Microsystem Technologies
https://doi.org/10.1007/s00542-019-04366-7

TECHNICAL PAPER

Compact and power efficient SEC-DED codec for computer memory

Jagannath Samanta[1] · Jaydeb Bhaumik[2] · Soma Barman[3]

Received: 10 February 2017 / Accepted: 21 February 2019

Abstract
Frequently, soft errors occur due to striking of radioactive particles in memory cells which reduce the reliability of memory systems. Generally, single error correction-double error detection (SEC-DED) codes are employed to detect and correct the soft errors in semiconductor memory systems. In this paper, a new optimization algorithm is proposed based on common sub-expression elimination method. By employing this proposed optimization algorithm, more simplified expressions for encoder and decoder are obtained from parity check matrix (*H*-matrix). Proposed optimization technique has been used to implement seven different SEC-DED codecs with message length of 8 bits, 16 bits, 32 bits, 64 bits, 128 bits, 256 bits and 512 bits. Compact design requires at most 21.66% lesser number of two-input XOR gates compared to related SEC-DED codes. All codec architectures are simulated and synthesized using both FPGA and ASIC platforms. Area and power consumption of proposed designs are reduced compared to the existing design without affecting its speed. Proposed designs are beneficial in computer memory systems due to its compactness and lower power consumption.

1 Introduction

Soft errors put a constraint on the design of any high performance semiconductor memory devices. These errors are mainly random, non-destructive and rarely catastrophic type (Baumann 2005). Soft errors occur due to striking of highly energetic alpha and cosmic particles on a register or memory cell's storage node (Chen and Hsiao 1984; Saleh et al. 1990). As a result, memory cells are corrupted due to unintentional change in the logical value of the stored information which creates a major problem in computing system. These errors reduce the system reliability and data integrity of computer memory (Penzo et al. 1995). Soft errors are occurred as single event upsets (SEUs) which can be either SBUs (single bit upsets: flips one bit information due to the passage of single energetic radiation particle) or MBUs (multiple bit upsets: flips of several bits in a word due to the passage of one or more radiation particles). In high density memory devices, multiple cell upsets (MCUs) occur due to single radiating particle (Bajura et al. 2007) which may upset the neighbouring cells. This may happens due to lowering of distances between the two sensitive nodes of the memory cells in deep submicron technology. MBUs are a particular case of MCUs in which more than one bit flipping occurs within a word. Four different types of MCUs that may appear in a static random access memory (SRAM) are discussed in (Tsiligiannis et al. 2014).

Both SEUs and MCUs can be reduced using different techniques (Satoh et al. 2000; Reviriego et al. 2010, 2016a, b). Wei et al. (2016), authors proposed a new scheme to mitigate the single event or multiple bit upsets (SEU/MBU) in a non-volatile SRAM memory. One of the widely used techniques to mitigate these upsets in memory cell is error correction codes (ECCs). Generally, Hamming code (Hamming 1950), Hsaio code (Hsiao 1970), Matrix code (Argyrides et al. 2010), Bose-Chaudhuri-Hocquenghem (BCH) code (Naseer and Draper 2008), Reed Solomon (RS) code (Pontarelli et al. 2015), Convolution code (Rastogi et al. 2009), Golay code (Bajura et al. 2007), Low density parity code (Kang and Park 2006), Orthogonal Latin square code (Demirci et al. 2016), etc., are employed to protect memory against different types of soft-errors. Reed Solomon code and Golay code can deal with multiple errors, but they introduce large area, power and delay overheads due to their complex encoding and decoding schemes. Convolution code can correct double errors via sequential encoding and decoding, but it cannot correct

✉ Jagannath Samanta
jagannath19060@gmail.com

[1] Haldia Institute of Technology, Haldia, India

[2] Department of ETCE, Jadavpur University, Kolkata, India

[3] Institute of Radio Physics and Electronics, University of Calcutta, Kolkata 700009, India

Published online: 28 February 2019

triple adjacent errors. The matrix code can correct adjacent errors, but it cannot correct non-adjacent errors.

But single error correction-double error detection (SEC-DED) codes are generally employed to protect the computer memory from different types of soft errors. In (Reviriego et al. 2013), additional parity bits are used in a high speed SEC-DED but overall area complexity increases. So designing a compact and low power codec is the main objective for any designer.

The main contributions in this paper are as follows: (a) A new optimization algorithm based on common subexpression elimination method is proposed. (b) The proposed algorithm has been employed in Reviriego et al. SEC-DED code. (c) The proposed design consumes lesser area and power compared to existing related design (Reviriego et al. 2013) without affecting its critical path delay. (d) Seven SEC-DED codes with message length of 8, 16, 32, 64, 128, 256 and 512 bits have been designed and implemented both on FPGA and ASIC platforms.

The rest of this paper is organized as follows. Literature review is presented in Sect. 2. Section 3 describes the design of conventional SEC-DED codes. Detail design of proposed codecs is discussed in Sect. 4. In Sect. 5, the performances of proposed designs are analyzed in terms of area, delay and power. Finally, the paper is concluded in Sect. 6.

