SEC–DED–DAEC Codes for On-chip Memory Protection against Multiple Cell Upsets

Hoyoon Jun
School of Electrical and Electronic Engineering
Yonsei University
Seoul, Korea
Email: hockma24 {at} yonsei.ac.kr

Yongsurk Lee
School of Electrical and Electronic Engineering
Yonsei University
Seoul, Korea

Abstract—**As integrated circuit technology scales into the deep-submicron regime, radiation-induced soft errors threat the reliability of on-chip memory applications. Neutron-induced soft errors cause Multiple Cell Upsets (MCUs) in physically adjacent regions and may lead to system failures. Recently Single Error Correction, Double Error Detection, and Double Adjacent Error Correction (SEC–DED–DAEC) codes are presented to correct these errors. However, these conventional codes do not resolve mis-correction of double non-adjacent errors because the syndromes for double non-adjacent errors are equal to those for double adjacent errors. In this paper, we propose SEC–DED–DAEC codes that have no mis-correction. To obviate the mis-correction, the column vectors in a H-matrix are selected from a column pool matrices with reversed colexicographic order and are alternately placed according to column weight. Experimental results show that the mis-correction rate for our proposed codes is zero and the hardware overhead of the decoder and parity check bits is small. The proposed codes are suitable for protecting on-chip memory applications from MCUs.**

Keywords-error correcting code, neutron-induced soft error, multiple cell upsets, on-chip memory

I. INTRODUCTION

Radiation-induced soft errors are a major concern in on-chip memory applications as the size and the supply voltages of silicon devices scale down to achieve high density and low power integrated circuits respectively. Although these errors are transient, when they occur in critical resigns of the operating system, they may cause system malfunctions [1,2]. To address these errors, Error Correcting Codes (ECC) schemes have emerged and Single Error Correction and Double Error Detection (SEC–DED) codes are widely employed in commercial on-chip memory applications [3,4]. The SEC–DED codes can correct single-bit errors and detect double-bits errors using redundant bits called parity check bits. The advantage of SEC–DED is fast decoding with a small number of parity check bits. However, neutron-induced soft errors lead to Multiple Cell Upsets (MCUs) in adjacent regions because the tracks of heavy ions due to nuclear reactions result in the information in storage nodes being changed.

Bose-Chaudhuri-Hocquenghem (BCH) codes [5], Orthogonal Latin Square (OLS) codes [6], and bit interleaving [7] are presented to address MCUs in on-chip memory applications. However, even though BCH codes correct both double adjacent and double non-adjacent errors, the implementation overhead in terms of parity check bits and design complexity for the decoder is large. OLS codes can correct double-bit errors with low logic complexity; however, they require an even greater number of parity check bits than that required by BCH codes. SEC–DEC codes with bit interleaving efficiently converts multiple bit errors to a single-bit error. However, the large-way bit interleaving not only incurs problems with the aspect ratio in the floor planning, performance degradation, area overhead and power consumption for on-chip memories, but it cannot be applied to Content Addressable Memories (CAM) or register files [8].

Single Error Correction, Double Error Detection, and Double Adjacent Error Correction (SEC–DED–DAEC) codes that correct a single error and double adjacent errors are recently proposed [9–13]. Further, the number of parity check bits used for these codes and the XOR gates that are used to generate syndrome in the decoder is lower than that of Double Error Correction (DEC) BCH codes. In addition, the corrector logic for the SEC–DED–DAEC codes decoder is simpler than that of DEC BCH codes, because double non-adjacent error correction capability is not required. However, conventional DAEC codes do not address mis-correction of double non-adjacent errors because syndromes for double non-adjacent errors are equal to those of double adjacent errors. To achieve high reliability in on-chip memory applications, this possible threat must be eliminated.

In this paper, we propose SEC–DED–DAEC code that corrects double adjacent errors with no mis-correction to achieve high reliability against MCUs in on-chip memory applications. The proposed code is shortened hamming codes comprising equivalent numbers of even and odd weight columns to obviate mis-correction problems. Experimental results indicate that the proposed code provides DAEC capability with no mis-correction and that the implementation overhead is lower than that of conventional SEC–DED–DAEC and DEC BCH codes.

