

Received 14 November 2024, accepted 14 December 2024, date of publication 24 December 2024,
date of current version 28 February 2025.

Digital Object Identifier 10.1109/ACCESS.2024.3522023

RESEARCH ARTICLE

Double Adjacent Error Correction Codes for Ultra-Fast Cache Memories

RABAH ABOOD AHMED¹ AND KHAIRULMIZAM SAMSUDIN²

¹Department of Automation and Artificial Intelligence Engineering, College of Information Engineering, Al Nahrain University, Al Jadriya, Baghdad 64074, Iraq

²Department of Computer and Communication Systems, Faculty of Engineering, Universiti Putra Malaysia, Serdang, Selangor 43400, Malaysia

Corresponding author: Khairulmizam Samsudin (khairulmizam@upm.edu.my)

ABSTRACT Error correction codes are commonly used to protect cache memories from soft errors. As technology feature size scales deeper into sub-nanometer regime, radiation-induced soft error can cause double adjacent error (DAE). Several double adjacent error correction (DAEC) codes have been introduced to address DAEs, however, they miscorrect some nonadjacent double errors. In progress, a class of DAEC orthogonal Latin squares (OLS) codes is introduced to eliminate all miscorrections, using the orthogonality property of OLS codes, and also reduces the decoding delay time. The main drawback comes from the large number of check bits, imposed by the conventional OLS codes. In this paper, two coding approaches are developed based on a modified SEC OLS coding scheme that requires less number of check bits. The first approach is a class of SEC-DED-DAEC codes proposed to reduce the number of check bits compared to the existing SEC-DED-DAEC OLS codes. The second approach is a class of SEC-DAEC codes with a very high speed decoding process. This approach is designed as SEC OLS scheme and integrated with new modules for detecting and correcting the DAE error. The evaluation of the proposed SEC-DAEC codes in 45nm ASIC technology shows promising results. The decoding delay for protecting 16, 64, and 256 bit data words is less by at least 20% over existing SEC-DED and SEC-DAEC codes.

INDEX TERMS Error correction codes, Orthogonal Latin square codes, parity check bits, SEC-DED-DAEC, SEC-DAEC, cache memory.

I. INTRODUCTION

Enhancing cache reliability against radiation-induced errors is paramount. Especially as technology advances into the nanometer regime, where the effects of miniaturization become more pronounced on bit cell integrity [1]. Traditionally, error correction codes serve as a tool to ensure the integrity of cache's stored data [2]. One widely used coding, known as single error correction double error detection (SEC-DED), has the capability to correct single errors and detect double errors within a sequence of stored bits known as a data word. This coding introduced in [3] and improved by many approaches [4], [5], [6], [7], [8]. This coding remains effective as long as the error rate remains within its tolerance capability. With the ongoing miniaturization process, the impact of errors induced by radiation, often referred to as soft errors, has become increasingly significant.

The associate editor coordinating the review of this manuscript and approving it for publication was Zihuai Lin³.

These soft errors tend to occur in adjacent bits, primarily due to the heightened density of cache cells within confined spaces operating at lower voltage levels to reduce power consumption. Observations reveal that errors induced by neutron radiation are more likely to result in double adjacent error (DAE) compared to other types of multiple errors [9].

To correct DAE, bit interleaving can be employed in conjunction with SEC DED codes. Interleaving involves constructing a logical data word rather than a physical one [10]. This approach assembles the data word from physically scattered bits, ensuring that at most one bit within a word is affected by adjacent errors. While this solution proves effective for memories designed in blocks or separate modules allowing parallel access, it may not be efficient for cache memory applications: – since it introduces overheads as interconnections become more complex and can result in an area and power overhead to duplicate when the distance between cells of the same word is large [11], [12].

Another approach is the adoption of multi-bit error correction codes, such as Reed-Solomon (RS) codes [13] or Bose-Chaudhuri-Hocquenghem (BCH) codes [14], which provide robust protection against double errors with low check bit rates. While effective, these codes often increase cache complexity, as these codes are typically applied to blocks of multiple data words at a time [9]. On the other hand, applying double error correction codes at the data word level requires a large number of extra check bits for each data word, as will be exemplified by the adoption of orthogonal Latin square (OLS) codes [15] to correct complete double errors (DEC).

To mitigate these associated costs, another approach to leverage spatial error patterns to design simpler codes for correcting DAE, often referred to as SEC-DED-DAEC codes [9], [16]. Although these codes provide a simpler design compared to multi-bit error correction, they have a limitation: the potential for miscorrection when handling nonadjacent double errors. These codes are particularly suitable for cache applications where the occurrence of double nonadjacent errors is rare (i.e. can be considered as SEC-DAEC codes). To address the limitation of miscorrection, efforts have been made in [17] to develop a SEC-DED-DAEC coding scheme. As a subclass of DEC OLS coding, the developed scheme provides codes can correct single and double adjacent errors while reliably detecting all nonadjacent double errors, thus eliminating the possibility of miscorrection. Furthermore, these codes fall under the category of one-step majority logic decodable (OSMLD) codes, known for their simplicity and fast decoding process. Additionally, this coding reduces the count of check bits stored with each data word compared to conventional DEC OLS codes, despite still requiring a substantial number of check bits.

This paper presents SEC-DED-DAEC and SEC-DAEC codes, both derived from the modified SEC OLS coding scheme. These codes offer a simple design and fast decoding, making them ideal for high-speed cache memory. Specifically, SEC-DED-DAEC codes are developed to reduce the count of check bits compared to the existing SEC-DED-DAEC codes [17], saving 1 percent of check bit storage area. The enhanced SEC-DAEC codes are designed to reduce decoding delay time, outperforming existing SEC-DAEC and SEC-DED codes.

The rest of this paper is organized as follows: *Section II* provides an overview of the Latin square code and its application to correct adjacent double errors. *Section III* details the derivation of the proposed codes and demonstrates its error correction and detection capabilities. *Section IV* presents the evaluation of the proposed codes, focusing on aspects like area, delay, and power consumption for both encoding and decoding circuits in comparison to existing codes. Finally, *Section V* presents the key conclusions drawn from the application and evaluation of the proposed codes.

II. ORTHOGONAL LATIN SQUARE CODES

Orthogonal Latin Square (OLS) was introduced decades ago as a multiple-error correction coding scheme for data [15]. This linear coding scheme requires $2tm$ check bits to protect $K = m^2$ data bits, where t is the amount of the correctable data bits, and m is an integer value (usually a power of 2 value when used for memory systems). This scheme stands out due to its modularity and its simple and high-speed decoding process. As mentioned previously, the simplicity of the decoding is attributed to their operation based on the one-step-majority decodable method, or simply *OS – MLD*, which has the fastest decoding process [18]. In simple terms, the concept of the *OLS* involves creating $2t + 1$ independent copies for every data bit. These independent copies come about by incorporating the bit into equations that produce $2t$ check bits alongside the data bit itself. This procedure guarantees the correction of t error data bits by using the majority of $t + 1$ correct copies of the data bit. Also, the OLS coding scheme ensures that miscorrections are avoided by ensuring that each of the other data bits is involved in these equations only once. These features made OLS codes highly desirable for protecting high-speed data storage and transmission systems, such as caches [19], [20] and 3D memories [21].