2 Literature review

The Hamming code (Hamming 1950) is a single error correction (SEC) code which is commonly used to protect standard memories and circuits to detect and correct single bit error. Reviriego et al. introduced an extended SEC code to protect up to seven additional control bits in networking applications. The most commonly used error correction codes (ECC) for computer memory system is single error correction-double error detection (SEC-DED). Original SEC-DED code was constructed from extended Hamming code in the year 1950 (Hamming 1950), where an extra parity bit was added to form a distance-4 code. This SEC-DED code is mostly used in computer memory to enhance the reliability (Hamming 1950). It is typically used to protect each memory word, so that a single bit error in a word can be corrected and two bits errors can be detected. These codes are very popular in semiconductor memory due to its simplicity and least encoding and decoding circuit complexities. These SEC-DED codes have a nominal impact in terms of area and latency to the memory system.

In 1970, Hsaio proposed an odd-weight SEC-DED code (Hsiao 1970) and showed that by using minimum odd weight columns, the number of 1's in the parity check matrix (H-matrix) could be minimized than extended Hamming code (Hamming 1950). This translates to lesser area in the related ECC circuitry. Moreover, by selecting the odd weight columns in a way which balances the number of 1's in each row of the H-matrix, the delay of the checker can be minimized. This code improves the speed, cost and reliability of the decoding logic circuit.

Gherman et al. (2011) proposed a generalized SEC-DED code based on the concept of extended Hamming and Hsaio code. This codec is area efficient compared to extended Hamming and Hsaio code. Sanghhn et al. (Cha and Yoon 2012) proposed a SEC-DED code based on odd weight SEC-DED code. In this code, during memory write operation, the H-matrix is same as that of the odd-weight column code (Hsiao 1970) and during read operation 0's are replaced by 1's in the last row of the H-matrix of the odd-weight-column code (Hsiao 1970). This technique requires lesser hardware and latency compared to the conventional SEC-DED code (Hamming 1950).

Reviriego et al. proposed a technique to design a SEC and SEC-DED code which has low encoding and decoding delay (Reviriego et al. 2013). They had modified the Hamming's generator matrix and reduced numbers of 1's. Thus number of gates required to design encoder and decoder circuits are reduced. This design is compact and faster compared to extended Hamming code (Hamming 1950) and generalized SEC-DED code (Gherman et al. 2011). This scheme is applicable for any data block size and require lesser area for encoding and decoding with lower delay. But this method requires more check bits than conventional codes.

Recently, SEC-DED codes are introduced which can correct a single bit error when double bit errors occur in a memory (Liu et al. 2017). These codes are also able to recover some of the critical bits from certain double bits error patterns. Authors provided the "percentage of double error patterns that can be corrected" and "synthesis results" for four different schemes namely no recovery, scheme1, scheme2 and scheme3 (Liu et al. 2017).

Advancement in SEC-DED codes are also carried out for more errors correction and detection capacity. Single error correction-double error detection–double adjacent error correction (SEC-DED-DAEC) codes are also used in computer memory systems (Dutta and Touba 2007; Ming et al. 2011; Reviriego et al. 2014). Recently a fast and systematic single and adjacent error correction (SAEC) code has been proposed to protect the critical bits by reducing the information bits in the H-matrix (Namba and Lombardi 2018).

3 Design of conventional SEC-DED codec

Design of conventional SEC-DED codecs are discussed in this section. The error correcting codes are employed to protect memories which are mostly systematic in nature (Richter et al. 2008; Neale and Sachdev 2013). A systematic linear block code takes k-message bits and generates n-bits codeword. The $(n - k)$ numbers of parity bits are generated during encoding process.

The H-matrix plays an important role for architectural design of SEC-DED codec. The H-matrix of SEC-DED codes are selected based on following four constraints: (1) there is no all-0 column. (2) each column is distinct from other columns. (3) total number of 1's should be minimal (to reduce area complexity) and (4) numbers of 1's in each row should be an integer as close to the ratio of total numbers of 1's to the number of rows (to reduce latency).

Table 1 shows the numbers of 1's in the H-matrices of different existing SEC-DED codes. Here, we have considered six existing codes namely extended Hamming code, Hsaio et al. code, Gherman et al. code, Cha and Yoon code, Reviriego et al. code and Liu et al. code. It is observed that least numbers of 1's are found in Reviriego et al. code for message length of 64 bits.

The encoding operation can be represented as a matrix multiplication of a generator matrix (G) with message bits. The decoding of SEC-DED code is performed in three steps namely: (1) syndrome computation, (2) locating error position and (3) error correction. The decoding operation begins with the syndrome computation that is obtained by multiplying the stored word with the parity check matrix. There are three consequences. (1) if no error occurs then all syndrome values are zero. (2) when single bit error occurs then all syndrome value is equal to the column of the H-matrix that corresponds to the erroneous bit. Therefore, when all the columns of H-matrix are independent then single error can be located and corrected. (3) when double bit errors occur then the syndrome value is equal to the modulo-2 sum of the columns of the affected bits of H-matrix. For double error detection, commonly odd-weight columns of the parity check matrix are used. Odd-weight syndromes are considered to correspond to single bit error. Finally, estimated error pattern is added with the stored data bits to obtain corrected codeword. There are number of SEC-DED codes available in literature. In this work, we have considered the SEC-DED code proposed by Reviriego et al. (Reviriego et al. 2013), so for the sake of completeness, it has been briefly described in the following subsection.

3.1 Design of Reviriego et al. codec

This SEC-DED code is suitable for protecting memories or caches where speed is more important than area. Figure 1 shows the generator matrix of Reviriego et al. SEC-DED (22, 16) code. The first sixteen columns are identity matrix and last six columns correspond to parity matrix. The H-matrix can be easily obtained from G-matrix in Fig. 1. Figure 2 shows the parity check matrix (H-matrix) of 16 bit Reviriego et al. SEC-DED code, where first sixteen columns constitute the parity matrix and last six columns correspond to identity matrix.