In the rest of this paper, we briefly discuss conventional SEC–DED–DAEC codes in Section II, presents an overview of ECC basic in Section III, describe our proposed SEC–DED–DAEC code in Section IV, show the experimental results and comparisons with conventional codes in Section V, and conclude this paper in Section VI.

II. Conventional SEC–DED–DAEC Codes

There are several SEC–DED–DAEC codes based on binary linear block codes. They are shortened hamming codes that provide error correction and detection capabilities that are superior to that of Hamming SEC [4]. To provide DAEC, the syndrome for double adjacent errors must be separated from those for single errors and other double adjacent errors. However, the hazard due to mis-correction are not eliminated unless the syndromes for double adjacent errors and double non-adjacent errors is mutually separated.

Dutta code [9] provide DAEC by relocating columns in the H-matrix derived from that of the Hsiao SEC–DED code [6], however, their mis-correction rate is very high. Although even and odd weight columns in the H-matrix are utilized in the Richter [10], Datta [11], and Ming [12] codes in an attempt to reduce mis-correction rates, the mis-correction still occurs in these codes. Further, even though Datta codes have the lowest mis-correction rate, they require a large number of parity check bits and XOR gates to generate syndromes. Repeated identity matrices of Fire code [14] is employed in the H-matrix for Neale code to detect consecutive errors and correct double adjacent errors [13]. Although the number of XOR gates to syndrome in Neale codes is lower than that in other codes, the mis-correction rate is marginally higher than that of Datta codes.

III. Error Correcting Codes Basic

A. Binary linear block codes

A (n, k) binary linear block code is a subspace of Galois Field GF(2)^n, if a block code of length n and 2^k codewords form a k-dimensional subspace of the vector space of all the n-tuple over the GF(2). In 2^n binary codes, there are only 2^k distinct information. The set of 2^k codewords is refer to a block code. Thus, k-tuple information **u** should be one-to-one corresponded its n-tuple codeword **v**. In addition, the codeword is closed under modulo-2 addition. In fact, a modulo-2 addition of two binary block codes is also a codeword.

A linear block code possess linearity. A linear combination $\boldsymbol{lc}$ of k vectors $v_0, \ldots, v_{k-1}$ can be defined as a sum of the follow form:

$$\boldsymbol{lc} = a_0 v_0 + a_1 v_1 + \cdots + a_{k-1} v_{k-1} \quad (1)$$

The a_i is scalar in the GF(2). If the $\boldsymbol{lc}$ is not zero, a set of vectors $v_0, \ldots, v_{k-1}$ is linearly independent. In any vector space, the number of linearly independent vectors is refer to the dimension of the space. Therefore, a (n, k) code contains the k-dimensional subspace of n-tuple vectors. If the inner product of two vector v_1 and v_2 is zero, they are orthogonal. These orthogonality is used to detect error. In addition, if a vector is orthogonal to every vector of a vector space $\mathbf{V}_1$, it belongs to the null space of $\mathbf{V}_1$. If $\mathbf{V}_1$ is (n, k) code, the dimension of null space is determined as $n - k$.

A correction and detection capability of a binary linear block depends on minimum Hamming distance. In two vectors, the hamming distance is defined as the number of positions in which they differ. For example, if there are two vectors $\mathbf{v}_1$ (01001) and $\mathbf{v}_2$ (00101), the hamming distance between two vectors is 2. The hamming distance is easily calculated using XOR operation. After these operation, the number of non-zero of the XOR result represents the hamming distance. A capability t of random error correction of a (n, k) code can be define as follows:

$$t = \frac{\lfloor d_{min} - 1 \rfloor}{2} \quad (2)$$

If a (n, k) code can simultaneously correct and detect errors, the capability t of correction and detection is defined as follows:

$$t = \frac{\lfloor d_{min} - 2 \rfloor}{2}, \quad for\ correction \quad (3)$$

$$t = \frac{d}{2}, \quad for\ detection \quad (4)$$

Therefore, a hamming distance of a SEC–DED codes is four.

