Extending 3-bit Burst Error-Correction Codes With Quadruple Adjacent Error Correction

Jiaqiang Li, *Student Member, IEEE*, Pedro Reviriego, *Senior Member, IEEE*, Liyi Xiao, *Member, IEEE*, Costas Argyrides, *Senior Member, IEEE*, and Jie Li

***Abstract*— The use of error-correction codes (ECCs) with advanced correction capability is a common system-level strategy to harden the memory against multiple bit upsets (MBUs). Therefore, the construction of ECCs with advanced error correction and low redundancy has become an important problem, especially for adjacent ECCs. Existing codes for mitigating MBUs mainly focus on the correction of up to 3-bit burst errors. As the technology scales and cell interval distance decrease, the number of affected bits can easily extend to more than 3 bit. The previous methods are therefore not enough to satisfy the reliability requirement of the applications in harsh environments. In this paper, a technique to extend 3-bit burst error-correction (BEC) codes with quadruple adjacent error correction (QAEC) is presented. First, the design rules are specified and then a searching algorithm is developed to find the codes that comply with those rules. The *H* matrices of the 3-bit BEC with QAEC obtained are presented. They do not require additional parity check bits compared with a 3-bit BEC code. By applying the new algorithm to previous 3-bit BEC codes, the performance of 3-bit BEC is also remarkably improved. The encoding and decoding procedure of the proposed codes is illustrated with an example. Then, the encoders and decoders are implemented using a 65-nm library and the results show that our codes have moderate total area and delay overhead to achieve the correction ability extension.**

***Index Terms*— Burst error-correction codes (ECCs), ECC, multiple bit upset (MBU), memory, quadruple adjacent error correction (QAEC).**

I. Introduction

RELIABILITY is an important requirement for space applications [1]. Memories as the data storing components play a significant role in the electronic systems. They are widely used in the system on a chip and application-specific integrated circuits [2], [3]. In these applications, memories account for a large portion of the circuit area [4]. This makes memories suffer more space radiation than other components. Therefore, the sensitivity to radiation of memories has become a critical issue to ensure the reliability of electronic systems. In modern static random access memories (SRAMs), radiation-induced soft errors in the form of the single event upset (SEU) and multiple bit upset (MBU) are two prominent single event effects [5]. As semiconductor technology develops from the submicrometer technology to the ultradeep submicrometer (UDSM) technology, the size of memory cells is smaller and more cells are included in the radius affected by a particle [6], [7] as shown in Fig. 1.

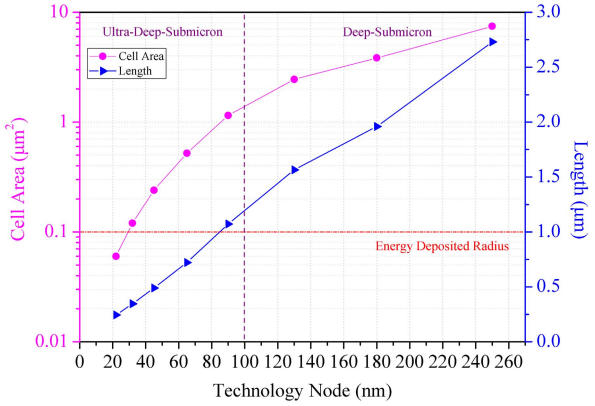

Fig. 1. Memory cell area of different technology (cell area shape is simplified to a square, and Length is the length of side).

When a particle from a cosmic ray hits the basic memory cell, it generates a radial distribution of electron–hole pairs along the transport track [8]. These generated electron–hole pairs can cause soft errors by changing the values stored in the memory cell leading to data corruption and system failure [9]. For the transistors with a large feature size, a radiation event just affects one memory cell, which means that only the SEU occurs. In this case, the use of single error-correction (SEC)-double error-detection (DED) codes [10] is enough to protect the memory from radiation effects.

As the feature size enters into DSM range, the critical charge keeps decreasing and the area of the memory cell scales down for each successive technology node. This makes more memory cells affected by a particle hit as shown in Fig. 2. For the CMOS bulk technology, with the cell-to-cell spacing decreasing, the electron–hole pairs generated in the substrate can diffuse to nearby cells and induce MBUs [11]–[14]. This

Manuscript received June 4, 2017; revised August 25, 2017 and October 10, 2017; accepted October 17, 2017. Date of publication November 8, 2017; date of current version January 19, 2018. This work was supported in part by the Fundamental Research Funds for the Central Universities under Grant HIT.KISTP.201404, in part by the Harbin Science and Innovation Research Special Fund under Grant 2015RAXXJ003, in part by the Special Found for Development of Shenzhen Strategic Emerging Industries under Grant JCYJ20150625142543456, and in part by the Spanish Ministry of Economy and Competitiveness under Grant ESP2014-54505-C2-1-R. *(Corresponding author: Liyi Xiao.)*

J. Li, L. Xiao, and J. Li are with the Microelectronics Center, Harbin Institute of Technology, Harbin 150001, China (e-mail: jiaqiang.li@cern.ch; xiaoly@hit.edu.cn; 13s021064@hit.edu.cn).

P. Reviriego is with the ARIES Research Center, Universidad Antonio de Nebrija, C.P. 28040 Madrid, Spain (e-mail: previrie@nebrija.es).

C. Argyrides is with the Radeon Technologies Group, Advanced Micro Devices, Inc., Sunnyvale, CA 94088 USA (e-mail: caa@ieee.org).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVLSI.2017.2766361

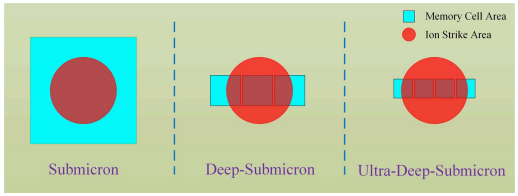

Fig. 2. Schematic description of memory cells included in the radiation effect with variation of the technology node.

compares with the FDSOI technology, which isolates transistors and limits the multicollection mechanism. Therefore, the multicollection mechanism is more prominent for a bulk technology, and the MBU probability is higher [15]–[18]. To protect against MBUs, ECCs that correct adjacent bit errors [19]–[24] or multiple bit errors [25]–[27] are utilized. Although multiple bit error-correction codes (ECCs) can correct multiple bit errors in any error patterns not limited to the adjacent bits, the complexity of the decoding process and the limitation of the code block size limit their use. Meanwhile, from the generation principle of MBUs in [28], the type of the MBUs depends on the initial angle of incidence and scattering angle of the secondary particles. Based on this, adjacent bit errors are dominant error patterns among the multiple bit errors. Therefore, adjacent bits correction ECCs become popular in memory-hardened designs. Many codes are proposed, and the capability of adjacent bits correction mainly focuses on the double adjacent error correction (DAEC) [19]–[22], triple adjacent error correction (TAEC), and 3-bit burst error-correction (BEC) [23]. An alternative to codes that can correct adjacent errors is to use an SEC or SEC-DED code combined with an interleaving of the memory cells. Interleaving ensures that cells that belong to the same logical word are placed physically apart. This means that an error on multiple adjacent cells affects multiple words each having only one bit error that can be corrected by an SEC code. As noted in previous studies, interleaving makes the interconnections and routing of the memory more complex and it will lead to an increase area and power consumption or limitations in the aspect ratio [19], [20]. Therefore, whether it is better to use SEC plus interleaving or a code that can correct adjacent errors will be design-dependent and both alternatives are of interest.

