Pacific Rim Int. Symp. Dependable Comput.*, Dec. 2013, pp. 88–97.

[23] X. Jian, H. Duwe, J. Sartori, V. Sridharan, and R. Kumar, "Low-power, low-storage-overhead chipkill correct via multi-line error correction," in *Proc. Int. Conf. High Perform. Comput., Netw., Storage Anal. (SC)*, 2013, pp. 1–12.

[24] C.-S. Oh *et al.*, "A 1.1 V 16 GB 640 GB/s HBM2E DRAM with a data-bus window-extension technique and a synergetic on-die ECC scheme," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2020, pp. 330–332.

[25] M. Y. Hsiao, "A class of optimal minimum odd-weight-column SEC-DED codes," *IBM J. Res. Develop.*, vol. 14, no. 4, pp. 395–401, Jul. 1970.

[26] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *J. Soc. Ind. Appl. Math.*, vol. 8, no. 2, pp. 300–304, Jun. 1960.

[27] R. Lidl and H. Niederreiter, *Introduction to Finite Fields and Their Applications*. Cambridge, U.K.: Cambridge Univ. Press, 1994.

[28] R. W. Hamming, "Error detecting and error correcting codes," *Bell Syst. Tech. J.*, vol. 29, no. 2, pp. 147–160, Apr. 1950.

[29] R. Ahlswede, H. K. Aydinian, and L. H. Khachatrian, "On perfect codes and related concepts," *Designs, Codes Cryptogr.*, vol. 22, no. 3, pp. 221–237, 2001.

[30] J. Chung, I. Lee, M. Sullivan, J. H. Ryoo, D. W. Kim, D. H. Yoon, L. Kaplan, and M. Erez, "Containment domains: A scalable, efficient, and flexible resilience scheme for exascale systems," in *Proc. Int. Conf. High Perform. Comput., Netw., Storage Anal.*, Nov. 2012, pp. 1–11.

[31] R. Naseer and J. Draper, "DEC ECC design to improve memory reliability in sub-100 nm technologies," in *Proc. 15th IEEE Int. Conf. Electron., Circuits Syst.*, Aug. 2008, pp. 586–589.

[32] R. Datta and N. A. Touba, "Generating burst-error correcting codes from orthogonal Latin square codes—A graph theoretic approach," in *Proc. IEEE Int. Symp. Defect Fault Tolerance VLSI Nanotechnol. Syst.*, Oct. 2011, pp. 367–373.

[33] P. Reviriego, S. Liu, J. A. Maestro, S. Lee, N. A. Touba, and R. Datta, "Implementing triple adjacent error correction in double error correction orthogonal Latin squares codes," in *Proc. IEEE Int. Symp. Defect Fault Tolerance VLSI Nanotechnol. Syst. (DFTS)*, Oct. 2013, pp. 167–171.

[34] S. Shamshiri and K.-T. Cheng, "Error-locality-aware linear coding to correct multi-bit upsets in SRAMs," in *Proc. IEEE Int. Test Conf.*, Nov. 2010, pp. 1–10.

[35] A. Das and N. A. Touba, "Low complexity burst error correcting codes to correct MBUs in SRAMs," in *Proc. Great Lakes Symp. VLSI*, May 2018, pp. 219–224.

[36] J. Li, P. Reviriego, L. Xiao, C. Argyrides, and J. Li, "Extending 3-bit burst error-correction codes with quadruple adjacent error correction," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 26, no. 2, pp. 221–229, Feb. 2018.

[37] A. Dutta and N. A. Touba, "Multiple bit upset tolerant memory using a selective cycle avoidance based SEC-DED-DAEC code," in *Proc. 25th IEEE VLSI Test Symp. (VTS)*, May 2007, pp. 349–354.

[38] Z. Ming, X. Li Yi, and L. H. Wei, "New SEC-DED-DAEC codes for multiple bit upsets mitigation in memory," in *Proc. IEEE/IFIP 19th Int. Conf. VLSI System-Chip*, Oct. 2011, pp. 254–259.

[39] A. Neale and M. Sachdev, "A new SEC-DED error correction code subclass for adjacent MBU tolerance in embedded memory," *IEEE Trans. Device Mater. Rel.*, vol. 13, no. 1, pp. 223–230, Mar. 2013.

[40] A. A. Hwang, I. A. Stefanovici, and B. Schroeder, "Cosmic rays don't strike twice: Understanding the nature of DRAM errors and the implications for system design," in *Proc. 17th Int. Conf. Architectural Support Program. Lang. Operating Syst.*, New York, NY, USA, 2012, pp. 111–122, doi: 10.1145/2150976.2150989.

[41] A. Bacchini, M. Rovatti, G. Furano, and M. Ottavi, "Characterization of data retention faults in DRAM devices," in *Proc. IEEE Int. Symp. Defect Fault Tolerance VLSI Nanotechnol. Syst. (DFT)*, Oct. 2014, pp. 9–14.