A (n, k) binary linear block code is generated by $(n-k) \times n$ parity check matrix called a H-matrix. The H-matrix consists of n column vector with $(n-k)$-tuple. The H-matrix example for (7, 4) code with SEC is shown in the (5).

$$H = [h_0, h_1, \ldots, h_{n-1}] = \begin{bmatrix} 1011100 \\ 1110010 \\ 0111001 \end{bmatrix} \quad (5)$$

The $h_0 \sim h_{k-1}$ column vectors represent a parity equation matrix while $h_k \sim h_{n-1}$ do an identity matrix. To obtain a $n-k$ parity check bits, an AND and XOR operations are carried out between an information vector with k-tuple and k out of each row vector of the H-matrix. Therefore, the $h_0 \sim h_{k-1}$ columns are effective. For example, if the information $\boldsymbol{v}$ = (1, 0, 1, 1), its parity check bits are (1, 0, 0). As a result, codeword $\boldsymbol{c}$ of $\boldsymbol{v}$ is (1, 0, 1, 1, 1, 0, 0).

A decoding for error correction and detection of a binary linear block code is achieved by calculating a $(n-k)$-tuple syndrome. A non-zero vector syndrome represents error position in a codeword, whereas zero vector one does no-error. A calculation of syndrome is similar to generating the parity check bits without the range of effective columns ($h_0 \sim h_{n-1}$). In fact, if a syndrome is equal to a column vector, the position of the column vector corresponds to the error position. For example, if the $\boldsymbol{v}'$ is (0, 0, 1, 1, 1, 0, 0), its syndrome is (1, 1, 0) which is equal to column vector h_0. Therefore we know that the vector $\boldsymbol{v}'$ contains a single error and its position is the first.

B. SEC–DED codes

The H-matrix of SEC–DED codes guarantee a minimum hamming distance of four. The hamming distance is depended on a linear combination of column vectors in the H-matrix. Indeed, for distance *w*, if every combination of $w-1$ or fewer columns in H-matrix is linearly independent, the H-matrix has a minimum hamming distance of *w*. To achieve a SEC–DED, the H-matrix should be complied with the following rules:

1) Every column is not zero vector.
2) Every column is distinct.
3) An XOR result of any two columns is not equal to any column.

The first two rules give a hamming distance of three for a SEC. The third rule guarantees a hamming distance of four a SEC–DED. The distinct column implies that any column is not overlapped. In Hsiao code which is broadly adopted in memory applications, its H-matrix achieves a SEC–DED using following three rules:

1) There is no all-zero column vector.
2) Every column is distinct.
3) Every column contains an odd number of 1's (hence odd weight)

Because an XOR result vector of any odd weight column vector is even weight, the XOR result vector is not equal to any single column vector.

IV. Proposed SEC–DED–DAEC Code

A. H-matrix Generation Rules

To design the H-matrix for our proposed code with no mis-correction, the SEC and DAEC syndromes must be unique whereas the syndromes for double non-adjacent error detection can be allowed to overlap. To meet these constraints, the H-matrix of our proposed code satisfies the following rules:

1) There is no all-zero column vector.
2) Every column vector is distinct.
3) Every XOR result for double adjacent column vector is distinct.
4) Odd or even weight column vectors are not placed sequentially except for columns in identity matrix.
5) All even weight column vectors have weight greater than two.