In high-density cache memory, as declared earlier, the errors induced by the exposure to high-energy radiation are mostly observed as DAE bits. Double error correction OLS code or simply DEC OLS can be used to control these errors. The DEC OLS code is designed to correct complete double errors in a m^2 size data word using $4m$ check bits. The H matrix of this code has been structured using four sub-matrices, denoted as M_1 , M_2 , M_3 , and M_4 , as follows:

$$H = \begin{bmatrix} M_1 & & & \\ M_2 & I_{4m} & & \\ M_3 & & & \\ M_4 & & & \end{bmatrix} \quad (1)$$

I_{4m} is the identity matrix of size $4m$. $M_1 \dots M_4$ are sub-matrices of size $m \times m^2$.

For illustration, Fig. 1 shows the the H matrix for the DEC OLS (32,16) code, serving as a representative example of the DEC OLS codes.

Employing DEC OLS code would require a large amount of check bits. For example correcting any double errors in 64 bit data word requires $4 \times \sqrt[2]{64} = 32$ bit, which need to be stored with each data word. This would challenge the feasibility of using this coding, particularly in area-constrained cache applications.

Modifying the original DEC OLS code to become a single error correction, double error detection, and double adjacent error correction (SEC DED DAEC) can be an effective approach for tolerating DAE bits. Notably, this modification reduces 25 percent of the imposed check bits, as in [17]. Table. 1 lists the required check bits for both codes. For example, protecting 64-bit data would require only 24 bits

TABLE 1. The check bit parameter of DEC OLS codes and SEC-DED-DAEC OLS codes for different data word sizes.

word size	DEC OLS	Stored check bits	SEC-DED-DAEC OLS	Stored check bits ($3m$)
16	(32,16)	16	(28,16)	12
64	(96,64)	32	(88,64)	24
256	(320,256)	64	(304,256)	48

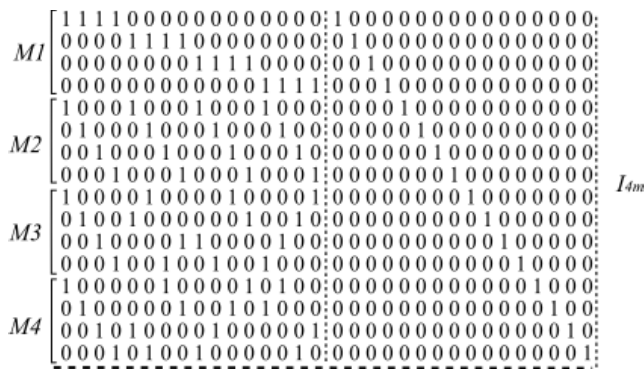


FIGURE 1. The original H matrix of the (32,16) DEC OLS code.

of stored parities instead of 32 bits for the original DEC OLS code. This reduction is achieved by removing the $M1$ sub-matrix from the original DEC OLS H matrix shown in Fig.1. The ability to correct DAE is achieved by applying the condition that no two adjacent columns in the modified H matrix can share any check bits [11]. To implement this condition, researchers in [17] redistributed the data bits and the check bits within the modified H matrix. Although effective, there is still an urgent need to further decrease the stored check bits to make this coding scheme more attractive for practical applications. In the following section, two coding approaches are proposed for correcting DAEs in high-speed cache memory.

III. PROPOSED APPROACHES

In this paper, two coding classes are proposed to correct DAE bits caused by soft errors. The first one introduces a class of SEC-DED-DAEC codes designed to correct DAEs using fewer check bits. These fast-decoding codes can also detect nonadjacent double errors without any risk of miscorrection. The second one presents straightforward SEC-DAEC codes, which are designed for faster decoding than existing SEC-DED codes and provide full protection against miscorrection.

A. PROPOSED SEC-DED-DAEC CODES

This category of SEC-DED-DAEC codes is based on a two-step coding scheme. The first step involves the derivation of the SEC DED codes from the modified SEC OLS coding scheme presented in [22]. The reconfigured matrices serve as a basis for the construction of the H matrices in the second step, which are the foundation of the proposed SEC-DED-DAEC codes.

1) SEC DED H MATRICES

As demonstrated in [22], a modified SEC H matrix with the same specifications as the SEC OLS H matrix was developed to reduce a significant percentage of the required check bits, without substantially affecting the simplicity and speed of encoding and decoding circuits. The matrix was initially devised with double-weighted, or $w = 2$, data columns, wherein each column in the data set is comprised of two distinct check bits. To facilitate the creation of codes capable of correcting one error and detecting two errors, the matrix has been reconfigured to be triple-weighted, or $w = 3$, data columns, with each data column now involving three distinct check bits. An example of the reconfigured matrix for producing the code (35,21) SEC DED is presented in the Fig. 2. It can be observed that the matrix follows the OLS two conditions (i.e., each data bit is associated with three distinct check bits, and each of these three check bits involves at most one of the other data bits). Subsequently, voting gates can be used to correct single errors in the protected bits. However, to simplify the decoding circuit and to achieve double error detection, 3 input AND gates, similar to those presented in [17], have been used to replace the voting gates.

In this paper, the focus is on the protection of data word sizes that are powers of 2, which are widely used in memory systems. To generate a matrix for a 16-bit data word, the matrix depicted in Fig. 2 is reformed by eliminating 5 columns that contain the value “1” in rows 13 and 14. This selection reduces the count of rows (i.e. check bits). The resulting matrix can then be utilized to generate a (28, 16) SEC DED code. This approach has been also used to generate matrices for 64-bit and 256-bit data words. The parameters of the resulting codes are presented in Table 2, indicating that the proposed SEC DED coding scheme reduces the count of check bits compared to the original code listed in Table 1.

The reduction is noticeable in longer data words, particularly in 64-bit the reduction reaches 1 out of 24 bits and even more in 256-bit 4 out of 48 bits. Also, the proposed coding scheme is flexible and can be implemented for any data word size, while the original OLS coding is limited to m^2 data word sizes.

TABLE 2. The check bit parameter of the proposed SEC-DED for different data word sizes.

k data bit	Modified SEC-DED	Stored check bits
16	(28,16)	12
64	(87,64)	23
256	(300,256)	44

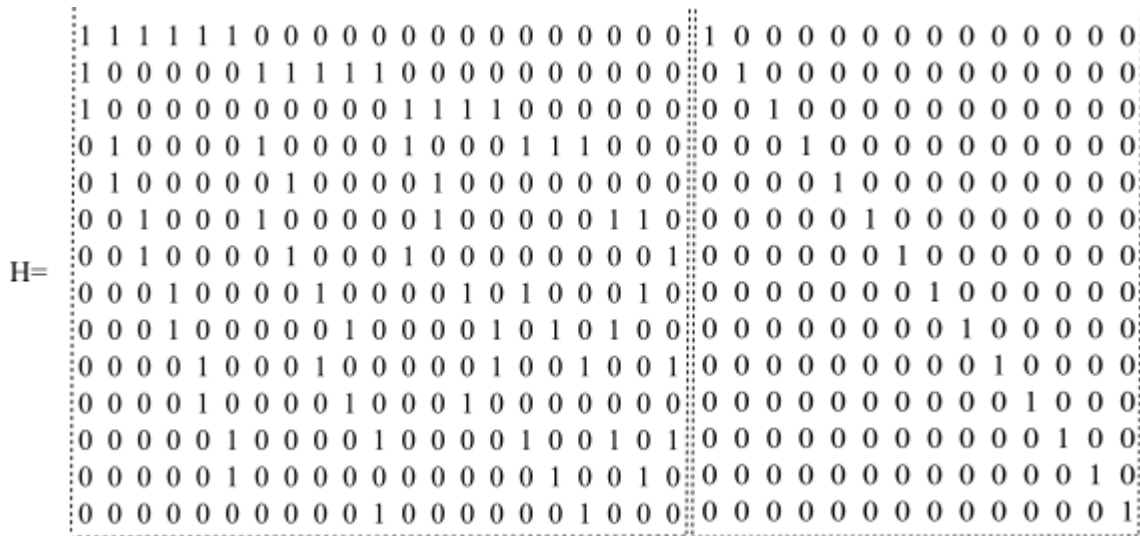


FIGURE 2. The reconfigured H matrix for the SEC DED (35,21) code.