As the technology comes to the UDSM, the area of the memory cell keeps further decreasing and even memories having atomic-dimension transistors appear. The ionization range of ions with the order of magnitude in micrometer can include more memory cells in the word-line direction as shown in Fig. 2 than the three bits previously considered [29]. This means that the SEC-DAEC-TAEC codes may not be effective to ensure the memory reliability. Codes with more advanced correction ability are demanded. For example, codes designed with low redundancy for SEC-DAEC-TAEC and 3-bit BEC are presented in [23]. Therefore, extending the correction ability to quadruple adjacent bit errors would be of interest, especially if it can be done without adding extra parity bits.

In this paper, we present an improvement of 3-bit BEC codes to also provide quadruple adjacent error correction (QAEC). The code design technique for the QAEC with low redundancy is specified from two aspects: 1) error space satisfiability; and 2) unique syndrome satisfiability. Codes with QAEC for 16, 32, and 64 data bits are presented. From the view of the integrated circuits design, two criteria have been used to optimize the decoder complexity and decoder delay at the ECCs level: 1) minimizing the total number of ones in the parity check matrix and 2) minimizing the number of ones in the heaviest row of the parity check matrix. Additionally, based on the traditional recursive backtracing algorithm, an algorithm with the function of weight restriction and recording the past procedure is developed. The new algorithm not only reduces the cost of program run time, but also remarkably improves the performance of previous 3-bit BEC codes. The encoders and decoders for the QAEC codes are implemented in Verilog hardware description language (HDL). Area overhead and delay overhead are obtained by using a TSMC bulk 65-nm library. Compared with the previous 3-bit BEC codes, the area and delay overhead is moderate to achieve the correction ability extension.

The rest of this paper is organized as follows. In Section II, the rules of binary block linear codes for encoding and decoding are briefly described. The detailed code design technique for QAEC codes is presented in Section III. In Section IV, the improved searching algorithm is presented and the searching tool is developed. In Section V, the QAEC codes for 16, 32, and 64 data bits are presented. The procedure of encoding and decoding for QAEC codes is discussed and illustrated with an example in Section VI. In terms of area and delay overhead, the comparison with previous 3-bit BEC codes is performed in Section VII that also discusses the benefits in reliability. Finally, conclusions are summarized in Section VIII.

II. Binary Block Linear Codes

In previous works, codes for SEC-DAEC-DED, SEC-DAEC-TAEC, and 3-bit BEC have been proposed. All of them are binary linear block codes. The process used to design these codes is based on some rules for linear block codes construction. In this paper, the proposed codes are also binary linear block codes and obey similar construction rules. Normally, the binary codes are described by the number of data-bits, k, redundancy bits, $(n-k)$, and the block size of the encoded-word, n. An (n,k) code is defined by its generator matrix G or parity check matrix H in

$$G = [P_{k\times(n-k)} \cdot I_{k\times k}] \quad H = [P^T \cdot I_{(n-k)}] \tag{1}$$

where $I_{k\times k}$ is the identity matrix, P is the matrix with size $k \times (n-k)$, and P^T is the transpose of P. In the encoding process, the generator matrix G is used to encode the data bits through the process in

$$v = u \cdot G \tag{2}$$

where $u(u_0, u_1, \ldots, u_{k-1})$ are the data bits to be encoded, and $v(v_0, v_1, \ldots, v_{n-1})$ is the codeword. In the decoding process, the parity check matrix H is used to decode the received codeword through the process in

$$S = r \cdot H^T \tag{3}$$

where $r(r_0, r_1, \ldots, r_{n-1})$ is the received codeword, $S(s_0, s_1, \ldots, s_{n-k-1})$ is the syndrome, and a significant parameter for correcting errors. The errors injected into the received code can be described by using

$$r = v + e \quad (e_0, e_1, \ldots, e_{n-1}) \tag{4}$$

where e $(e_0, e_1, \ldots, e_{n-1})$ is the error vector indicating that an error occurs in the i_{th} bit when $e_i = 1$. When multiple bit errors occur in the received codeword r with the error vector $e = (e_0, e_1, \ldots, e_{n-1})$, the syndrome of the code considering the error vector in decoding process can be calculated by the method in

$$S = e \cdot H^T. \tag{5}$$

This equation formulates the relationship between the syndrome and the corresponding error pattern. Considering the detailed structure of the parity matrix H, when one error occurs in the i_{th} bit, the corresponding syndrome is equal to the i_{th} column vector. When errors occur in the i_{th} bit and the j_{th} bit, the corresponding syndrome is equal to the xor result of the i_{th} column vector and the j_{th} column vector. Therefore, if one error can be corrected or detected, it obeys the following rules.

1) *Correctable Restriction:* The corresponding syndrome vector is unique in the set of the syndromes.
2) *Detectable Restriction:* The corresponding syndrome vector is nonzero.

III. Code Design Technique

In this section, we discuss the code design technique for 3-bit BEC with QAEC. The approach used is based on syndrome decoding and the analysis of the requirements in terms of parity check bits and the formulation of the problem is similar to the one used in some previous studies like [23] and [30]. The process used to design QAEC codes can be divided into error space satisfiability problem and unique syndrome satisfiability. Section III-A explains the error space satisfiability problem, and Section III-B is the unique syndrome satisfiability problem.

A. Error Space Satisfiability

For a code with k data bits and c check bits, its input can represent 2^k binary values. If one error occurs, it has $k+c$ bit positions for a single error with output space of $(k+c)\cdot 2^k$ values. If adjacent errors occur, it has $k+c-1$ bit positions for double adjacent errors with output space of $(k+c-1)\cdot 2^k$ values, $(k+c-2)$ bit positions for triple adjacent errors with output space of $(k+c-2)\cdot 2^k$ values, $\ldots$, $(k+c-(N-1))$ bit positions for N adjacent errors with output space of $(k+c-(N-1))\cdot 2^k$ values. If almost adjacent errors occur, it has $(k+c-2)$ bit positions for errors in 3-bit window with output space of $(k+c-2)\cdot 2^k$ values, $(k+c-3)$ bit positions for each type of errors in 4-bit window with output space of $(k+c-3)\cdot 2^k$ values. To obtain the codes that can correct the errors with certain fault types, the sum of the output space value of error patterns should be less than or equal to the whole output space value 2^{k+c}.