The first two rules provide a hamming distance of three for SEC. The distinct column and XOR result of double adjacent column vectors imply that they do not overlap another single column vectors and XOR result for any double column vectors in the H-matrix. The second and third rules provide elimination of mis-correction. The fourth rule is helpful to separate syndromes for double adjacent errors from those of double non-adjacent errors. The fifth rule provides segregation of

$$\begin{pmatrix}
1111111111111110000000000000000000\\
1111100000000001111111110000000000\\
1000011110000001111000001111110000\\
0100010001110001000111000111000111\\
0010001001001100100100110100110110\\
0001000100101010010010101010110 11\\
0000100010010110001001011001011 0111
\end{pmatrix}$$

Figure 1. Column pool matrix with reversed colexicographic order

double errors in information bits and in parity check bits. Although syndromes for double non-adjacent errors between information bits and parity check bits can be overlapped with those for double adjacent errors, this condition is acceptable. In general, interleaving using column Mux is used to achieve effective regularity of the Static Random Access Memory (SRAM) layout because the bit-cell pitch in the horizontal direction of the SRAM columns is typically smaller than that of an I/O circuit, including the sense amplifier and the write driver. As a result, most information bits are physically stored away from the parity check bits.

Input k : the number of information bits
r : the number of parity check bits
Output H-matrix for SEC–DED–DAEC

```
1:  while Hcol < k do
2:    create column_pools, tH−matrix
3:    for i ← 3 to i ≤ r do
4:      select seed_column from column_pools
5:      Hcol ← 1
6:      while Hcol < k do
7:        overlap_weight ← 1
8:        for j ← 3 to j ≤ r do
9:          select column from column_pools
10:         if distinct_test() is true then
11:           Hcol + +
12:           update tH − matrix
13:           go to line 6
14:         end if
15:       end for
16:       overlap_weight + +
17:       if overlap_weight > r then
18:         exit line 6 while loop
19:       end if
20:     end while
21:     if Hcol ≡ k && min_one() is true then
22:       H−matrix ← tH−matrix
23:     end if
24:   end for
25:   if Hcol < k then
26:     r + +
27:   end if
28: end while
```

Figure 2. The algorithm for generating the H-matrix of the proposed code.

B. H-matrix Construction Procedure

The H-matrix for our proposed codes is constructed by a heuristic method. A column pool matrix with reversed colexicographic order as shown in Fig.1 is utilized to obviate mis-correction. This matrix is useful for separating double adjacent errors from double non-adjacent errors because successive ones and zeroes in the same row help to distinguish double adjacent errors from double non-adjacent errors occurring on both sides. The diagonal ones in the same sections where successive ones occur also help to differentiate double errors in those sections. In addition, there is an overlap weight representing the number of overlapped ones between double adjacent columns. The overlap weight determines the appropriate column from column pools to implement low power encoder and decoder. The default value of the overlap weight is one because a large overlap weight requires large weight column. In Fig. 2, we propose the algorithm to construct an H-matrix satisfying all of the proposed rules.

First, column pools and a temporary H-matrix are created according to the SEC–DED parity check bits. Second, column selection from the column pools is determined according to overlap weight, i.e., the weight of the XOR result from the last confirmed column and a new column selected. In addition, the column pools are retrieved from left to right taking advantage of the reverse colexicographic order. Third, the temporary H-matrix including a new column is tested on distinction for SEC and DAEC. If an appropriate H-matrix is not found, this process is repeated with augmented values for overlap weight, seed column, and the length of the parity check bits. Finally, the H-matrix with the smallest number of ones is selected to reduce the decoding logic area and power consumption.

V. Experimental Results and Comparisons

To verify the effectiveness of our proposed SEC–DED–DAEC codes, we simulated it in a high-level language. Table I shows the major parameters for the proposed codes. The *k*, *r* and, *n* are length of information, parity check and codeword bits. The fourth column in Table I represents the overhead in terms of parity check bits compared to those of the SEC–DED and the DEC BCH codes respectively. The last column in Table I represents the number of ones in the H-matrix. The number is equal to XOR gates used in generating syndrome in the decoder. A large number of XOR gates cause large power consumption and low decoding speed due to high logic complexity.

