TELKOMNIKA Telecommunication, Computing, Electronics and Control
Vol. 18, No. 5, October 2020, pp. 2533~2542
ISSN: 1693-6930, accredited First Grade by Kemenristekdikti, Decree No: 21/E/KPT/2018
DOI: 10.12928/TELKOMNIKA.v18i5.13345

Modifying Hamming code and using the replication method to protect memory against triple soft errors

Wael Toghuj
Faculty of Information Technology, Al-Ahliyya Amman University, Jordan

Article Info

Article history:

Received Jun 19, 2019
Revised May 1, 2020
Accepted May 11, 2020

Keywords:

Critical applications
Multi-bit upset
Radiation
Reliability
Soft error

ABSTRACT

As technology scaling increases computer memory's bit-cell density and reduces the voltage of semiconductors, the number of soft errors due to radiation induced single event upsets (SEU) and multi-bit upsets (MBU) also increases. To address this, error-correcting codes (ECC) can be used to detect and correct soft errors, while x-modular-redundancy improves fault tolerance. This paper presents a technique that provides high error-correction performance, high speed, and low complexity. The proposed technique ensures that only correct values get passed to the system output or are processed in spite of the presence of up to three-bit errors. The Hamming code is modified in order to provide a high probability of MBU detection. In addition, the paper describes the new technique and associated analysis scheme for its implementation. The new technique has been simulated, evaluated, and compared to error correction codes with similar decoding complexity to better understand the overheads required, the gained capabilities to protect data against three-bit errors, and to reduce the misdetection probability and false-detection probability of four-bit errors.

This is an open access article under the CC BY-SA license.

Corresponding Author:

Wael Toghuj,
Department of Computer Science, Faculty of Information Technology,
Al-Ahliyya Amman University,
Amman, Jordan.
Email: w.toghuj@ammanu.edu.jo

1. INTRODUCTION

As manufacturers continue to shrink the dimensions and operating voltages of computer electronics, the susceptibility to cosmic radiation phenomena is increasing in these advanced electronics, particularly in memory semiconductor devices. Cosmic radiation contains particles from space with energies much greater than 1 GeV, including particles from solar wind. The cosmic radiation reacts with the earth's atmosphere via strong nuclear interactions, producing complex cascades of second and higher generation particles [1, 2]. The NASA study [3] emphasizes that cosmic radiation increases with altitude due to a smaller shielding effect from the atmosphere. The neutron flux at sea level is several hundred times lower than at airplane flight altitudes, as shown in Figure 1 [4].

A recent study from researchers at Harvard University confirms that trace radiation dose rates increase inside an airplane flight from Baltimore to Las Vegas, as shown in Figure 2 [5]. Hence, memory modules in airplanes and other devices on board, for instance implementable medical devices (IMD), are highly susceptible to soft errors by a factor of a few hundred to a few thousand times compared to modules on the ground [6]. Moreover, researchers ensure that at typical altitudes, pilots, crew, and passengers typically receive a dose rate of 40 to 70 times higher than natural radiation on the ground. These doses

increase for passengers flying international routes over the earth's poles [5]. Consequently, the impacts of cosmic radiation on today's electronic devices are considered to be a serious and growing problem. Therefore, mitigating the effects of cosmic radiation is essential for proper operation of these devices.

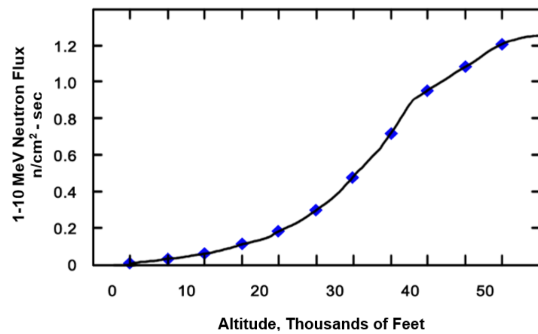


Figure 1. The 1-10 MeV atmospheric neutron flux as a function of altitude based on aircraft and balloon measurements