TABLE I
Minimum Check Bit and Syndrome Condition

Data Bits	Check Bits	Total Bits	Syndromes Needed	Syndromes Available	Syndromes Left
16	7	23	107	120	13
32	8	40	192	247	55
64	9	73	357	502	145

For the proposed code, to correct 3-bit burst and quadruple adjacent errors, the total condition of the error patterns is $(k+c-3)+(k+c-2)+(k+c-1)+(k+c)+(k+c-2)$, respectively, for quadruple adjacent errors, triple adjacent errors, double adjacent errors, single errors, and 3-bit almost adjacent errors. Based on the error space satisfiability principle, the relation between the space of the correct codeword and the space of the erroneous codeword can be derived from

$$2^k(5(k+c)-8+1) \le 2^{k+c}. \tag{6}$$

Based on (6), the minimum number of check bits used for 16, 32, and 64 data bits is shown in Table I. Meanwhile, the available syndromes and the ones needed in the best case to correct 3-bit burst and quadruple adjacent error are also shown in Table I.

Here, we should note that the number of parity check bits needed is the same as for 3-bit BEC codes [23].

In this section, we discussed the error space satisfiability issue of the correctable errors. The number of check parity bits used for the proposed code should meet the requirement of (6). This restricts the check bit number and the minimum size of the dimension of H matrix. The issue of the column vector selection for the unique syndrome of a correctable error is discussed in Section III-B.

B. Unique Syndrome Satisfiability

From the view of binary block linear codes, if a type of error patterns can be corrected, the syndrome of individual error patterns should be unique. For the proposed code, the error patterns are $(\ldots, 1, \ldots)$ for SEC, $(\ldots, 11, \ldots)$ for DAEC, $(\ldots, 111, \ldots)$ for TAEC, $(\ldots, 1111, \ldots)$ for QAEC, and $(\ldots, 101, \ldots)$ for 3-bit almost adjacent errors correction. Therefore, the unique syndrome satisfiability can be expressed by

$$S_{0i} \neq S_{0j}, \quad S_{1i} \neq S_{1j}, \quad S_{2i} \neq S_{2j} \tag{7}$$

$$S_{3i} \neq S_{3j}, \quad S_{4i} \neq S_{4j} \tag{8}$$

$$S_{0i} \neq S_{1j} \neq S_{2k} \neq S_{3l} \neq S_{4m} \tag{9}$$

where S_{0i} is the syndrome for single bit error, S_{1i} is the syndrome for double adjacent bit errors, S_{2i} is the syndrome for triple adjacent bit errors, S_{3i} is the syndrome for 3-bit almost adjacent bit errors, and S_{4i} is the syndrome for quadruple adjacent bit errors. The syndrome variables S_{0i}, S_{1i}, S_{2i}, S_{3i}, and S_{4i} are linear combinations of the H matrix columns

obeying the rules in

$$S_{0i} = h_i \tag{10}$$
$$S_{1i} = h_i \oplus h_{i-1} \tag{11}$$
$$S_{2i} = h_i \oplus h_{i-1} \oplus h_{i-2} \tag{12}$$
$$S_{3i} = h_i \oplus h_{i-2} \tag{13}$$
$$S_{4i} = h_i \oplus h_{i-1} \oplus h_{i-2} \oplus h_{i-3} \tag{14}$$

where $i, j \in [1, n]$, $i \neq j$. Equations (10)–(14) indicate the detail relation between the syndromes and the columns. It is also used to design the circuits of syndrome calculation block.

The code design is a kind of Boolean satisfiability problem. Normally, the solution of this problem is based on the recursive backtracing algorithm, which is presented in Section IV. Here, from the view of the integrated circuits design, two criteria are considered to optimize the target codes.

1) Smallest Hamming Weight of H: This criteria commonly indicates that the solution can be completed by using the lowest number of the logic gates in the synthesis process of the encoder and decoder.

2) Smallest Hamming Weight of the Heaviest Row of H: Logic depth in the encoding and decoding process depends on the logic path with the largest delay. The smallest hamming weight in the heaviest row can decrease the delay of the encoder and decoder.

With these restrictions discussed previously, based on the algorithm in Section IV, the solution for 3-bit BEC with QAEC codes can be found.

IV. Searching Algorithms and Tool Development

In this section, an algorithm is proposed to solve the Boolean satisfiability problem based on the discussion in the former section. Based on the algorithm, a code optimization tool is developed to obtain the target H matrix with custom optimization restrictions. The introduction of the algorithm is divided into two subsections. In Section IV-A, the basic part of the algorithm is introduced to find the solutions meeting the requirement of the Boolean satisfiability. In Section IV-B, based on the basic part, the method with column weight restriction is designed to force the optimization process to use as few ones as possible, thus optimizing the total number of ones in the matrix and the number of ones in the heaviest row. This optimized version of the algorithm has been used to obtain all the codes presented in this paper.

A. Basic Part of Code Design Algorithm

In order to construct the expected codes, the first step is to ensure the number of the check bits. From the aspect of low redundancy, the number of the check bits is set to the values shown in Table I. The number of the check bits is seven for codes with 16 data bits, eight for codes with 32 data bits, and nine for codes with 64 data bits.

The main idea of the algorithm is based on the recursive backtracing algorithm. At first, an identity matrix with block size $(n - k)$ is constructed as the initial matrix, and the corresponding syndromes of the error patterns are added to the syndrome set. Then, a column vector selected from the $2^{n-k} - 1$ column candidates is added to the right side of the initial matrix. This process is defined as column-added action. Meanwhile, the new syndromes which belong to the new added column are calculated. If none of new syndromes is equal to the elements in a syndrome set, the column-added action is successful and the corresponding new syndromes are added into the syndrome set. The base-matrix is updated to the previous matrix with the added column. Otherwise, the column-added action fails and another new column from the candidates is selected. If all the candidates are tried and the column-added action still fails, one column from the right side of previous matrix and the corresponding syndromes are reduced from the base-matrix and the syndrome set, respectively. Then, the algorithm continues the column-added action until the matrix dimension reaches the expected value. The code design algorithm flow is shown in Fig. 3.

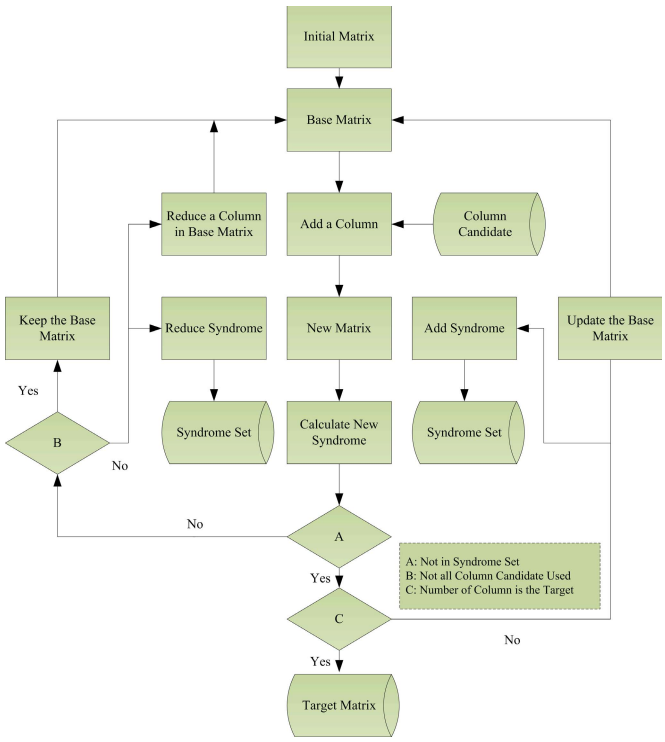

Fig. 3. Flow of code design algorithm.