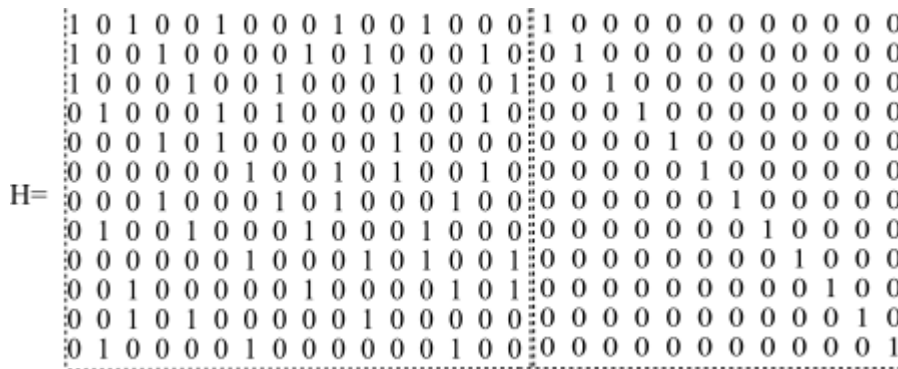


FIGURE 3. The reformed H matrix for the (28,16) SEC-DED-DAEC code.

2) PROPOSED SEC-DED-DAEC CODES

The proposed SEC-DED-DAEC codes are generated using the previously described SEC DED matrices. As shown in Fig. 3, the H matrix of the proposed (28,16) SEC-DED-DAEC code is presented as an example of the proposed SEC-DED-DAEC codes. This H matrix is constructed by reorganizing the (28,16) SEC DED matrix described in subsection III-A1. By using the derived SEC DED matrices, the proposed SEC-DED-DAEC codes offer a robust solution for correcting single and double adjacent errors. Notably, the H matrix fulfills the critical condition that no two adjacent data columns share any check bit, which is essential for accurate double adjacent error correction. As demonstrated in Fig. 3, this condition was achieved without the need for redistribution of check bits with data bits, as the case in [17]. The simplicity of implementation of the proposed codes is an advantage for VLSI and ASIC applications.

In the case of single error in the protected data word, this type of error is corrected by using the corresponding three input AND gate. The AND’s inputs consist of a distinct set of

syndrome bits. Syndrome bit is calculated by XOR the saved check bit and the recomputed check bit from the fetched data bits, which are marked with “1” in the corresponding row of the modified H matrix. For instance, the first syndrome bit, S_1 , is calculated by XOR the first stored check bit P_1 and the first recomputed check bit P'_1 . The P'_1 is the modulo-2 sum of all the involved data bits (marked as “1”) in row 1 of the proposed H matrix in Fig 3. Another example, S_2 is resulted from XOR P_2 and P'_2 , which computed by modulo-2 sum all the involved data bits in row 2 of the H matrix, and so on. According to the principles described, any single error bit can be corrected using the three corresponding syndrome bits. To illustrate this, let’s assume that d_0 is the error bit. This particular error will impact the values of the first three computed check bits (P'_1, P'_2, P'_3). As a result, the value of the first three syndrome bits (S_1, S_2, S_3) becomes 1, thus activating the related AND gate that will correct d_0 by toggling its value. This correction process can be applied to all bits within the data word.

In the occasion of a single error within the stored check bits, this error will lead to the generation of a single erroneous syndrome bit. It is important to emphasize that this single erroneous syndrome bit does not compromise the integrity of data or cause any silent error.

In the case of double error, it can occur in one of the following scenarios:

- 1) Double Error in adjacent data bits.
- 2) Double Error in nonadjacent data bits.
- 3) Double error in check bits.
- 4) Double error in one of check bits and one of data bits (adjacent).
- 5) Double error in one of check bits and one of data bits (nonadjacent).

In the first scenario, the occurrence of double errors in adjacent data bits does not result in any overlapping of parities. This facilitates the distinction and correction of these two errors. This can be clearly observed in the parity matrix illustrated in Fig. 3. Any error in adjacent bits will result in the activation of six syndrome bits. Therefore, the proposed coding scheme can be considered a corrector for adjacent double errors in data bits.

For the second scenario, the proposed coding scheme is designed to satisfy the original OLS coding conditions. This ensures that any pair of data bits does not intersect with more than one check bit. Therefore, in the event of nonadjacent double errors in the data, their effect will result in one of the following two error patterns. First, if the two erroneous bits intersect with a single check bit, this will cancel out the effect of errors on the common check bit (according to the well-known parity theory). In other words, only two of syndrome bits per error bit is activated, not three, which prevents error correction. This pattern of double errors can simply be detected by using the detection circuit proposed in [17], based on the condition of activated even number of syndrome bits without resulting any correction process. The second pattern occurs when the erroneous bits do not intersect with any check bit. For example, if the errors occur in b_0 and b_{13} in the H matrix displayed in Fig. 3. This pattern of nonadjacent double errors is correctable, meaning that the three necessary syndrome bits are available for error correction. It is noteworthy that any combination of the affected check bits in this scenario does not create any miscorrection. This is due to the orthogonality property, which prevents activating more than two syndrome bits in any data bit correction set, thereby preventing any miscorrection state. Therefore, this type of double error is either detectable or correctable.

Referring to the third scenario, the presence of two errors, whether adjacent or not, in the saved check bits will activate an even number of syndrome bits. This will trigger the double error detection circuit and identify it as a double error. Although this type of error does not result in any error, it cannot be disregarded due to its similarity in effect to the errors in the second scenario.

In scenario four, the proposed codes effectively correct the data bit. Importantly, it remains unaffected by the error present in the adjacent check bit. The reason behind this lies in the design of the proposed H matrices, which are based on OLS conditions and also ensure that there is no overlap between the adjacent check and data bits.

If the erroneous check and data bits are not adjacent, as in the fifth scenario, the data bit can be corrected provided there is no intersection between the check bit and the check bits generated by the data bit. Otherwise, if there is an overlap, the suggested codes will detect this error as a double error.

In summary, this section introduces a two-step scheme for generating SEC-DED-DAEC codes using modified orthogonal Latin squares codes. These codes have the capability to correct single and double adjacent error bits and detect double errors accurately, using less percentage of stored check bits, without any instances of miscorrections. A notable advantage of proposed codes is that they offer flexibility in implementation for protecting data of various sizes, surpassing the limitations of the conventional DEC OLS codes.