[42] J. Liu, B. Jaiyen, Y. Kim, C. Wilkerson, and O. Mutlu, "An experimental study of data retention behavior in modern DRAM devices: Implications for retention time profiling mechanisms," in *Proc. 40th Annu. Int. Symp. Comput. Archit.*, Jun. 2013, vol. 41, no. 3, pp. 60–71, doi: 10.1145/2485922.2485928.

[43] A. Hiraiwa, M. Ogasawara, N. Natsuaki, Y. Itoh, and H. Iwai, "Statistical modeling of dynamic random access memory data retention characteristics," *J. Appl. Phys.*, vol. 80, no. 5, pp. 3091–3099, Sep. 1996.

[44] M. K. Qureshi, D.-H. Kim, S. Khan, P. J. Nair, and O. Mutlu, "AVATAR: A variable-retention-time (VRT) aware refresh for DRAM systems," in *Proc. 45th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw.*, Jun. 2015, pp. 427–437.

[45] J. Park, J. H. Lee, S.-K. Park, K. C. Chun, K. Sohn, and S. Kang, "An in-DRAM BIST for 16 Gb DDR4 DRAM in the 2 nd 10-nm-class DRAM process," *IEEE Access*, vol. 9, pp. 33487–33497, 2021.

[46] Y. Kim, R. Daly, J. Kim, C. Fallin, J. H. Lee, D. Lee, C. Wilkerson, K. Lai, and O. Mutlu, "Flipping bits in memory without accessing them: An experimental study of DRAM disturbance errors," *ACM SIGARCH Comput. Archit. News*, vol. 42, no. 3, pp. 361–372, Jun. 2014, doi: 10.1145/2678373.2665726.
[47] B. Schroeder, E. Pinheiro, and W.-D. Weber, "DRAM errors in the wild: A large-scale field study," *Commun. ACM*, vol. 54, no. 2, pp. 100–107, Feb. 2011, doi: 10.1145/1897816.1897844.
[48] *TN-40–07: Calculating Memory Power for DDR4 SDRAM*, Micron, Boise, ID, USA, 2017.
[49] U. Kang, H.-S. Yu, C. Park, H. Zheng, J. Halbert, K. Bains, S. Jang, and J. S. Choi, "Co-architecting controllers and DRAM to enhance DRAM process scaling," in *Proc. Memory Forum*, vol. 14, 2014, p. 11.
[50] P. J. Nair, D.-H. Kim, and M. K. Qureshi, "ArchShield: Architectural framework for assisting DRAM scaling by tolerating high error rates," *ACM SIGARCH Comput. Archit. News*, vol. 41, no. 3, pp. 72–83, 2013.
[51] D. Radaelli, H. Puchner, S. Wong, and S. Daniel, "Investigation of multi-bit upsets in a 150 nm technology SRAM device," *IEEE Trans. Nucl. Sci.*, vol. 52, no. 6, pp. 2433–2437, Dec. 2005.
[52] N. Mahatme, B. Bhuva, Y.-P. Fang, and A. Oates, "Analysis of multiple cell upsets due to neutrons in SRAMs for a deep-N-well process," in *Proc. Int. Rel. Phys. Symp.*, Apr. 2011, pp. 1–6.
[53] G. Gasiot, D. Giot, and P. Roche, "Multiple cell upsets as the key contribution to the total SER of 65 nm CMOS SRAMs and its dependence on well engineering," *IEEE Trans. Nucl. Sci.*, vol. 54, no. 6, pp. 2468–2473, Dec. 2007.
[54] *PCI Express Base Specification Revision 6.0*, PCI-SIG, Beaverton, OR, USA, Jan. 2022.

YUSEOK SONG received the B.S. degree in semiconductor systems engineering from Sungkyunkwan University, South Korea, in 2021, where he is currently pursuing the M.S. degree in electrical and computer engineering. His current research interests include memory sub systems and reliability.

SANGJAE PARK received the B.S. degree in electronic and electrical engineering from Sungkyunkwan University, South Korea, in 2021, where he is currently pursuing the M.S. degree in electrical and computer engineering. His current research interest includes memory reliability.

MICHAEL B. SULLIVAN (Member, IEEE) received the bachelor's and master's degrees in computer science, computer engineering, and mathematics from George Mason University, and the M.S. and Ph.D. degrees in computer architecture from the University of Texas at Austin. Specifically, he has studied system-level reliability modeling with cross-layer coordination, strong memory system protection, low-cost pipeline protection, and efficient and reliable application-specific acceleration. He is currently a Senior Research Scientist with the Architecture Research Group, NVIDIA. His research interests include design of efficient, dependable, and secure large-scale computer systems.

JUNGRAE KIM (Senior Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from Seoul National University, South Korea, and the Ph.D. degree from the University of Texas at Austin, USA. From 2003 to 2016, he was a Senior Engineer at Samsung Electronics, where he developed application processors. From 2016 to 2020, he was a Senior Software Engineer at Microsoft corporation, USA, architecting data center accelerators. He joined Sungkyunkwan University, in 2020, as an Assistant Professor. His current research interests include heterogeneous architecture, memory sub systems, and reliability.