Normally, the recursive backtracing algorithm demands a large amount of computing resources and computing time. In order to accelerate the computing speed of the algorithm operation, firstly, we adopt the decimal operation instead of the matrix operation by conversing the column vectors into decimal numbers. Even so, the algorithm completing the execution of all conditions is not possible. In general, if the code we expect exists, it is easy to obtain the first solution. With different optimization criteria, the algorithm can get better solutions. However, searching the best solution requires the complete result of the whole searching process, which is in most cases, unfeasible with today's computing resources. Therefore, it is more practical to use the best result obtained in a reasonable computation time. To be consistent with [23], that time was set to one week for all the results presented in this paper.

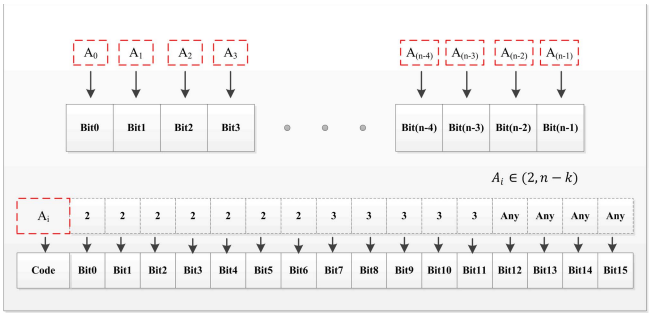

Fig. 4. Restriction of the column weight.

B. Optimization Part of Code Design Algorithm

For both optimization criteria mentioned before, the column used to construct the matrix should have as low weight as possible. In this case, the weight of the used column is restricted so as to minimize the weight of the matrix and speed up the process of finding better solutions. The method of column weight restriction for 3-bit BEC codes is shown in Fig. 4, where A_i is the available minimum weight of the column. Through the control of the weight of each column, the computing operation can quickly cover the whole cycle list by decreasing the amount of available candidates of the columns added to the base-matrix. Although this exhausted control can quickly narrow the searching scope, it is unwise to use it for all the bits as it can lead to missing better solutions due to harsh restriction. With the proposed column weight restriction method, the searching algorithm with more optimization criteria for the columns can find the solutions more effectively. The performance of the new finding process is remarkably effective for codes with small block size. When the number of data bits reaches 32 or 64, the computing time is still huge amount. Here, a function of recording the past procedure is also developed with the column weight restriction method to shorten the time cost of searching the target codes. The detailed algorithm flow is shown in Fig. 5.

At the initialization step, all the column weight restrictions are set to $(n-k)$. Then, A_0 is set to 2, which means that the number of 1 in the corresponding column is 2. The searching algorithm with column weight restriction starts to find the solution. If the solution exists, record and update the status of the bit restricted and shift to the next bit column weight setting. Otherwise, releases the column weight restriction to $A_i = A_i + 1$. With the program constraints increasing, the target code is close to appear.

To prove the performance of the algorithm proposed in this paper, we apply the algorithm to the finding process of 3-bit BEC in [23]. Compared with the presented codes in [23], codes with more advanced performance are found by using the mentioned algorithm. The performance comparison is shown in Table II. The value in brackets of "Smallest Total Ones" represents the number of total ones. The value in brackets of "Smallest Heaviest Row" represents the number of one in the heaviest row.

From Table II, it can be observed that the new algorithms are able to find better solutions than the existing algorithms

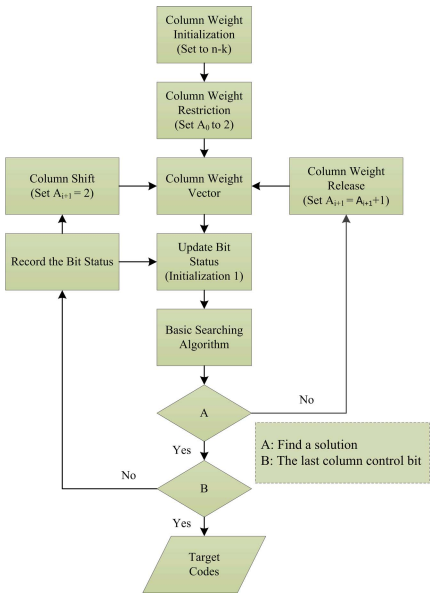

Fig. 5. Flow of algorithm with column weight restriction and past procedure record.

TABLE II
PERFORMANCE OF PROPOSED ALGORITHM FOR 3-bit BEC CODES

(n, k)	Smallest Total Ones		Smallest Heaviest Row	
	Codes [23]	Proposed Algorithm	Codes [23]	Proposed Algorithm
(23,16)	46(8)	45(7)	7(49)	7(45)
(40,32)	90(15)	88(13)	12(91)	12(89)
(73,64)	180(25)	180(21)	23(182)	21(180)

such as the one used in [23]. Therefore, they can be applied for the finding process of the QAEC codes. The solutions found for QAEC codes are presented Section V.

V. PROPOSED CODES

In terms of computing time, it is possible for QAEC codes with 16 data bits to have access to all the solutions, but it is impossible for QAEC codes with 32 and 64 data bits. Therefore, in this paper, the best solutions are presented for QAEC codes with 16 data bits and the best solutions found in a reasonable time (one week) using the proposed searching algorithm are presented for QAEC codes with 32 and 64 data bits. In terms of the two optimization criteria mentioned in Section III, for codes (23, 16), the two best parity check matrices are shown in Figs. 6 and 7 with both criteria best optimized. The parity check matrix for (40, 32) optimized to reduce the total number of ones is shown in Fig. 8 and the parity check matrices for (40, 32) optimized to reduce the maximum number of ones in a row is shown in Fig. 9. For codes (73, 64), the solutions obtained with both criteria better optimized are the same. The matrix is shown in Fig. 10.

VI. PROCEDURE OF ENCODING AND DECODING FOR QAEC CODES

In this section, we elaborate on the encoding and decoding procedure of the proposed 3-bit BEC-QAEC codes. The fundamental theory of encoding and decoding were discussed in

$$H=\begin{bmatrix}
1&0&0&0&0&0&0&1&0&0&1&0&1&0&1&1&1&0&0&0&1&0&0\\
0&1&0&0&0&0&0&0&1&0&1&1&0&1&1&0&0&1&0&0&0&1&0\\
0&0&1&0&0&0&0&0&0&1&0&0&0&1&0&0&1&1&0&1&0&0&1\\
0&0&0&1&0&0&0&1&0&0&1&0&0&0&1&0&1&0&1&1&0&0&0\\
0&0&0&0&1&0&0&0&1&0&0&1&1&1&1&0&0&0&0&1&1&0&0\\
0&0&0&0&0&1&0&0&1&0&0&0&1&1&0&0&0&1&0&1&0&0&1\\
0&0&0&0&0&0&1&0&0&1&0&1&0&0&0&1&1&1&1&0&0&1&1
\end{bmatrix}$$