B. PROPOSED SEC-DAEC CODES

The idea of the proposed SEC-DAEC coding scheme is to use a simple H matrix with $w = 2$ data columns, and orthogonal rows. This matrix must have the following constraints:

- 1) Each column is distinct from the other columns.
- 2) Each column must be non-zero.
- 3) All columns assigned to check bits must have $w = 1$.
- 4) All columns assigned to data bits must have $w = 2$.
- 5) All columns must not have adjacent 1's.
- 6) Every pair of adjacent columns is distinct.
- 7) Every row must be orthogonal to the other rows.

The first three constraints are aimed at correcting single errors. Any single error can be detected and corrected. A single error can affect a single column, thus enabling the decoder to use a distinct syndrome pair to identify and correct the error. The fourth constraint reduces the number of check bits by involving only two check bits in each column assigned for data bits. The fifth constraint prevents consecutive check bits in any column to avoid miscorrection errors induced by adjacent errors in check bits. This can be achieved by reordering the check bits to ensure that any double adjacent error (DAE) in the check bits set will produce an invalid syndrome pair and be discarded. The sixth constraint corrects DAE in data bits by using different valid syndrome pairs for every adjacent data column. It is noteworthy that in existing codes, the occurrence of DAE in data columns may induce a miscorrection error in one or more data bits. However, in this SEC-DAEC coding class, these miscorrections are avoided using a newly proposed module called the double adjacent error correction module, as will be described later in this section. The seventh constraint ensures the fulfillment of the sixth constraint with a minimum number of data bits involved in check equations. This aims to minimize the delay in the

encoding and decoding circuits. This property is based on the orthogonal Latin square coding scheme.

TABLE 3. The check bit parameter of the proposed SEC-DAEC codes in comparison with the SEC OLS codes for different data word sizes.

k data bit	Codes	Stored check bits
16	SEC OLS (23,16) [22]	7
	Proposed SEC-DAEC (24,16)	8
64	SEC OLS (76,64) [22]	12
	Proposed SEC-DAEC (77,64)	13
256	SEC OLS (280,256) [22]	24
	Proposed SEC-DAEC (281,256)	25

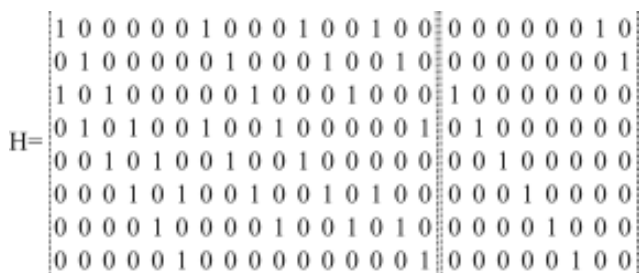


FIGURE 4. H matrix of the proposed (24,16) SEC-DAEC code.

Initially, we adopted the modified SEC OLS matrices presented in [22]. To fulfill the constraints of the proposed matrices, and in comparison with the SEC OLS codes, one additional row had to be incorporated (resulting in the inclusion of one extra check bits), as shown in table 3. Figure 4 provides a visual representation of the proposed H matrix for the proposed SEC-DAEC code (24,16) as an example of the proposed SEC-DAEC codes.

The encoding circuit is a simple circuit to generate check bits according to the check bit equations provided by the proposed H matrix. For example, the encoding equations of the proposed (24,16) SEC-DAEC code are the following:

$$\begin{aligned}
 p_0 &= d_0 \oplus d_6 \oplus d_{10} \oplus d_{13} \\
 p_1 &= d_1 \oplus d_7 \oplus d_{11} \oplus d_{14} \\
 p_2 &= d_0 \oplus d_2 \oplus d_8 \oplus d_{12} \\
 p_3 &= d_1 \oplus d_3 \oplus d_6 \oplus d_9 \oplus d_{15} \\
 p_4 &= d_2 \oplus d_4 \oplus d_7 \oplus d_{10} \\
 p_5 &= d_3 \oplus d_5 \oplus d_8 \oplus d_{11} \oplus d_{13} \\
 p_6 &= d_4 \oplus d_9 \oplus d_{12} \oplus d_{14} \\
 p_7 &= d_5 \oplus d_{15}
 \end{aligned}$$

Initially, the decoder circuits of the proposed SEC-DAEC codes are designed to generate syndrome bits. Every syndrome bit is generated by XORing the stored check bit with the check bit regenerated using the corresponding encoding equation. The single error can be located and corrected using the activated syndrome pair, which must match the assigned column in the parity matrix. This syndrome pair will activate a 2-input AND gate, which will flip the error bit with an XOR gate. This correction mechanism is applied to all protected

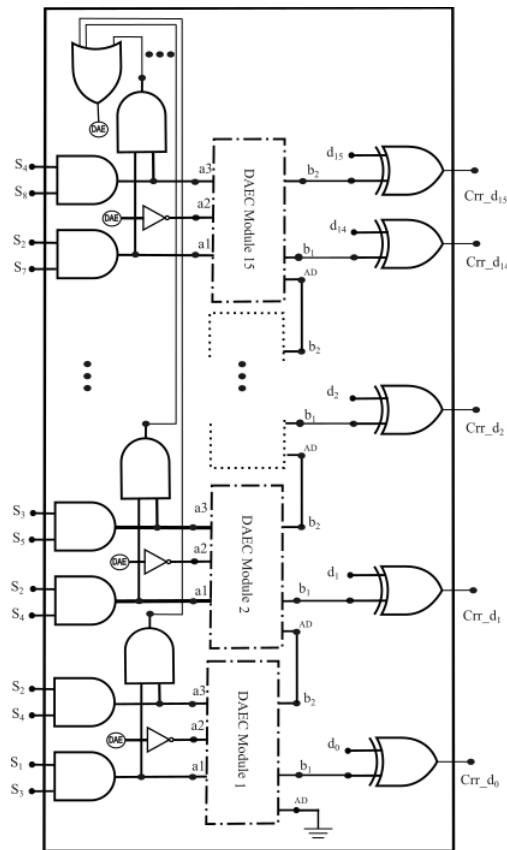


FIGURE 5. Decoder diagram of the proposed (24,16) code.

data bits. Fig. 5 shows this correction mechanism in the decoder circuit of the (24,26) SEC-DAEC code.

As mentioned earlier, the proposed H matrix is designed with $w = 2$ columns to reduce the number of encoded check bits. However, the generated codes have a hamming distance equal to 3, thus guaranteeing only the correction of single errors. In the event of an adjacent double error in the protected data bits, a single or multiple miscorrection errors may occur by mistakenly correcting non-erroneous data bits, which leads to degrading the correction of the proposed SEC-DAEC codes. Here, rather than undertaking the step of increasing the check bits to make the hamming distance 4 as it is in the existing codes, the new approach here is to mask any correction operation on any double error bits except the double adjacent bits. This masking can be achieved due to the fact that any case of double adjacent error occurs only once at a time (as per the sixth constraint) and can be detected using a simple ORing circuit of $k-1$ adjacent correction signals. The activation of the OR circuit, the DAE signal shown in Fig. 5, with a value of 1, is used to pass only the double adjacent error correction and disable any other correction signal. Fig. 6 presents the logic of the proposed Double Adjacent Error Correction (DAEC) Module used to perform the selective correction process. The expert in logic circuits can recognize that the DAEC module will pass the single error correction when the DAE single is not activated

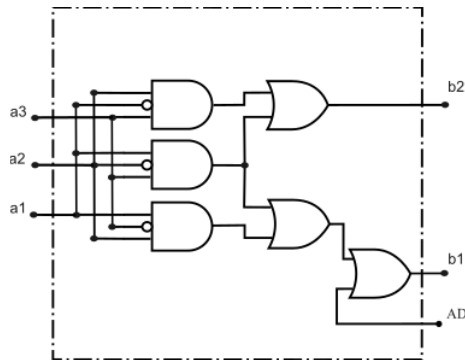


FIGURE 6. Proposed DAEC module.

and its value is 0. But when the DAE signal is activated with a 1 value, then the decoder will enable the correction of the double adjacent errors that belong to the same DAEC module (i.e., the module with active $a1$ and $a3$ signals shown in Fig. 6).