During the write operation of memory using Reviriego et al. SEC-DED (22, 16) code, the six write_check bits $(p_1, \ldots, p_6)$ are generated using the modulo-2 sums of the message bits $(i_1, \ldots, i_{16})$.

The corresponding write_check bits $(p_1, \ldots, p_6)$ are expressed as:

$$\begin{aligned}
p_1 &= i_{11} \oplus i_{12} \oplus i_{13} \oplus i_{14} \oplus i_{15} \oplus i_{16} \\
p_2 &= i_5 \oplus i_6 \oplus i_7 \oplus i_8 \oplus i_9 \oplus i_{10} \\
p_3 &= i_2 \oplus i_3 \oplus i_4 \oplus i_8 \oplus i_9 \oplus i_{10} \oplus i_{14} \oplus i_{15} \oplus i_{16} \\
p_4 &= i_1 \oplus i_3 \oplus i_4 \oplus i_6 \oplus i_7 \oplus i_{10} \oplus i_{12} \oplus i_{13} \oplus i_{16} \\
p_5 &= i_1 \oplus i_2 \oplus i_4 \oplus i_5 \oplus i_7 \oplus i_9 \oplus i_{11} \oplus i_{13} \oplus i_{16} \\
p_6 &= i_1 \oplus i_2 \oplus i_3 \oplus i_5 \oplus i_6 \oplus i_8 \oplus i_{11} \oplus i_{12} \oplus i_{14},
\end{aligned} \tag{1}$$

where $\oplus$ denotes the modulo-2 sum or XOR operation. Total 42 numbers of XOR2 gates are required to implement Eq. (1).

Table 1 Total numbers of 1's in H-matrix of different existing SEC-DED codes

k	Hamming (1950)	Hsiao (1970)	Gherman et al. (2011)	Cha and Yoon (2012)	Reviriego et al. (2013)	Liu et al. (2017)
8	35	29	27	36	29	29
16	65	54	51	67	54	54
32	126	103	104	128	103	103
64	258	216	216	261	201	216
128	545	481	461	566	395	–
256	1153	1050	1010	1215	781	–
512	2508	2241	2182	2566	1552	–

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

Fig. 1 Generator matrix (G) of Reviriego et al. SEC-DED (22, 16) code

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

Fig. 2 Parity check matrix (H) of SEC-DED (22, 16) Reviriego et al. code

During the memory read operation, the read_check bits $(p'_1, \ldots, p'_6)$ are recomputed using the modulo-2 sums of the stored data bits $(r_1, \ldots, r_{16})$ and their corresponding expressions are as follows:

$$\begin{aligned}
p'_1 &= r_{11} \oplus r_{12} \oplus r_{13} \oplus r_{14} \oplus r_{15} \oplus r_{16}\\
p'_2 &= r_5 \oplus r_6 \oplus r_7 \oplus r_8 \oplus r_9 \oplus r_{10}\\
p'_3 &= r_2 \oplus r_3 \oplus r_4 \oplus r_8 \oplus r_9 \oplus r_{10} \oplus r_{14} \oplus r_{15} \oplus r_{16}\\
p'_4 &= r_1 \oplus r_3 \oplus r_4 \oplus r_6 \oplus r_7 \oplus r_{10} \oplus r_{12} \oplus r_{13} \oplus r_{16}\\
p'_5 &= r_1 \oplus r_2 \oplus r_4 \oplus r_5 \oplus r_7 \oplus r_9 \oplus r_{11} \oplus r_{13} \oplus r_{16}\\
p'_6 &= r_1 \oplus r_2 \oplus r_3 \oplus r_5 \oplus r_6 \oplus r_8 \oplus r_{11} \oplus r_{12} \oplus r_{14},
\end{aligned} \tag{2}$$

The syndromes $(s_1, \ldots, s_6)$ are generated using the modulo-2 sums of the write_check bits $(p_1, \ldots, p_6)$ and read_check bits $(p'_1, \ldots, p'_6)$. The syndromes $(s_1, \ldots, s_6)$ of Reviriego et al. SEC-DED (22, 16) codes are presented in Eq. (3):

$$\begin{aligned}
s_1 &= p'_1 \oplus p_1\\
s_2 &= p'_2 \oplus p_2\\
s_3 &= p'_3 \oplus p_3\\
s_4 &= p'_4 \oplus p_4\\
s_5 &= p'_5 \oplus p_5\\
s_6 &= p'_6 \oplus p_6.
\end{aligned} \tag{3}$$

Depending on these syndromes values, the SEC-DED decoder decides whether error occurs or not. The final corrected data bits are as follows:

$$\begin{aligned}
dc_1 &= r_1 \oplus s_4 \cdot s_5 \cdot s_6\\
dc_2 &= r_2 \oplus s_3 \cdot s_5 \cdot s_6\\
&\ldots\\
dc_{15} &= r_{15} \oplus s_1 \cdot s_3 \cdot s_5\\
dc_{16} &= r_{16} \oplus s_1 \cdot s_3 \cdot s_4,
\end{aligned} \tag{4}$$

where $dc_1, \ldots, dc_{16}$ are corrected output message bits and '.' denotes the AND-ing operation.

In Reviriego et al. code, these expressions are directly implemented without any optimization. We have optimized these expressions using proposed optimization algorithm. In the following section, a compact design technique is described.