Fig. 6. Best parity check matrix for the (23, 16) 3-bit burst with QAEC optimized to reduce the total number of ones and the maximum number of ones in a row.

$$H=\begin{bmatrix}
1&0&0&0&0&0&0&1&0&1&1&1&1&0&0&0&1&0&0&1&0&0&1\\
0&1&0&0&0&0&0&0&1&0&0&1&0&1&1&1&0&0&1&0&1&0&0\\
0&0&1&0&0&0&0&0&0&0&1&1&0&0&1&0&0&1&0&0&1&1&0\\
0&0&0&1&0&0&0&1&0&0&1&0&0&1&0&0&1&0&0&0&0&0&0\\
0&0&0&0&1&0&0&1&0&1&1&0&1&1&0&0&0&0&1&0&1&1&0\\
0&0&0&0&0&1&0&0&1&0&0&0&1&1&0&0&0&1&0&1&0&0&1\\
0&0&0&0&0&0&1&0&0&1&0&1&1&0&1&1&0&1&0&0&0&1&0
\end{bmatrix}$$

Fig. 7. Best parity check matrix for the (23, 16) 3-bit burst with QAEC optimized to reduce the total number of ones and the maximum number of ones in a row.

$$H=\begin{bmatrix}
1&0&0&0&0&0&0&0&1&0&0&1&0&0&1&0&0&0&0&1&0&0&0&1&0&0&0&0&1&0&0&0&1&0&0&0&0&1&0&0\\
0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&1&0&0&0&1&0&0&0&0&1&0&0&0&1&0&0&0&0&1&0\\
0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&1&0&0&0&1&1&0&0&0&1&0&0&0&0&1&0&0&0&1&0&0&0&0&1&1\\
0&0&0&1&0&0&0&0&0&0&0&1&0&1&0&0&0&1&0&0&0&1&1&0&0&0&1&0&0&0&1&0&0&0&1&0&1&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&1&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&1&0&1&0\\
0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&1&0&1&1\\
0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&1&1&0\\
0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&1&1\\
0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&1\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0
\end{bmatrix}$$

Fig. 8. Parity check matrix for the (40, 32) 3-bit burst with QAEC optimized to reduce the total number of ones.

$$H=\begin{bmatrix}
1&0&0&0&0&0&0&0&1&0&0&1&0&0&1&0&0&0&0&1&0&0&0&1&0&0&0&0&1&0&0&0&1&0&0&0&0&1&0&0\\
0&1&0&0&0&0&0&0&0&1&0&0&1&0&0&0&0&1&0&0&0&1&1&0&1&0&0&1&0&0&0&1&0&0&0&1&0&1&0&0\\
0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&1&0&0&0&1&0&0&0&1&1&0&0&0&1&1&0&0&0&1&0&1&1&1&0&1\\
0&0&0&1&0&0&0&0&0&0&0&1&0&1&0&0&0&1&0&0&0&1&0&0&0&0&0&1&0&0&0&1&0&0&0&1&0&1&0&1\\
0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&1&0&0&1&0&0&1&0&1&0&0&0&1&0&0&1&0&0&1&1&0&0&0&0&0\\
0&0&0&0&0&1&0&0&1&0&0&0&0&0&1&0&0&1&0&1&0&1&0&0&0&1&0&0&1&0&0&1&0&0&1&1&0&0&0&0\\
0&0&0&0&0&0&1&0&0&1&0&0&1&0&0&0&1&0&0&1&0&0&1&0&0&0&1&0&0&1&0&0&1&0&0&1&1&1&0&0\\
0&0&0&0&0&0&0&1&0&0&1&0&0&1&0&0&0&1&0&0&1&0&0&1&0&0&0&1&0&0&1&0&0&1&0&0&1&1&1&0\\
0&0&0&0&0&0&0&0&1&0&0&1&0&0&1&0&0&0&1&0&0&1&0&0&1&0&0&0&1&0&0&1&0&0&1&0&0&1&1&1
\end{bmatrix}$$

Fig. 9. Parity check matrix for the (40, 32) 3-bit burst with QAEC optimized to reduce the maximum number of ones in a row.

$$H=\begin{bmatrix}
1&0&0&0&0&0&0&0&0&1&0&0&1&0&0&1&0&0&0&0&1&0&0&0&1&0&0&1&1&1&0&1&0&1&0&0&0&0&0&1&1&0&0&1&0&1&0\\
0&1&0&0&0&0&0&0&0&0&1&0&0&1&0&0&0&0&0&1&0&0&0&1&0&0&1&0&0&1&1&0&1&0&0&1&0&1&0&0&0&0&0&1&0&1&0\\
0&0&1&0&0&0&0&0&0&0&0&1&0&0&1&0&0&0&0&1&0&0&0&1&0&0&1&0&0&1&1&0&0&1&0&1&1&1&1&0&0&0&0&1&0&1&1\\
0&0&0&1&0&0&0&0&0&0&0&0&1&0&1&0&0&0&0&1&0&0&0&1&0&0&1&0&0&1&1&0&1&0&0&1&1&1&0&1&0&0&0&1&0&1&1\\
0&0&0&0&1&0&0&0&0&0&0&0&0&1&0&0&1&0&0&1&0&0&0&1&0&0&1&0&0&1&1&0&0&1&0&1&1&1&0&1&0&0&0&1&0&1&1\\
0&0&0&0&0&1&0&0&1&0&0&0&0&0&1&0&0&1&0&1&0&0&0&1&0&0&1&0&0&1&1&0&0&1&0&1&1&1&0&1&0&0&0&1&0&1&1\\
0&0&0&0&0&0&1&0&0&1&0&0&1&0&0&0&1&0&0&1&0&0&0&1&0&0&1&0&0&1&1&0&0&1&0&1&1&1&0&1&0&0&0&1&0&1&1\\
0&0&0&0&0&0&0&1&0&0&1&0&0&1&0&0&0&1&0&0&1&0&0&1&0&0&1&0&0&1&1&0&0&1&0&1&1&1&0&1&0&0&0&1&0&1&1\\
0&0&0&0&0&0&0&0&1&0&0&1&0&0&1&0&0&0&1&0&0&1&0&0&1&0&0&0&1&0&0&1&0&0&1&0&0&1&1&1&0&1&0&0&0&1&0&1
\end{bmatrix}$$

Fig. 10. Parity check matrix for the (73, 64) 3-bit burst with QAEC optimized to reduce the total number of ones and the maximum number of ones in a row.

Section II. Here, an example for 16 data bits is illustrated in Fig. 11 with the H matrix used in Fig. 6. Based on the structure of the parity check matrix, the check bits are calculated by the corresponding data bits. The new encoded codeword, the combination of check bits and data bits is stored in the memory. When the particles hit the memory resulting in MBUs, the contents of affected memory cells are flipped. Here, to elaborate on the correction ability of QAEC codes, quadruple adjacent bits are flipped on D_2, D_3, D_4, and D_5. In the decoding process, the syndrome is calculated using the stored check bits and data bits and the structure of the parity check matrix. Through the corresponding relationship between the syndrome and the XOR result of the columns mentioned in Section II, the flipped bits can be located. With the flipped bits inverted, the errors from the storage stage in the memory are effectively corrected. This is the whole procedure of encoding and decoding for the proposed QAEC codes.