In the preliminary assessment of overheads imposed by the proposed SEC-DAEC codes and compared with simple codes that have gained industry acceptance, such as Odd Weight SEC DED by Hsiao [4], the following can be noted: The check bits used are 6 and 8, compared to 8 and 13 bits needed by the proposed SEC-DAEC code to protect 16- and 64- bits data words, respectively.

Regarding the proposed matrix's simplicity, the initial analysis was carried out using the following criteria:

- 1) The total number of 1's in the matrix
- 2) the average number of 1's per row
- 3) Maximum number of 1's in a row.

For the first criterion, the smaller the number of 1's in the matrix means that it needs fewer gates and therefore a simpler encoding and decoding circuit. As for the second criterion, the lower the average number of 1's per rows, the fewer xor gates there will be in the encoding and decoding equations. The third criterion refers to the longest encoding and decoding path, which determines the delay in both processes. Table 4 lists the parameters of Hsiao SEC DED coding matrix compared with the proposed matrix for 16- and 64-bit. From this comparison, it can be inferred that the overheads of adopting the proposed matrices will be within the acceptable limits by the industry, especially from the perspective of delay and simplicity, where they were reduced by up to 59% and 60%, respectively, when used to protect 64-bit data word. But we must keep in mind that the proposed codes imposed the addition of DAEC modules to the decoding circuit outputs to isolate unwanted corrections; this adds a burden to the delay of the decoding process. The consequences of using the proposed codes will be evaluated in more detail in the evaluation section and compared with a set of existing codes.

IV. EVALUATION

This section aims to evaluate and compare the proposed SEC-DAEC coding with Dutta's SEC-DAEC coding [9], as well as the proposed SEC-DED-DAEC coding with Pedro's

SEC-DED-DAEC coding [17]. Dutta's SEC-DAEC coding was selected for its simplicity as it is designed to be as straightforward as the widely used SEC-DED coding. Pedro's SEC-DED-DAEC coding was chosen as one significant coding derived from orthogonal Latin square codes. The focus is on assessing the overheads imposed by reducing the number of check bits on the parameters of the proposed encoding and decoding circuits. These circuits are implemented using hardware description language (HDL), and synthesized by field-programmable gate array (FPGA) and application-specific integrated circuit (ASIC) technologies.

For the proposed codes, the first one described in subsection III-A will be named as $DAEC_{Pro1}$, and the second one described in III-B as $DAEC_{Pro2}$. For the protection requirements, both the $DAEC_{Pro1}$ and $DAEC_{Pro2}$ codes are capable of correcting single and double adjacent errors in a data word without any miscorrections. Furthermore, the $DAEC_{Pro1}$ codes are capable of detecting any non adjacent double errors. The application considered is protecting 16, 64, and 256 bit data words commonly used in high speed memories. For 16 bits, the codes would be $DAEC_{Pro1}$ (28, 16) and $DAEC_{Pro2}$ (24, 16). For 64 bits, the codes would be $DAEC_{Pro1}$ (87, 64) and $DAEC_{Pro2}$ (77, 64). The codes for 256 bits would be $DAEC_{Pro1}$ (300, 256) and $DAEC_{Pro2}$ (281, 256).

A. FPGA SYNTHESIS

All the evaluated codes are synthesized using Xilinx FPGA Zynq-7000 device family [23]. The analyzed parameters are area, delay, and power consumption of the encoder/decoder circuits. Some analysis on area, delay, and power overheads will be referenced to Hsiao odd-weight-column SEC-DED scheme [4] since its overheads are widely accepted in the industry. In this case, with codes for protecting 16 and 64 bit data words.

Table 5 presents a comparison of the mentioned techniques based on the imposed extra area of the check bits, and the area utilization of the encoder/decoder circuits (in Lookup Tables, or LUTs). For the cases of short and long data words (16 and 64 bit), it can be observed that the area overheads of the check bit and the sum of encoder and decoder LUTs of the SEC-DAEC Dutta codes [9] are less than the proposed $DAEC_{Pro2}$ codes. This is reasonable, since Dutta SEC-DAEC scheme is simple. As the results indicate, the $DAEC_{Pro2}$ scheme requires a small extra area for check bits, increasing by only 6% compared to the Hsiao codes or Dutta codes when protecting 16-bit or 64-bit data words. Regarding the encoder and decoder area, the $DAEC_{Pro2}$ scheme requires 51% and 32% more LUTs than the Hsiao scheme for 16-bit and 64-bit data words, respectively. This indicates that despite the simplicity of the $DAEC_{Pro2}$ H matrix, the addition of DAEC modules increases the decoder area. However, this increase is minor compared to the area needed for storing the check bits, which are stored with each data word. Therefore, the $DAEC_{Pro2}$ codes is still a competitive solution from an area perspective when protection is required only for correcting

TABLE 4. Simplicity parameters of the proposed matrices.

K	Code	check bits	# of 1's in matrix	Average # of 1's /row	Max. # of 1's in row	# check bits	% of 1's in matrix	%Max # of 1's in row
16	(22,16) SECDED [4]	6	54	9	9			
	Proposed (24,16) SEC-DAEC	8	40	5	6	+2	-44%	-33%
64	(72,64) SECDED [4]	8	209	27	27			
	Proposed (77,64) SEC-DAEC	13	141	10.8	11	+5	-60%	-59.25%

TABLE 5. FPGA synthesis results for area measurement using LUTs and extra check Bis area.

K	Code	Enco.	Deco.	Extra LUTs	Extra Check bits area
16	Pedro (28,16) [17]	12	39	+29%	43%
	Dutta (22,16) [9]	9	28	+22%	27%
	Hsiao (22, 16) [4]	9	20	-	27%
	DAEC _{Pro1} (28,16)	12	43	+47%	43%
	DAEC _{Pro2} (24, 16)	8	51	+51%	33%
64	Pedro (88,64) [17]	39	147	+35%	27%
	Dutta (72,64) [9]	42	114	+22%	11%
	Hsiao (72, 64) [4]	39	82	-	11%
	DAEC _{Pro1} (87,64)	40	137	+32%	26%
	DAEC _{Pro2} (77,64)	26	153	+32%	17%
256	Pedro (304,256) [17]	144	542	-	16%
	DAEC _{Pro1} (300,256)	161	543	-	15%
	DAEC _{Pro2} (281,256)	101	629	-	9%

single-bit and double-adjacent-bit errors. For applications requiring higher protection capabilities, the results showed that the proposed DAEC_{Pro1} scheme is also a competitive option to Pedro et al scheme, particularly as it reduces the check bits storage area by 1 percent when protecting large data words such as 64-bit and 256-bit.