$$\begin{pmatrix}
1111111111111111100000000000010100000000\\
0111111100000000010101010000011010000000\\
0101010001010000011101000101010010000000\\
0100000011010101010011100000010000010000\\
0010000010101000000111110011101100001000\\
1001000100100100000100111101010000000100\\
0000100001000111001000001011001100000010\\
0000010100001010110010010110100100000001\\
1000001000010001101000001000111100000001
\end{pmatrix}$$

Figure 3. H-matrix for (41, 32) SEC–DED–DAEC code

TABLE I. Proposed SEC–DED–DAEC Code Parameters

n	*k*	*r*	*r*-OH	# of 1s
41	32	9	+2/-3	116
75	64	11	+3/-3	236
141	128	13	+4/-3	502

TABLE II. Comparison of SEC–DED–DAEC Code(*K*=32)

Codes	*r*	Mis-correction rate	# of 1s
Dutta	7	53.4%	96
Richter	7	39.0%	115
Datta	10	8.8%	140
Ming	7	34.3%	128
Neale	10	9.0%	80
DEC BCH	12	0%	200
Proposed	9	0%	116

Table II shows that our proposed codes and the DEC BCH codes do not occur mis-correction. Although the number of ones in the Neale codes is the lowest, mis-correction occurs in those codes. Further, the number of parity check bits required by our proposed codes is smaller than that for Datta and Neale codes, which both have mis-correction rates under 10%. The H-matrix of our proposed (41, 32) codes is shown in Fig. 3. Most of the columns are located such that they comply with the reversed colexicographic order to address mis-correction.

VI. Conclusion

In this paper, we proposed single error correction, double error detection, and double adjacent error correction with no mis-correction codes that improve soft error reliability in on-chip memory systems. The powerful DAEC is achieved by an H-matrix comprising columns that are selected from column pool matrices with reversed colexicographic order and are evenly located according to column weight. Our experimental results show that our propose codes have SEC–DED–DAEC capability without mis-correction and incur small overhead. The proposed codes increase interleaving distance in SRAMs without concern for mis-correction. In addition, they provide pseudo interleaving for CAM or register files that cannot employ interleaving structure. Thus, our proposed codes are suitable for use in protection schemes against MCUs in on-chip memory systems.

References

[1] U. Taiki, K. Takashi, M. Hideya, and M. Hashimoto, “Soft-Error in SRAM at Ultra-Low Voltage and Impact of Secondary Proton in Terrestrial Environment,” IEEE Trans. Nucl. Sci., vol. 60, no. 6, pp. 4232–4237, Dec. 2013.

[2] R. C. Baumann, "Radiation-induced soft errors in advanced semiconductor technologies," IEEE Trans. Device Mater. Rel., vol. 5, no. 3, pp. 305–316, Sep. 2005.

[3] 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.

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

[5] R. Naseer and J. Draper, "Parallel double error correcting code design to mitigate multi-bit upsets in SRAMs," in Proc. 34th European Solid-State Circuits Conf., Edinburgh, SC, UK, Sep. 2008, pp. 222–225.

[6] 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, 1970.

[7] 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.

[8] S. M. Abbas, S. Baeg, and S. Park, "Multiple cell upsets tolerant content-addressable memory," in Proc. 49th IEEE Int. Rel. Physics Symp., Monterey, CA, US, Apr. 2011, pp. 863–867.

[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., Berkeley, CA, US, May 2007, pp. 349–354.

[10] M. Richter, K. Oberlaender, and M. Goessel, "New linear SEC-DED codes with reduced triple bit error miscorrection probability," in Proc. 14th IEEE Int. On-Line Testing Symp., Rhodes, Greece, 2008, pp. 37–42.

[11] R. Datta and N. A. Touba, "Exploiting unused spare columns to improve memory ECC," in Proc. 27th IEEE VLSI Test Symp., Santa Cruz, CA, US, May 2009, pp. 47–52.

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

[13] 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.

[14] W. Adi, "Fast Burst Error-Correction Scheme with Fire Code," IEEE Trans. Comput., no. 7, pp. 613–618, 1984.