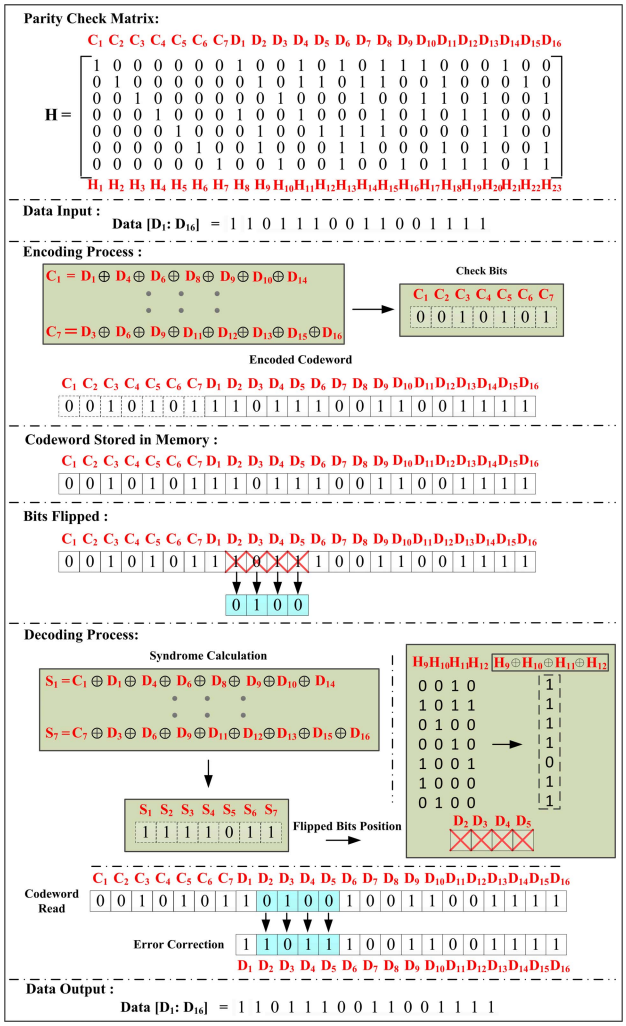

Fig. 11. Procedure of encoding and decoding for QAEC codes.

VII. Evaluation and Comparison

In this section, we first compare the complexity and implementation cost of the proposed codes against existing 3-bit BEC codes [23]. Then, in Section VII-B, the potential benefits in terms of reliability are discussed.

A. Complexity and Implementation Cost

The comparison is implemented from two aspects. One is the comparison of redundancy and complexity and the other one is the comparison of implementation overhead on area and delay. The results for each solution are summarized in Table III, where *TS* stands for the condition that total number of ones in H is the smallest, and *HRS* stands for the condition that the number of ones in the heaviest row of H is the smallest.

In the redundancy and complexity part, the proposed codes hold the same number of check bits with the previous codes in [23]. Although the new codes are optimized to have more advanced correction ability, the number of parity check bits is unchanged and therefore the area used for check bits in memories is still the same as before. However, the weight

TABLE III

OVERHEAD COMPARISON OF THE PROPOSED CODES TO 3-bit BEC CODES [23]. AREA EXPRESSED IN μm^2 AND DELAY IN NANOSECONDS

k	Code		Redundancy and Complexity			Area Optimization				Delay Optimization			
			(n-k)	Weight of *H* Matrix	Weight of Heaviest Row	Enc		Dec		Enc		Dec	
						Area	Delay	Area	Delay	Area	Delay	Area	Delay
16	Proposed QAEC	Fig. 6	7	54	9	103.99	0.68	635.20	3.23	279.99	0.40	3011.59	0.83
		Fig. 7	7	54	9	105.99	0.67	606.80	3.36	257.59	0.41	3055.99	0.89
	Codes [23]	*TS*	7	46	8	101.19	0.55	507.20	2.89	433.99	0.32	2350.79	0.80
		HRS	7	49	7	101.19	0.63	540.00	3.32	324.79	0.35	2723.99	0.81
32	Proposed QAEC	Fig. 8	8	96	14	232.39	0.99	1173.20	3.69	550.79	0.48	5321.99	0.96
		Fig. 9	8	97	13	217.19	0.83	1252.00	3.99	439.99	0.46	5799.59	0.99
	Codes [23]	*TS*	8	90	15	208.39	0.77	993.60	3.81	740.79	0.42	4943.19	0.87
		HRS	8	91	12	210.39	0.74	935.20	3.28	743.59	0.39	4269.59	0.90
64	Proposed QAEC	Fig. 10	9	186	22	536.39	1.27	2506.80	5.80	1261.59	0.59	12218.79	1.02
	Codes [23]	*TS*	9	180	25	435.59	0.92	1978.00	4.05	1191.59	0.53	9140.39	0.96
		HRS	9	182	23	440.39	0.94	2004.40	4.09	1440.39	0.50	8855.99	1.03

of *H* matrix and the weight of heaviest row increase with the improvement of correction ability. These make the encoding and decoding process more complex. But in terms of correction ability improved, this complexity increase is reasonable. The details of this complexity increase will be evaluated in the following comparison between area and delay.

The encoders and decoders have been modeled in Verilog and synthesized for a TSMC bulk 65-nm library. To obtain a better comparison, the synthesis is done twice for each code with different synthesis constraints optimizing area and delay, respectively. The results are presented in Table III. As for each word length, there are two codes (total number of ones and heaviest row), to compare the encoder and decoder complexity of the proposed codes with those presented in [23], for each parameter code with the best result is used. Focusing on 16-bit data words, for area optimization, the proposed codes require an area overhead of 2.8% for the encoder and of 19.6% for the decoder. The area overheads for 32- and 64-bit data words are 4.3% and 23.2% for the encoder and 25.6% and 26.7% for the decoder. This suggests that the overheads tend to increase with the word size. It is worth noticing that the area of the encoder and decoder will be in many cases small compared with that of the memory. Therefore, the area overheads will be much lower when looking at the entire memory design. Turning now to the delay, the oveheads for the encoder are 25%, 18%, and 18% for 16, 32, and 64 bit data words. For the decoder, the overheads are lower: 3.8%, 10%, and 6.3% again for 16, 32, and 64 bit data words. This is interesting as the decoder delay is typically larger than that of the encoder and is the limiting factor for speed in an ECC.

As a summary, the proposed codes with advanced correction ability keep the redundancy unchanged, but have an increase in encoding and decoding complexity. For an implementation in the HDL and synthesized for a 65-nm library, the increase of area and delay is moderate (less than 27% for area and 10% or less for decoding delay) and tolerable in exchange for a significant improvement of correction ability.