The delay measurements of the encoding and decoding processes in FPGA synthesis for the 16, 64, and 256 values of k are presented in Table 6. These measurements are determined by the slowest path in nanoseconds. Analysis of the encoders results shows that the encoding speed is related to the number of the generated check bits. This is because reducing the number of generated check bits increases the complexity of the H matrix and increases the number of 1's in the rows, making the encoding circuit more complex and slower.

This relationship is evident in the proposed DAEC_{Pro1} (28, 16) and DAEC_{Pro2}(24, 16) 16-bit encoders, where the encoding speed matched Pedro et al. (28, 16) [17] encoding and increased by 10 percent compared to the Hsiao (22, 16) [4] and Dutta (22,16) [9] encoders. This indicates that the encoding speed will not be negatively affected but will actually increase when the proposed codes are adopted. In the 64-bit encoding, the effect is similar, with an increase of 7.2 percent compared to the Hsiao (72, 64). In the 256-bit encoding, the reduction in the number of check bits in the proposed DAEC_{Pro2}(281, 256) code did not affect the encoding speed when compared with the proposed encoding in DAEC_{Pro1}(300, 256). However, compared to Pedro et. al. (304, 256) code [17], a slight increase of 7.4 percent can be observed.

Decoding time is a key factor in high speed cache memories due to its influence on the speed of the data

reading process. Table 6 displays the results for decoding time for all the evaluated codes. For the proposed DAEC_{Pro2}, the analysis shows that the delay time resulting from the addition of DAEC modules will be limited compared to the positive effect of the simplicity of the H matrix on the decoding speed. This can be observed in the decoding time of the proposed DAEC_{Pro2}(24, 16) 16-bit decoder, where it appears equal to the decoding time of the Hsiao (22, 16) SEC DED code.

The important benefit of adopting the DAEC_{Pro2} is evident when it is applied to protect 64-bit data words, as the decoding time is reduced by 24.5 percent compared to the Hsiao (72, 64) SEC DED code. This reduction is attributed to the simplicity of the H matrix and the parallel operation of the DAEC modules. For 256-bit data word protection, the DAEC_{Pro2}(281,256) code is the best option if applications only require correcting one or two adjacent errors. We would like to point out here that the reason that the decoding time in the DAEC_{Pro2}(24, 16) code is slower than the DAEC_{Pro2}(77, 64) and DAEC_{Pro2}(281, 256) codes is due to the differences in the synthesis process. Specifically, the synthesis mostly used LUT4 for the codes DAEC_{Pro2} (77, 64) and DAEC_{Pro2}(281, 256), while using LUT6 for the code DAEC_{Pro2}(24, 16) that increased the number of stages (nets) and with it increased the length of the routes and decoding time.

For applications that require SEC-DED-DAEC capability, the proposed DAEC_{Pro1}(28,16) code for 16-bit protection shows a faster decoding time than the Hsiao (22, 16) code. Also, DAEC_{Pro1}(28,16) code is competitive with Pedro et al. (28,16) SEC-DED-DAEC code, since both are based on OLS method. However, the results show that the decoding process of DAEC_{Pro1} is 6 percent and 5.6 percent faster than that of Pedro et al codes when applied to 64-bit and 256-bit protection, respectively.

Based on the obtained results, the analysis shows that adopting the proposed DAEC_{Pro2} scheme for correcting adjacent double errors is preferable when evaluated from the perspective of encoding and decoding times. The encoding time for this scheme is less than that of the Hsiao codes and Dutta codes. In terms of decoding time, the superiority of the DAEC_{Pro2} codes remains evident, especially when applied to protect long data words, as the decoding speed in DAEC_{Pro2}(77,64) code increased by 24.5 percent compared to Hsiao (72, 64) code. The DAEC_{Pro1} proposed scheme can also be a significant competitive option for protecting long data words from adjacent double errors and detecting nonadjacent ones, as it outperforms

TABLE 6. FPGA synthesis results for delay time (IN ns).

K	Code	Enco.	Deco.	Deco. Extra Delay
16	Pedro (28,16) [17]	5.351	6.428	-10.8%
	Dutta (22,16) [9]	5.935	7.241	+0.4%
	Hsiao (22, 16) [4]	5.961	7.211	-
	DAEC _{Pro1} (28,16)	5.351	6.459	-10.4%
	DAEC _{Pro2} (24, 16)	5.351	7.211	0%
64	Pedro (88,64) [17]	5.924	7.958	-9.6%
	Dutta (72,64) [9]	6.623	8.524	-3.2%
	Hsiao (72, 64) [4]	6.388	8.806	-
	DAEC _{Pro1} (87,64)	5.924	7.480	-15.1%
	DAEC _{Pro2} (77,64)	5.924	6.646	-24.5%
256	Pedro (304, 256) [17]	5.925	10.497	
	DAEC _{Pro1} (300,256)	6.403	9.909	
	DAEC _{Pro2} (281,256)	6.377	6.681	

Pedro et al codes in decoding speed while using fewer check bits.

Table 7 lists the results of power consumption in the encoding and decoding circuits. These results are the sum of the static and dynamic power consumption (in milliwatts) based on the FPGA synthesis. About the encoder, the analysis shows that the DAEC_{Pro2} coding is acceptable from the power consumption perspective compared to Hsiao coding or Dutta coding. This is proved by the consumption of the 16-bit encoder of the DAEC_{Pro2}(24, 16) code, and even the 64-bit encoder of the DAEC_{Pro2}(77, 64) code, which requires a slight increase of only 9 percent compared to the consumption in Hsiao (72, 64) code. As for the encoder of the DAEC_{Pro1}, the increase is slightly up to 7 percent for the 16-bit encoder, but for the 64-bit encoder, the increase in the generated check bits will consume more power than the 64-bit Hsiao encoder by 17 percent. However, the power consumed in the encoder in the DAEC_{Pro1}(28, 16) code is almost equal to that imposed by the encoder in Pedro (28,16) code. This can be justified because both schemes rely on the orthogonal Latin square mechanism to generate an equal number of check bits. From this, it is possible to understand the slight decrease in encoder’s power consumption in the proposed DAEC_{Pro1}(87, 64) code with a size of 64-bit and encoder in the DAEC_{Pro1}(300, 256) code with a size of 256-bit, resulting from the lower number of generated check bits compared to Pedro (88, 64), and Pedro (304, 256) codes.

Table 7 shows the results of power consumption in the decoder circuits. The results of the proposed DAEC_{Pro2} decoder show a similarity with the results of the Hsiao decoder. This is proved by the consumption of DAEC_{Pro2}(24, 16) code, which consumes 3.4 percent less power, as well as the consumption of DAEC_{Pro2}(77, 64) code, which consumes 2.7 percent more power compared to the corresponding Hsiao codes. This slight variation indicates that the simplicity of the syndrome generation circuit reduces the effects of the power consumption of the double adjacent error detection circuit and the DAEC modules. As for the proposed decoder in the DAEC_{Pro1} scheme, the consumption is almost similar to Pedro scheme decoder. This is because, as mentioned earlier, both schemes

TABLE 7. Power requirements (IN mWatt).