4 Proposed design of SEC-DED code

Propagation delay, area and power dissipation vary with encoder and decoder circuits complexities. The delay includes both encoding (writing) and decoding (reading) delay in memory circuits. The area, delay and power are also varied due to the variation of encoding and decoding circuits and number of redundant bits. The number of redundant bits is an important factor which indicates the number of bits to be added in each memory word. In this respect, proposed codec design is necessary for memory devices. In this paper, we have proposed an optimization algorithm which can be employed to implement the equations obtained from the H-matrix of SEC-DED code.

4.1 Proposed optimization algorithm

Our proposed optimization technique based on common sub-expression elimination method is described here. The notations are denoted as, n = total number of columns in H-matrix (length of codeword), k = number of columns in H-matrix except identity matrix (length of message bits or word length), r = total number of rows in H-matrix (number of parity bits = $n - k$).

The steps of our proposed optimization algorithm are as follows.

4.1.1 Steps of proposed algorithm

Step 1 Start with the first row of *H*-matrix and search for at least two non-zero elements (i.e. '1').

Step 2 Compare these non-zero element's positions with all other successive rows of same positions.

Step 3 If at least two or more positions are matching between two successive rows, then replace with a new variable.

Step 4 Repeat steps 1–3 for unused non-zero elements of row-1.

Step 5 Repeat steps 1–4 for remaining (*r-1*) rows in the *H*-matrix.

4.1.2 Pseudo code of proposed algorithm

The pseudo code representation of our algorithm is given in Fig. 3.

4.1.3 Explanation of proposed algorithm

Two loop variables *i* and *j* are used to search row and column position of the *H*-matrix respectively. These variables will start their searching from first row and first column element of *H*-matrix. The inner loop will search when all column elements in a same row in the *H*-matrix and the outer loop will continue it searching for all the row elements. In this pseudo code, *u* is another loop variable which represents the next row position ($i + 1$). Here, $[h_{i,j}]$ and $[h_{i+u,j}]$ represents the two successive rows of same positions of *H*-matrix. We have replaced with a new variable if two or more non-zero elements are same in two successive rows in the *H*-matrix. We have applied proposed algorithm for designing *H*-matrix. By employing this proposed optimization algorithm, more simplified expressions for SEC-DED encoder and decoder circuits are obtained from *H*-matrix.

```
for (i = 1; i ≤ r; i = i + 1)
{
for ( j = 1; j ≤ k; j = j + 1)
{
Check the non − zero elements in row - i of H matrix
for (u = 1; u ≤ r − i; u = u + 1)
{
if two or more match found between [h_{i,j}] and
[h_{i+u,j}] then replace by a new variable
}
}
}
```

Fig. 3 Pseudo code representation of proposed algorithm

4.2 Design of SEC-DED (22, 16) codec

Now from this *H*-matrix, Eq. (5) is obtained. The number of XOR2 gates is reduced for implementing the encoder and decoder circuit. The optimized expressions for 16 bits proposed SEC-DED code are as follows:

$$\begin{aligned}
d_1 &= i_{14} \oplus i_{15} \oplus i_{16} \\
d_2 &= i_{12} \oplus i_{13} \\
d_3 &= i_8 \oplus i_9 \oplus i_{10} \\
d_4 &= i_6 \oplus i_7 \\
d_5 &= i_3 \oplus i_4 \\
d_6 &= i_1 \oplus i_2 \oplus i_5 \oplus i_{11} \\
po_1 &= i_{11} \oplus d_2 \oplus d_1 \\
po_2 &= d_4 \oplus i_5 \oplus d_3 \\
po_3 &= i_2 \oplus d_5 \oplus d_3 \oplus d_1 \\
po_4 &= i_1 \oplus d_5 \oplus d_4 \oplus i_{10} \oplus d_2 \oplus i_{16} \\
po_5 &= d_6 \oplus i_4 \oplus i_7 \oplus i_9 \oplus i_{13} \oplus i_{15} \\
po_6 &= d_6 \oplus i_3 \oplus i_6 \oplus i_8 \oplus i_{12} \oplus i_{14},
\end{aligned} \tag{5}$$

where $(d_1, \ldots, d_6)$ are new variable and $(po_1, \ldots, po_6)$ are optimized write_check bits during memory write operation. To implement Eq. (5), total 32 XOR2 gates are required. In case of un-optimized design total 42 XOR2 gates are required to implement 16 bits SEC-DED encoder. Figure 4 shows the compact SEC-DED (22, 16) encoder circuits. This circuit mainly consists of XOR2 gates. Critical path of SEC-DED (22, 16) encoder follows through po_4 or po_5 or po_6.

During read operation, the read_check bits $(po'_1, \ldots, po'_6)$ are recomputed from stored data $(r_1, \ldots, r_{16})$ which are similar to write_check bit computation in encoder block.

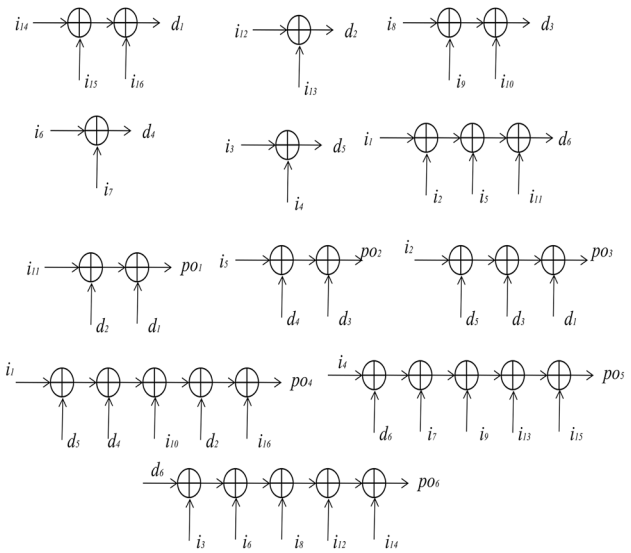

Fig. 4 Write_check bits generator

Syndrome bits $(so_1, \ldots, so_6)$ of SEC-DED (22, 16) code are computed by doing simple modulo-2 sum of respective check bits $(po_1, \ldots, po_6)$ and $(po'_1, \ldots, po'_6)$.