B. Reliability

The benefit that the QAEC feature provides in terms of reliability depends largely on the error patterns that occur in the memory and their frequency. Using that data, we can determine the percentage of errors that can be corrected by the ECC and those that will not be corrected. The error patterns and frequencies in turn depend on several factors like the technology node, the memory design, and the radiation particles to which the memory is exposed [31], [32]. Therefore, the reliability improvement will vary from case to case. To provide some insight into the benefits of implementing QAEC, we can discuss the percentage of four bit burst error patterns that will be corrected. There are four possible error patterns for a four bit burst error: 1001, 1011, 1101, and 1111, where "1" means that an error has occurred in that cell and 0 that there is no error as before. The QAEC corrects only the last one and thus the percentage of four bit burst errors that are corrected will be large when the 1111 pattern is the most frequent one. In that case, the proposed codes could remove most of the four bit burst errors. On the other hand, if the other patterns are much more frequent, the benefit of using QAEC will be low. In this paper, we have assumed that the quadruple adjacent error was the most common for the four bit burst error patterns as reported for some memories [31], [33]. In particular, in [33], the four adjacent bit error pattern is seventh most common MCU pattern observed in the configuration memory of a modern Xilinx FPGA and the only four bit burst pattern that is among the ten most frequent ones. In those cases, the QAEC will provide significant benefits. For memories or environments on which another of the four patterns is dominant, the proposed code design algorithm can be used to provide codes that are 3-bit BEC and that can also correct that four bit burst error-dominant pattern. Therefore, the benefit will again be significant. Finally, if all the four bit burst error patterns have a similar frequency, then the proposed codes can eliminate one fourth of the four bit burst error patterns. This will be the worst case for the proposed codes and it can be observed that even in this scenario, the reduction in the four bit burst errors is still significant. As a summary, the benefit that the proposed codes can provide for a given memory in a given environment needs to be assessed based on the error patterns and frequencies that are obtained as part of radiation testing. The proposed codes will be useful when there is a relevant number of four bit burst error patterns and especially if they are concentrated in one of the four possible patterns.

VIII. Conclusion

In this paper, a technique to extend the 3-bit BEC codes with the QAEC is presented. The proposed codes have the same redundancy as the previous 3-bit BEC codes [23]. To accelerate the searching process of target matrices, a new algorithm with column weight control and recording function is proposed. Based on the proposed algorithm, a searching tool is developed to execute the searching process automatically. To prove the validity of the proposed algorithm, it is applied to the previous 3-bit BEC codes [23] and the codes are remarkably improved on the two optimization criteria. Then, the proposed algorithm is used to find the solution for the QAEC. The complete solution searching process is finished for 16 data bits and the searching process using optimization algorithm is carried out for 32 and 64 data bits. Therefore, in this paper, the best solutions are presented for 16 data bits and the best solutions found in a reasonable computation time are presented for 32 and 64 data bits. The encoder and decoder of the proposed codes are implemented by using the HDL and synthesized for a 65-nm library. The overhead of area and delay is moderate versus previous 3-bit BEC codes [23]. This suggests that the proposed 3-bit BEC with QAEC codes can be effectively used by designers to protect the SRAM memories from radiation effect and mitigate the MBUs that affect up to four adjacent bits. Finally, as noted before, the proposed scheme could be extended to design 3-bit burst ECCs that can correct another of the 4-bit burst error patterns instead of the quadruple adjacent error. This can be of interest for applications in which there is a dominant 4-bit burst error pattern that is not the quadruple adjacent error.

Acknowledgment

AMD, the AMD Arrow logo, and combinations thereof are trademarks of Advanced Micro Devices, Inc. Other product names used in this publication are for identification purposes only and may be trademarks of their respective companies.

References

[1] R. D. Schrimpf and D. M. Fleetwood, *Radiation Effects and Soft Errors in Integrated Circuits and Electronic Devices*. Singapore: World Scientific, 2004.

[2] S. K. Vishvakarma, B. S. Reniwal, V. Sharma, C. B. Khuswah, and D. Dwivedi, "Nanoscale memory design for efficient computation: Trends, challenges and opportunity," in *Proc. IEEE Int. Syst. Nanoelectron. Inf. Syst.*, Dec. 2015, pp. 29–34.

[3] S.-Y. Wu, C. Y. Lin, S. H. Yang, J. J. Liaw, and J. Y. Cheng, "Advancing foundry technology with scaling and innovations," in *Proc. Int. Symp. VLSI-TSA*, Apr. 2014, pp. 1–3.

[4] ITRS. *International Technology Roadmap for Semiconductors (ITRS) Report, Semiconductor Industry Association*. Accessed: 2009. [Online]. Available: http://www.itrs.net

[5] Y. Bentoutou, "A real time EDAC system for applications onboard earth observation small satellites," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 48, no. 1, pp. 648–657, Jan. 2012.

[6] E. Ibe, H. Taniguchi, Y. Yahagi, K.-I. Shimbo, and T. Toba, "Impact of scaling on neutron-induced soft error in SRAMs from a 250 nm to a 22 nm design rule," *IEEE Trans. Electron Devices*, vol. 57, no. 7, pp. 1527–1538, Jul. 2010.

[7] L. Artola, M. Gaillardin, G. Hubert, M. Raine, and P. Paillet, "Modeling single event transients in advanced devices and ICs," *IEEE Trans. Nucl. Sci.*, vol. 62, no. 4, pp. 1528–1539, Aug. 2015.

[8] M. Murat, A. Akkerman, and J. Barak, "Electron and ion tracks in silicon: Spatial and temporal evolution," *IEEE Trans. Nucl. Sci.*, vol. 55, no. 6, pp. 3046–3054, Dec. 2008.

[9] R. C. Baumann, "Soft errors in advanced computer systems," *IEEE Design Test. Comput.*, vol. 22, no. 3, pp. 258–266, May/Jun. 2005.

[10] 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.

[11] J. D. Black, P. E. Dodd, and K. M. Warren, "Physics of multiple-node charge collection and impacts on single-event characterization and soft error rate prediction," *IEEE Trans. Nucl. Sci.*, vol. 60, no. 3, pp. 1836–1851, Jun. 2013.

[12] O. A. Amusan *et al.*, "Charge collection and charge sharing in a 130 nm CMOS technology," *IEEE Trans. Nucl. Sci.*, vol. 53, no. 6, pp. 3253–3258, Dec. 2006.

[13] J. D. Black *et al.*, "Characterizing SRAM single event upset in terms of single and multiple node charge collection," *IEEE Trans. Nucl. Sci.*, vol. 55, no. 6, pp. 2943–2947, Dec. 2008.

[14] A. D. Tipton *et al.*, "Multiple-bit upset in 130 nm CMOS technology," *IEEE Trans. Nucl. Sci.*, vol. 53, no. 6, pp. 3259–3264, Dec. 2006.

[15] W. Calienes, R. Reis, C. Anghel, and A. Vladimirescu, "Bulk and FDSOI SRAM resiliency to radiation effects," in *Proc. IEEE 57th Int. Midwest Symp Circuits Syst.*, Aug. 2014, pp. 655–658.

[16] D. F. Heidel *et al.*, "Single-event upsets and multiple-bit upsets on a 45 nm SOI SRAM," *IEEE Trans. Nucl. Sci.*, vol. 56, no. 6, pp. 3499–3504, Dec. 2009.