K	Codes	Enco.	Deco.	Deco. Power
16	Pedro (28,16) [17]	91	112	+22.3%
	Dutta (22,16) [9]	83	81	-6.8%
	Hsiao (22, 16) [4]	84	87	-
	DAEC _{Pro1} (28,16)	91	112	+22.3%
	DAEC _{Pro2} (24, 16)	83	84	-3.4%
64	Pedro (88,64) [17]	106	146	+23.9%
	Dutta (72,64) [9]	92	87	-21.6%
	Hsiao (72, 64) [4]	87	111	-
	DAEC _{Pro1} (87,64)	105	145	+23.4%
	DAEC _{Pro2} (77,64)	96	114	+2.7%
256	Pedro (304,256) [17]	135	292	
	DAEC _{Pro1} (300,256)	132	267	
	DAEC _{Pro2} (281,256)	111	222	

are based on the orthogonal Latin square method. It is worth noting that the power consumption is proportional to the number of regenerated check bits in the decoder circuit. This can be proved by the slight decrease in the power consumption results of the proposed DAEC_{Pro1}(87,64) and DAEC_{Pro1}(300,256) codes noted compared to Pedro (88,64) and Pedro (304,256) codes, respectively.

The results obtained from the FPGA synthesis illustrate that the decoding circuits of the proposed codes, especially for long data words (64-bit and 256-bit), exhibit superior speed compared to the current codes. Another significant benefit is that the proposed codes reduce the count of check bits required for encoding. These improvements render the proposed codes highly attractive for high-speed cache designers. Moreover, these codes offer implementation flexibility, enabling them to protect data of various sizes and surpass the limitations of the original DEC OLS codes. It is important to note that while these advantages are considerable, there is a slight trade-off in terms of an increase in encoder and decoder area. However, this trade-off is overshadowed by the significant benefits they bring in terms of decoding speed, implementation flexibility, and reduction of check bit.

B. ASIC SYNTHESIS

Generally, encoder and decoder circuits are integrated into the cache and implemented using ASIC technology. For ASIC evaluation, the existing DAEC codes, including the proposed DAEC_{Pro1} and DAEC_{Pro2} codes, are synthesized for the 45nm FreePDK Standard Cell Library [24]. For SEC-DAEC, the results of the proposed 16-bit and 64-bit DAEC_{Pro2} codes are compared with Dutta codes [9] to determine the reduction in area and operation speed (encoding and decoding times). As for SEC-DED-DAEC, the 16-bit, 64-bit, and 256-bit DAEC_{Pro1} codes are compared with the corresponding Pedro SEC-DED-DAEC codes [17] to evaluate the overheads imposed in area and decoding time as a result of decreasing the required check bits. The results for the area (in μm^2) and delay time (in ns) for both encoder (Enc) and decoder (Dec) circuits are presented in Tables 8 and 9.

TABLE 8. ASIC synthesis results for area measurement μm^2 .

K	Dutta SEC-DAEC [9]		Proposed DAEC _{Pro2}		Reduction	
	Enc	Dec	Enc	Dec	Enc	Dec
16	54.3	129.5	38.3	125.5	29.5%	3.0%
64	312	517.6	183.5	340.5	41.2%	34.2%
256	-	-	777	1382	-	-

(a) Encoder and decoder area reduction for the proposed DAEC_{Pro2} codes

K	Pedro SEC-DED-DAEC [17]		Proposed DAEC _{Pro1}		Overheads	
	Enc	Dec	Enc	Dec	Enc	Dec
16	57.4	162.5	57.4	162.8	0%	0.18%
64	266.5	619	268	618	0.6%	-0.2%
256	1150	2371	1155	2383	0.4%	0.5%

(b) Encoder and decoder area overheads for the proposed DAEC_{Pro1} codes.**TABLE 9. ASIC synthesis results for delay time (IN ns).**

K	Dutta SEC-DAEC [9]		Proposed DAEC _{Pro2}		Reduction	
	Enc	Dec	Enc	Dec	Enc	Dec
16	0.224	0.513	0.168	0.41	25.0%	20.0%
64	0.335	0.678	0.212	0.458	36.7%	32.4%
256	-	-	0.301	0.483	-	-

(a) Encoder and decoder delay time reduction for the proposed DAEC_{Pro2} codes

K	Pedro SEC-DED-DAEC [17]		Proposed DAEC _{Pro1}		Overheads	
	Enc	Dec	Enc	Dec	Enc	Dec
16	0.127	0.52	0.175	0.525	27.4%	1.0%
64	0.214	0.728	0.219	0.634	2.3%	-13.0%
256	0.268	0.751	0.268	0.77	0%	2.5%

(b) Encoder and Decoder delay time overheads for the proposed DAEC_{Pro1} codes.

For area, Table 8.(a) shows the estimated reduction in encoder and decoder areas when adopting the proposed DAEC_{Pro2} codes compared to Dutta's SEC-DAEC codes. Significant reductions in area are achieved for encoders 16-bit and 64-bit by 29.5 percent and 41.2 percent, respectively, which proves the simplicity of the proposed DAEC_{Pro2} encoders. More importantly are the reductions achieved for Decoders 16-bit and 64-bit that evident the addition of the DAEC modules in DAEC_{Pro2} decoders will not have a significant effect on the consumed area.

Table 8.(b) provides the area overhead by the proposed DAEC_{Pro1} codes compare to Pedro SEC-DED-DAEC codes. Very limited overheads are found by the results of the 16, 64, and 256 bit DAEC_{Pro2} codes' encoder and decoder (less than 1 percent for all cases). Hence, From area perspective, adopting the DAEC_{Pro1} codes is comparable to Pedro SEC-DED-DAEC codes.

Table 9.(a) presents the reduction in encoding and decoding times when adopting the proposed DAEC_{Pro2} SEC-DAEC coding compared to Dutta SEC-DAEC coding. The results prove that the proposed DAEC_{Pro2} coding is capable to protect 16-, 64-, and even 256- bit data words in caches operate at 0.5 ns clock rate, which is beyond the capability of Dutta coding even for the simple 16-bit code. This enhancement is achieved by the simple design

of the decoder circuit, which comes with a small increase in check bits storing area as explained earlier. Therefore, for cache applications that require SEC-DAEC tolerance technique to operate at 0.5 ns clock rate, adopting the proposed DAEC_{Pro2} codes is an important option for cache designers.

Table 9.(b) lists the overhead in encoding and decoding time imposed by the proposed DAEC_{Pro1} codes compared to Pedro SEC-DED-DAEC codes. Analysis of the results shows that adopting the proposed DAEC_{Pro1} coding for protecting 64-bit or 256-bit is an attractive option for cache designers, since the overhead in encoding and decoding time is small in 256-bit and even the decoding time is significantly faster by 13 percent in 64-bit code. This proves that adopting the DAEC_{Pro1} codes for protecting 64-bit and 256-bit data words is beneficial specially it can save 1 percent of the check bits storage area.