The syndrome bits $(so_1, \ldots, so_6)$ are as follows:

$$\begin{aligned} so_1 &= po'_1 \oplus po_1 \\ so_2 &= po'_2 \oplus po_2 \\ so_3 &= po'_3 \oplus po_3 \\ so_4 &= po'_4 \oplus po_4 \\ so_5 &= po'_5 \oplus po_5 \\ so_6 &= po'_6 \oplus po_6. \end{aligned} \tag{6}$$

Figure 5 shows the syndrome generator circuits of SEC-DED (22, 16) decoder. This circuit is fundamentally similar to write_check bits generator with an additional XOR2 gate.

The corrected data bits are represented as follows:

$$\begin{aligned} dco_1 &= r_1 \oplus (so_4 \cdot so_5 \cdot so_6) \\ dco_2 &= r_2 \oplus (so_3 \cdot so_5 \cdot so_6) \\ &\ldots \\ dco_{15} &= r_{15} \oplus (so_1 \cdot so_3 \cdot so_5) \\ dco_{16} &= r_{16} \oplus (so_1 \cdot so_3 \cdot so_4), \end{aligned} \tag{7}$$

where $r_1, \ldots, r_{16}$ are received bits, $dco_1, \ldots, dco_{16}$ are corrected output message bits.

Figure 6 shows the error corrector circuit of compact SEC-DED (22, 16) decoder. This circuit consists of a single XOR2 gate. This circuit will retrieve the original message from single bit affected received messages.

Theoretical complexities have been compared in the following subsection.

4.3 Theoretical complexity of SEC-DED (22, 16) codes

4.3.1 Area complexity

In Reviriego et al. code, area complexity in terms of XOR2 gates has been calculated using the Eq. (8).

Total number (N_t) of XOR2 gates required during encoding (memory write operation) and decoding (memory read operation) are calculated using following equation.

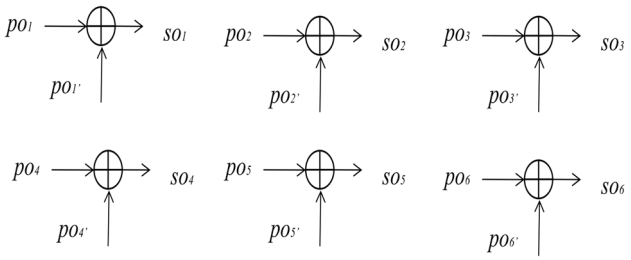

Fig. 5 Syndrome generator

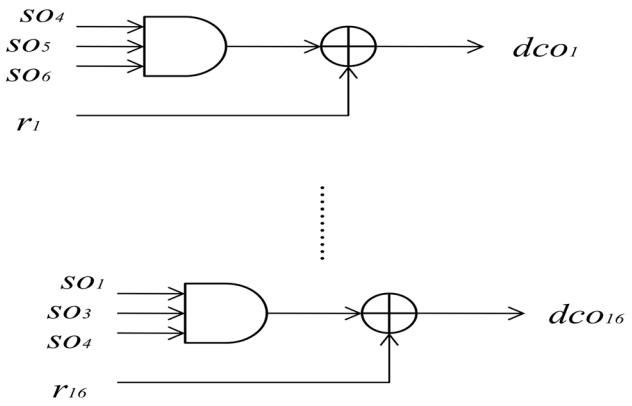

Fig. 6 Error corrector circuits

$$\begin{aligned} N_t &= (H_t - 2 \times r) + (H_t - r) + k, \\ N_t &= 2 \times H_t - 3 \times r + k, \end{aligned} \tag{8}$$

where H_t is the total numbers of 1's in H-matrix, r is the number of parity bits and k is word length.

Theoretical complexities in terms of only XOR2 gates of SEC-DED codes are provided in Table 2. Proposed optimization algorithm has been employed for designing extended Hamming code, Hsaio et al. code, Gherman et al. code, Cha and Yoon code, Reviriego et al. code and Liu et al. code. It is observed that proposed design has lowest hardware complexity compared to existing architectures. In this paper, we have considered the "no recovery scheme" in case of Liu et al. code (2017).

Improvement of our SEC-DED codec is represented in Table 3. It is noted that the reduction in number of XOR2 gates for proposed design has maximum 49.14, 43.66, 42.29, 47.12, 21.66 and 27.56% than extended Hamming (1950), Hsaio (1970), SEC-DED (Gherman et al. 2011), SEC-DED (Cha and Yoon 2012), Reviriego et al. (2013), and Liu et al. (2017) codes respectively. It is also observed that more improvement is achieved as codeword length increases.