[17] E. H. Cannon, D. D. Reinhardt, M. S. Gordon, and P. S. Makowensky, "SRAM SER in 90, 130 and 180 nm bulk and SOI technologies," in *Proc. 42nd Annu. IEEE Int. Reliab. Phys. Symp.*, Apr. 2004, pp. 300–304.

[18] M. Raine, G. Hubert, M. Gaillardin, P. Paillet, and A. Bournel, "Monte Carlo prediction of heavy ion induced MBU sensitivity for SOI SRAMs using radial ionization profile," *IEEE Trans. Nucl. Sci.*, vol. 58, no. 6, pp. 2607–2613, Dec. 2011.

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

[20] 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.

[21] Z. Ming, X. L. Yi, and L. H. Wei, "New SEC-DED-DAEC codes for multiple bit upsets mitigation in memory," in *Proc. IEEE/IFIP 20th Int. Conf. VLSI Syst.-Chip*, Oct. 2011, pp. 254–259.

[22] 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.

[23] L. J. Saiz-Adalid, P. Reviriego, P. Gil, S. Pontarelli, and J. A. Maestro, "MCU tolerance in SRAMs through low-redundancy triple adjacent error correction," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 23, no. 10, pp. 2332–2336, Oct. 2015.

[24] L. Xiao, J. Li, J. Li, and J. Guo, "Hardened design based on advanced orthogonal Latin code against two adjacent multiple bit upsets (MBUs) in memories," in *Proc. 16th Int. Symp. Quality Electron. Design*, Mar. 2015, pp. 485–489.

[25] R. Naseer and J. Draper, "Parallel double error correcting code design to mitigate multi-bit upsets in SRAMs," in *Proc. IEEE 34th Eur. Solid-State Circuits*, Sep. 2008, pp. 222–225.

[26] M. A. Bajura *et al.*, "Models and algorithmic limits for an ECC-based approach to hardening sub-100-nm SRAMs," *IEEE Trans. Nucl. Sci.*, vol. 54, no. 4, pp. 935–945, Aug. 2007.

[27] 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.

[28] S. Satoh, Y. Tosaka, and S. A. Wender, "Geometric effect of multiple-bit soft errors induced by cosmic ray neutrons on DRAM's," *IEEE Electron Device Lett.*, vol. 21, no. 6, pp. 310–312, Jun. 2000.

[29] A. I. Chumakov, A. V. Sogoyan, A. B. Boruzdina, A. A. Smolin, and A. A. Pechenkin, "Multiple cell upset mechanisms in SRAMs," in *Proc. 15th Eur. Conf. Radiation Effects Compon. Syst.*, Sep. 2015, pp. 1–5.

[30] 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.

[31] G. Tsiligiannis *et al.*, "Multiple cell upset classification in commercial SRAMs," *IEEE Trans. Nucl. Sci.*, vol. 61, no. 4, pp. 1747–1754, Aug. 2014.

[32] J.-L. Autran, D. Munteanu, G. Gasiot, P. Roche, S. Serre, and S. Semikh, *Soft-Error Rate of Advanced SRAM Memories: Modeling and Monte Carlo Simulation*. Rijeka, Croatia: INTECH, 2012.

[33] M. Wirthlin, D. Lee, G. Swift, and H. Quinn, "A method and case study on identifying physically adjacent multiple-cell upsets using 28-nm, interleaved and SECDED-protected arrays," *IEEE Trans. Nucl. Sci.*, vol. 61, no. 6, pp. 3080–3087, Dec. 2014.

Jiaqiang Li (S'17) received the B.S. and M.S. degrees in microelectronics and solid-state electronics from Harbin Institute of Technology, Harbin, China, in 2011 and 2014, respectively, where he is currently working toward the Ph.D. degree in microelectronics and solid-state electronics.

From 2014 to 2016, he was a member of the Alpha Magnetic Spectrometer (AMS)-02 experiment in European Organization for Nuclear Research, Geneva, Switzerland, where he was involved in the analysis of cosmic rays and the reliability analysis of AMS electronic devices on the international space station. His current research interests include fault tolerance in VLSI designs, reliability in memories, and radiation effects in electronic devices.

Pedro Reviriego (SM'15) received the M.Sc. and Ph.D. degrees in telecommunications engineering from the Technical University of Madrid, Madrid, Spain, in 1994 and 1997, respectively.

From 1997 to 2000, he was a Research and Development Engineer with Teldat, Madrid, where he was involved in router implementation. In 2000, he joined Massana, Madrid, where he was involved in the development of 1000BaseT transceivers. In 2003, he was a Visiting Professor at the University Carlos III, Madrid. From 2004 to 2007, he was a Distinguished Member of Technical Staff with the LSI Corporation, Madrid, where he was involved in the development of Ethernet transceivers. He is currently with the ARIES Research Center, Universidad Antonio de Nebrija, Madrid. He has authored numerous papers in international conference proceedings and journals. He has also participated in the IEEE 802.3 standardization for 10G-BASE-T. His current research interests include fault-tolerant systems and communication networks.

Liyi Xiao (M'07) received the B.S., M.S., and Ph.D. degrees in microelectronics and solid-state electronics from Harbin Institute of Technology, Harbin, China, in 1984, 1989 and 2001, respectively.

She has been a Professor at the Department of Electronic Science and Technology, Harbin Institute of Technology, since 2004. Her current research interests include reliability in semiconductor devices and extra-low power design in VLSI, ASIC/SoC.

Costas Argyrides (S'07–M'10–SM'16) received the B.Sc. degree (honors) in informatics and computer science from the Moscow Power Engineering Institute, Technical University, Moscow, Russia, in 2004 and the M.Sc. degree in advanced computing and the Ph.D. degree in computer science from the University of Bristol, Bristol, U.K., in 2005 and 2008, respectively.

He was a Validation Engineer at Intel Corporation, Santa Clara, CA, USA. Before joining the industry, he was a Postdoctoral Researcher at the Polytechnic University of Catalonia, Barcelona, Spain; Newcastle University, Newcastle upon Tyne, U.K.; Oxford Brookes University, Oxford, U.K.; University of Bristol, University of Warwick, Coventry, U.K.; and the University of Cambridge, Cambridge, U.K. He has authored or coauthored over 50 technical papers. He joined Advanced Micro Devices, Inc. (AMD), Sunnyvale, CA, USA, in 2014, where he is currently an RAS Architect with the Radeon Technologies Group, AMD. His current research interests include fault-tolerant computer systems, software fault tolerance, reliability improvement, error correcting codes, algorithmic based fault tolerance, and nanotechnology-based designs.

Jie Li received the B.S. and M.S. degrees in microelectronics and solid-state electronics from the Harbin Institute of Technology, Harbin, China, in 2013 and 2015, respectively, where he is currently pursuing the Ph.D. degree in microelectronics and solid-state electronics.

From 2015 to 2016, he participated in the projects of IP cores evaluation and the pulsed laser simulation technique for transient radiation effects and published several papers in international conference proceedings. His current research interests include fault tolerance in SoC designs and mechanism research on radiation effects in ICs.