V. CONCLUSION

In this paper, two approaches for double adjacent error correction have been developed based on modified SEC OLS codes. The first one is a class of SEC-DED-DAEC codes, as the second is one of SEC-DAEC codes. These approaches are tailored for high-speed cache memories, with the objective of controlling radiation-induced errors while reducing the decoding delay as much as possible with the least number of check bits. These codes have been evaluated using FPGA and ASIC technologies. For those, the developed SEC-DED-DAEC codes outperform the existing codes when applied to protect a 64-bit data word, wherein a reduction of 1% in check bits storage area is evident. These codes also achieve 13% acceleration in the decoding process. As for the proposed SEC-DAEC codes, the key finding is that the proposed SEC-DAEC codes in ASIC with 45nm technology show the capability to protect 16, 64, and 256 bit data words at 0.5ns clock rate, outperform in minimum point by 20% the existing SEC DED and SEC-DAEC codes. This achievement comes with a slight overhead in the check-bit storage area, not exceeding 6% in protecting 16- and 64- bits data words. These proposed approaches not only improve performance but also provide scalable solutions as the future cache and memory systems evolve toward higher speed technologies. Future work could investigate the integration of these approaches into other high-speed memory systems, like 3D memories, to further boost efficiency.

REFERENCES

- [1] L. W. Massengill, B. L. Bhuvu, W. T. Holman, M. L. Alles, and T. D. Lovelless, "Technology scaling and soft error reliability," in *Proc. IEEE Int. Rel. Phys. Symp. (IRPS)*, Apr. 2012, pp. 3C.1.1–3C.1.7.
- [2] R. A. Ahmed, K. Samsudin, and F. Z. Rokhani, "A framework for system dependability validation under the influence of intrinsic parameters fluctuation," *Int. J. Softw. Eng. Appl.*, vol. 8, no. 3, pp. 233–250, 2014.
- [3] R. W. Hamming, "Error detecting and error correcting codes," *Bell Syst. Tech. J.*, vol. 29, no. 2, pp. 147–160, Apr. 1950.
- [4] 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.

- [5] P. Reviriego, S. Pontarelli, J. A. Maestro, and M. Ottavi, "A method to construct low delay single error correction codes for protecting data bits only," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 32, no. 3, pp. 479–483, Mar. 2013.
- [6] S. Cha and H. Yoon, "Efficient implementation of single error correction and double error detection code with check bit pre-computation for memories," *J. Semicond. Technol. Sci.*, vol. 12, no. 4, pp. 418–425, Dec. 2012.
- [7] V. Gherman, S. Evain, N. Seymour, and Y. Bonhomme, "Generalized parity-check matrices for SEC-DED codes with fixed parity," in *Proc. IEEE 17th Int. On-Line Test. Symp.*, Jul. 2011, pp. 198–201.
- [8] R. A. Ahmed, "Error control codes based modified orthogonal Latin squares for high-speed cache memories," in *Proc. IEEE 16th Dallas Circuits Syst. Conf. (DCAS)*, Apr. 2023, pp. 1–6.
- [9] 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.
- [10] J. Maiz, S. Hareland, K. Zhang, and P. Armstrong, "Characterization of multi-bit soft error events in advanced SRAMs," in *IEDM Tech. Dig.*, 2003, pp. 21.4.1–21.4.4.
- [11] J. Li, P. Reviriego, L. Xiao, Z. Liu, L. Li, and A. Ullah, "Low delay single error correction and double adjacent error correction (SEC-DAEC) codes," *Microelectron. Rel.*, vol. 97, pp. 31–37, Jun. 2019.
- [12] S. Baeg, S. Wen, and R. Wong, "SRAM interleaving distance selection with a soft error failure model," *IEEE Trans. Nucl. Sci.*, vol. 56, no. 4, pp. 2111–2118, Aug. 2009.
- [13] 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.
- [14] R. C. Bose and D. K. Ray-Chaudhuri, "On a class of error correcting binary group codes," *Inf. Control*, vol. 3, no. 1, pp. 68–79, Mar. 1960.
- [15] M. Y. Hsiao, D. C. Bossen, and R. T. Chien, "Orthogonal Latin square codes," *IBM J. Res. Develop.*, vol. 14, no. 4, pp. 390–394, Jul. 1970.
- [16] Z. Ming, X. Li Yi, and L. Hong Wei, "New SEC-DED-DAEC codes for multiple bit upsets mitigation in memory," in *Proc. IEEE/IFIP 19th Int. Conf. VLSI Syst.-Chip*, Oct. 2011, pp. 254–259.
- [17] P. Reviriego, S. Pontarelli, A. Evans, and J. A. Maestro, "A class of SEC-DED-DAEC codes derived from orthogonal Latin square codes," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 23, no. 5, pp. 968–972, May 2015.
- [18] S. Lin and G. Markowsky, "On a class of one-step majority-logic decodable cyclic codes," *IBM J. Res. Develop.*, vol. 24, no. 1, pp. 56–63, Jan. 1980.
- [19] A. Das and N. A. Touba, "Online correction of hard errors and soft errors via one-step decodable OLS codes for emerging last level caches," in *Proc. IEEE Latin Amer. Test Symp. (LATS)*, Mar. 2019, pp. 1–6.
- [20] A. R. Alameldeen, Z. Chishti, C. Wilkerson, W. Wu, and S.-L. Lu, "Adaptive cache design to enable reliable low-voltage operation," *IEEE Trans. Comput.*, vol. 60, no. 1, pp. 50–63, Jan. 2011.
- [21] A. Sánchez-Macián, F. Garcia-Herrero, and J. A. Maestro, "Reliability of 3D memories using orthogonal Latin square codes," *Microelectron. Rel.*, vol. 95, pp. 74–80, Apr. 2019.
- [22] R. A. Ahmed, "Modified single error correction orthogonal Latin square scheme to reduce parity check bits," *Microprocessors Microsyst.*, vol. 94, Oct. 2022, Art. no. 104676.
- [23] "Zynq-7000 SoC data sheet: Overview," Xilinx Inc., San Jose, CA, USA, Tech. Rep. DS190, v1.11.1, Jul. 2018. [Online]. Available: <https://docs.amd.com/v/u/en-US/ds190-Zynq-7000-Overview>
- [24] J. E. Stine, I. Castellanos, M. Wood, J. Henson, F. Love, W. R. Davis, P. D. Franzon, M. Bucher, S. Basavarajaiah, J. Oh, and R. Jenkal, "FreePDK: An open-source variation-aware design kit," in *Proc. IEEE Int. Conf. Microelectron. Syst. Educ. (MSE)*, Jun. 2007, pp. 173–174.



RABAH ABOOD AHMED received the B.Sc. degree in computer engineering and the M.Sc. degree in software engineering from the Al-Rasheed College of Engineering and Science, University of Technology, Baghdad, Iraq, in 1996 and 2005, respectively, and the Ph.D. degree in computer systems engineering from Universiti Putra Malaysia, in May 2013. He is currently a member of the Department of Automation and Artificial Intelligent Engineering, College of Information Engineering, Al-Nahrain University. His major interests include computer systems analysis, memory fault tolerance, and computer networks.



KHAIRULMIZAM SAMSUDIN received the B.Eng. degree in computer engineering from Universiti Putra Malaysia, in 2002, and the Ph.D. degree in electronics and electrical engineering from the University of Glasgow, in 2006. He is currently a Distinguished Member with the Department of Computer and Communication Systems, Faculty of Engineering, Universiti Putra Malaysia. His research interests include embedded systems, computer security, robotics, and artificial intelligence.

• • •