The overall area complexities of three SEC-DED codecs (Reviriego et al. 2013; Liu et al. 2017 and our design) are also represented in Table 4. This table analyses the area complexity in terms of all basic logic gates like 2-input XOR gate (XOR2), 2-input AND gate (AND2), 2-input OR gate (OR2), NOT gate (NOT). The percentages of improvement in terms of equivalent 2-input NAND gates (NAND2) are also provided in the Table 4. It is observed that improvement in terms of NAND2 is minimum 13.56% (35.24%) and maximum 18.95% (47.81%) compared to Reviriego et al. code (Liu et al. code) respectively.

Table 2 Area complexity in terms of XOR2 gates for different SEC-DED codes

k	Hamming (1950)	Hsiao (1970)	Gherman et al. (2011)	Cha and Yoon (2012)	Reviriego et al. (2013)	Liu et al. (2017)	Our design
8	63	51	47	58	51	55	43
16	128	106	100	119	106	111	86
32	263	217	219	242	217	223	173
64	556	472	472	517	439	479	347
128	1191	1063	1023	1148	885	–	699
256	2532	2326	2246	2491	1779	–	1397
512	5495	4961	4843	5286	3568	–	2795

Table 3 Percentage improvement in terms of only XOR2 gates of compact design over existing schemes

k	% improv over (Hamming 1950)	% improv. over (Hsiao 1970)	% improv. over (Gherman et al. 2011)	% improv. over (Cha and Yoon 2012)	% improv. over (Reviriego et al. 2013)	% improv. over (Liu et al. 2017)
8	31.75	15.69	8.51	25.86	15.69	21.82
16	32.81	18.87	14.00	27.73	18.87	22.52
32	34.22	20.28	21.00	28.51	20.28	22.42
64	37.59	26.48	26.48	32.88	20.96	27.56
128	41.31	34.24	31.67	39.11	21.02	–
256	44.83	39.94	37.80	43.92	21.47	–
512	49.14	43.66	42.29	47.12	21.66	–

Table 4 Overall area complexity of three different SEC-DED codecs

k	(Reviriego et al. 2013)			(Liu et al. 2017)					Our design			% of improv. over (Reviriego et al. 2013)	% of improv. over (Liu et al. 2017)
	XOR2	AND2	Total equiv. NAND2	XOR2	AND2	OR2	NOT	Total Equiv. NAND2	XOR2	AND2	Total equiv. NAND2		
8	51	16	236	55	33	4	17	315	43	16	204	13.56	35.24
16	106	32	488	111	81	5	49	670	86	32	408	16.39	39.10
32	217	64	996	223	193	6	129	1425	173	64	820	17.67	42.46
64	439	128	2012	479	449	7	315	3150	347	128	1644	18.29	47.81
128	885	256	4052	–	–	–	–	–	699	256	3308	18.36	–
256	1779	512	8140	–	–	–	–	–	1397	512	6612	18.77	–
512	3568	1024	16,320	–	–	–	–	–	2795	1024	13,228	18.95	–

4.3.2 Critical path delay

Theoretical critical path delay of encoder and decoder block of Reviriego et al. code, Liu et al. code and our design are presented in Table 5. It is observed that the critical path delay of proposed design is similar to the Reviriego et al. code (2013). Critical path delay of our design is lower than Liu et al. code (2017).

5 Synthesis results

In this section, FPGA and ASIC based simulation and synthesis results are presented. All the functional blocks of SEC-DED codecs are written using Verilog hardware description language (HDL) code.

5.1 FPGA based synthesis results

Existing and proposed architectures are simulated and synthesized using FPGA based vertex6 device family using Xilinx14.7 tool. FPGA based synthesis results are shown in

Table 5 Theoretical critical path analysis of different SEC-DED codecs

k	Reviriego et al. (2013)		Liu et al. (2017)		Our design	
	Enc.	Dec.	Enc.	Dec.	Enc.	Dec.
8	$5t_{XOR}$	$7t_{XOR}+2t_{AND}$	$5t_{XOR}$	$10t_{XOR}+t_{AND}+4t_{OR}+t_{NOT}$	$5t_{XOR}$	$7t_{XOR}+2t_{AND}$
16	$8t_{XOR}$	$10t_{XOR}+2t_{AND}$	$7t_{XOR}$	$13t_{XOR}+t_{AND}+5t_{OR}+t_{NOT}$	$8t_{XOR}$	$10t_{XOR}+2t_{AND}$
32	$14t_{XOR}$	$16t_{XOR}+2t_{AND}$	$13t_{XOR}$	$20t_{XOR}+t_{AND}+6t_{OR}+t_{NOT}$	$14t_{XOR}$	$16t_{XOR}+2t_{AND}$
64	$24t_{XOR}$	$26t_{XOR}+2t_{AND}$	$25t_{XOR}$	$33t_{XOR}+t_{AND}+7t_{OR}+t_{NOT}$	$24t_{XOR}$	$26t_{XOR}+2t_{AND}$
128	$39t_{XOR}$	$41t_{XOR}+2t_{AND}$	–	–	$39t_{XOR}$	$41t_{XOR}+2t_{AND}$
256	$62t_{XOR}$	$64t_{XOR}+2t_{AND}$	–	–	$62t_{XOR}$	$64t_{XOR}+2t_{AND}$
512	$101t_{XOR}$	$103t_{XOR}+2t_{AND}$	–	–	$101t_{XOR}$	$103t_{XOR}+2t_{AND}$

Table 6 FPGA synthesis result based on Slice LUT

k	Reviriego et al. (2013)	Liu et al. (2017)	Our design	% improv. over (Reviriego et al. 2013)	% improv. over (Liu et al. 2017)
8	20	23	18	10.00	21.74
16	42	50	38	9.52	24.00
32	81	89	73	9.88	17.98
64	154	170	138	10.39	18.82
128	310	–	280	9.68	–
256	606	–	541	10.73	–
512	1243	–	1109	10.78	–

Table 7 Delay analysis of different SEC-DED codec

k	Reviriego et al. (2013)		Liu et al. (2017)		Our design	
	Enc. (ns)	Dec. (ns)	Enc.	Dec.	Enc. (ns)	Dec. (ns)
8	1.114	1.714	1.114	2.221	1.114	1.714
16	1.732	2.274	1.348	2.672	1.740	2.284
32	1.871	2.522	1.741	3.184	1.871	2.522
64	2.171	2.572	1.880	3.517	2.175	2.574
128	2.498	3.297	–	–	2.499	3.232
256	3.188	3.644	–	–	3.191	3.402
512	4.532	4.689	–	–	4.541	4.659

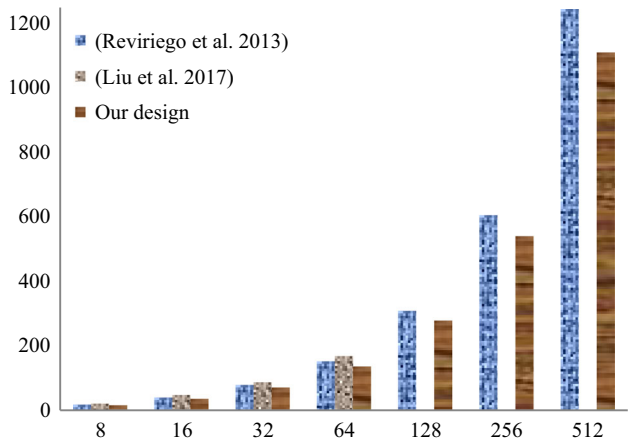

Fig. 7 LUT requirement in Reviriego et al. and proposed codec

Table 6 for compact design of Reviriego et al. code. Here area is represented in terms of Look-Up-Table (LUT). We have also compared our design in terms of percentage improvement over existing design. It is observed that proposed design provides significant area savings for $k = 512$ (10.78%). It is also observed that our design requires lesser LUTs compared to Liu et al. code (2017).

Table 7 represents the synthesized SEC-DED encoder and decoder delay (in ns) of Reviriego et al., Liu et al. and proposed designs. Estimated delay (in ns) for both the encoder (Enc.) and the decoder (Dec.) of SEC-DED codes are presented in Table 7. It is observed that speed of the proposed design is comparable to Reviriego et al. design. So the delay of our design and Reviriego et al. design is approximately equal. Our design is faster than Liu et al. design.

Table 8 Area estimation (in μm^2) for ASIC based synthesis of SEC-DED codec

k	Reviriego et al. (2013)	Our design	% improv. over (Reviriego et al. 2013)
8	1171	1131	3.42
16	2309	2129	7.80
32	4996	4621	7.51
64	9845	9124	7.32
128	19,742	18,312	7.24
256	38,740	36,344	6.18
512	80,896	74,254	8.21

Table 9 Total power estimates (in mW) for ASIC synthesis of SEC-DED codec

k	Reviriego et al. (2013)	Our design	% improv. over (Reviriego et al. 2013)
8	0.15	0.13	13.33
16	0.37	0.34	10.01
32	1.11	0.97	12.21
64	2.67	2.42	9.38
128	6.56	5.86	10.63
256	15.58	14.25	8.56
512	34.12	29.95	12.22

Figure 7 shows the comparison in terms of LUT requirements for implementing Reviriego et al. code, Liu et al. code and our design. The figure indicates that lesser numbers of LUTs are required in proposed design compared to Reviriego et al. and Liu et al. designs.

5.2 ASIC based synthesis results

Different SEC-DED codecs are synthesized using Cadence Genus synthesis tool (TSMC18 library). ASIC based synthesis results are presented in Tables 8 and 9. From Table 8, it is observed that proposed design saves significant area over existing design in (Reviriego et al. 2013). Maximum 8.21% and minimum 3.14% improvements are achieved for $k = 512$ and $k = 8$ respectively.

ASIC based synthesis results for total power consumptions are presented in Table 9. Our design has lesser power consumption compared to Reviriego et al. code. Minimum 8.56% and maximum 13.33% power savings are obtained for $k = 256$ and k = 8 respectively.

6 Conclusion

SEC-DED codes are generally employed in computer memory systems to enhance the reliability and data integrity. In these systems, Reviriego et al. codes are preferred due to its lesser encoding and decoding delay. But hardware complexity of this codec is more due to its additional redundant bits. In this respect, proposed designs require lesser hardware compared to the un-optimized designs without affecting its speed. Encoder and decoder circuits are also simulated and synthesized using vertex6 FPGA and ASIC based platforms. Speed of the proposed design is similar to the un-optimized Reviriego et al. codec. Proposed codec architectures require lesser area and power with speed comparable to the existing designs; hence these codecs are suitable for computer memory systems to combat soft errors.

Acknowledgements Authors would like to thank SMDP-C2SD, University of Calcutta for providing Cadence simulation software. Authors also like to thank reviewers for their valuable comments to improve the quality of this paper.

References

Argyrides CA, Reviriego P, Pradhan DK, Maestro JA (2010) Matrix-based codes for adjacent error correction. IEEE Trans Nuclear Sci 57(4):2106–2111

Bajura MA, Boulghassoul Y, Naseer R, Das Gupta S, Witulski AF, Sondeen J, Stansberry SD, Draper J, Massengill LW, Damoulakis JN (2007) Models and algorithmic limits for an ECC-based approach to hardening sub-100-nm SRAMs. IEEE Trans Nucl Sci 54(4):935–945

Baumann R (2005) Radiation-induced soft errors in advanced semiconductor technologies. IEEE Trans Device Mater Reliab 5(3):305–316

Cha S, Yoon H (2012) Efficient implementation of single error correction and double error detection code with check bit pre-computation for memories. JSTS 12(4):418–425

Chen CL, Hsiao MY (1984) Error-correcting codes for semiconductor memory applications: a state-of-the-art review. IBM J Res Dev 28(2):124–134

Demirci M, Reviriego P, Maestro JA (2016) Implementing double error correction orthogonal latin squares codes in SRAM-based FPGAs. Microelectron Reliab 56:221–227

Dutta A, Touba NA (2007) Multiple bit upset tolerant memory using a selective cycle avoidance based SEC-DED-DAEC code. In: 25th IEEE VLSI test symposium (VTS'07), pp 349–354

Gherman V, Evain S, Seymour N, Bonhomme N (2011) Generalized parity-check matrices for SEC-DED codes with fixed parity. In: Proceedings of IEEE on-line testing symposium, pp 198–201

Hamming RW (1950) Error detecting and error correcting codes. Bell Syst Tech J 29(2):147–160

Hsiao MY (1970) A class of optimal minimum odd-weight column SEC-DED codes. IBM J Res Dev 14(4):395–401

Kang SH, Park IC (2006) Loosely coupled memory-based decoding architecture for low density parity check codes. IEEE Trans Circ Syst I: Regul Pap 53(5):1045–1056

Liu S, Reviriego P, Xiao L, Maestro JA (2017) A method to recover critical bits under a double error in SEC-DED protected memories. Microelectron Reliab 73:92–96

Ming Z, Yi XL, Wei LH (2011) New SEC-DED-DAEC codes for multiple bit upsets mitigation in memory. In: 2011 IEEE/IFIP 19th International Conference on VLSI and System-on-Chip, IEEE, pp 254–259

Namba K, Lombardi F (2018) A single and adjacent error correction code for fast decoding of critical bits. IEEE Trans Comput 67(10):1525–1531

Naseer R, Draper J (2008) Parallel double error correcting code design to mitigate multi-bit upsets in SRAMs. In: ESSCIRC 2008–34th European Solid-State Circuits Conference, IEEE, pp 222–225

Neale A, Sachdev M (2013) A new SEC-DED error correction code subclass for adjacent MBU tolerance in embedded memory. IEEE Trans Device Mater Reliab 13(1):223–230

Penzo L, Sciuto D, Silvano C (1995) Construction techniques for systematic SEC-DED codes with single byte error detection and partial correction capability for computer memory systems. IEEE Trans Inf Theory 41(2):584–591

Pontarelli S, Reviriego P, Ottavi M, Maestro JA (2015) Low delay single symbol error correction codes based on reed solomon codes. IEEE Trans Comput 64(5):1497–1501

Rastogi A, Agarawal M, Gupta B (2009) SEU MITIGATION-using 1/3 rate convolution coding. In: 2009 2nd IEEE International Conference on Computer Science and Information Technology, IEEE, pp 180–183

Reviriego P, Maestrom JA, Baeg S, Wen S, Wong R (2010) Protection of memories suffering MCUs through the selection of the optimal interleaving distance. IEEE Trans Nucl Sci 57(4):2124–2128

Reviriego P, Pontarelli S, Maestro JA, Ottavi M (2013) A method to construct low delay single error correction codes for protecting data bits only. IEEE Trans Comput Aided Design Integr Circ Syst 32(3):479–483

Reviriego P, Martínez J, Pontarelli S, Maestro JA (2014) A method to design SEC-DED-DAEC codes with optimized decoding. IEEE Trans Device Mater Reliab 14(3):884–889

Reviriego P, Demirci M, Evans A, Maestro JA (2016a) A method to design single error correction codes with fast decoding for a subset of critical bits. IEEE Trans Circuits Syst II Express Briefs 63(2):171–175

Reviriego P, Liu SS, Sánchez-Macián A, Xiao L, Maestro JA (2016b) Unequal error protection codes derived from SEC-DED codes. Electron Lett 52(8):619–620

Richter M, Oberlaender K, Goessel M (2008) New linear SEC-DED codes with reduced triple bit error miscorrection probability. In: Proc. of IEEE on-line testing symp., pp 37–42

Saleh AM, Serrano JJ, Patel JH (1990) Reliability of scrubbing recovery-techniques for memory systems. IEEE Trans Reliab 39(1):114–122

Satoh S, Tosaka Y, Wender SA (2000) Geometric effect of multiple-bit soft errors induced by cosmic ray neutrons on DRAM's. IEEE Electron Device Lett 21(6):310–312

Tsiligiannis G, Dilillo L, Bosio A, Girard P, Pravossoudovitch S, Todri A, Virazel A, Puchner H, Frost C, Wrobel F, Saigne F (2014) Multiple cell upset classification in commercial SRAMs. IEEE Trans Nucl Sci 61(4):1747–1754

Wei W, Namba K, Kim YB, Lombardi F (2016) A novel scheme for tolerating single event/multiple bit upsets (SEU/MBU) in non-volatile memories. IEEE Trans Comput 1:1