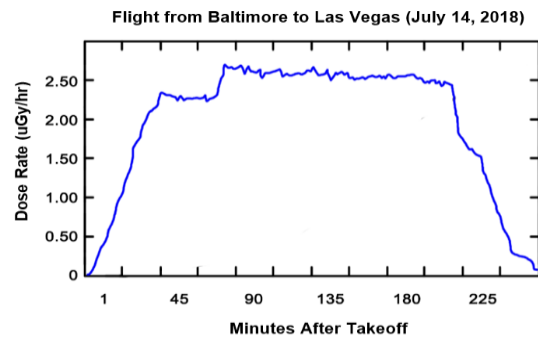


Figure 2. Traces radiation dose rates increased inside the airplane flight after takeoff

The impacts of cosmic radiation may result in either a transient error effect such as a bit flip in memory (e.g. in dynamic/static RAM and commercial electrically erasable and programmable read-only memory (EEPROM) [7]) or a voltage transient in logic, known as single event upset (SEU). The commonly used unit of measure for SER and other hard-reliability mechanisms is the FIT (failure in time). A FIT is equivalent to one failure in one billion hours of device usage. For instance, the ISO 26262 standard for functional safety of road vehicles mandates the overall FIT rate for the deep learning neural networks to be less than 10 FIT [8]. In contrast, without mitigation the radiation effects, the SER can easily exceed 50,000 FITs per chip [9]. An example of the cosmic radiation impact on static RAM shows that the SER per bit tends to worsen by a factor of 5 to 10 for each new process generation because the critical charge decreases faster than the charge-collection efficiency. The SER of six Tera static RAM operating at full speed rapidly exceeds the desirable threshold of 1,000 FITs per Mbit [1].

The second most concerning error is the Multiple Bit Upset (MBU), occurring when a single particle causes the upset of two or more memory cells [10]. Figure 3 shows the SER trend for a range of silicon technology generations reported in terms of FIT [11]. The Nominal curve illustrates past and present trends while the Vscale_L, Vscale_M, and Vscale_H curves assume low, medium, and high amounts (respectively) of voltage scaling in future deep submicron technologies. The user-visible failure rates highlighted at 45 nm and 16 nm are calculated assuming a 92% system-wide masking rate. At the present time, for a typical user of laptop or desktop computers, this phenomenon is imperceptible. However, in the near future, using 16 nm nodes could cause the user-visible fault rate to be as high as one failure per day per chip.

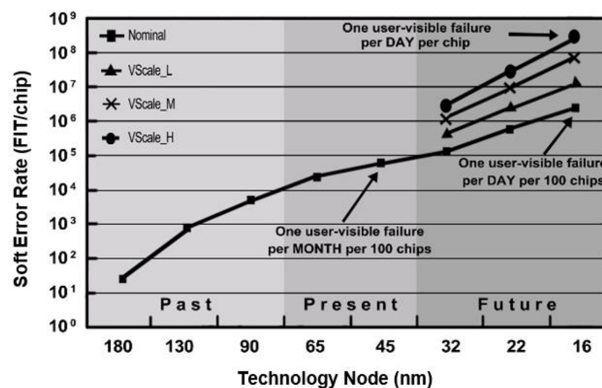
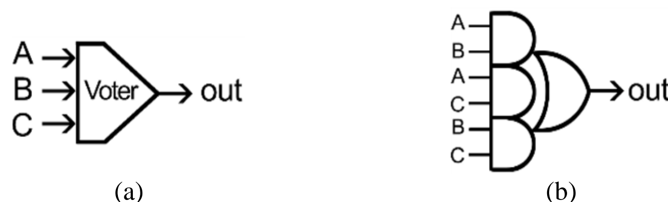


Figure 3. The soft error rate trend for processor logic across a range of silicon technology nodes

As semiconductor chips are used everywhere from standalone to high speed data communication devices, soft errors are considered one of the main threats to the reliability of these devices, and many pioneering works such as [12-15] have reached different results in designing memory architecture with some anti-soft-error capabilities. For mission-critical applications that demand very high reliability, one popular solution is the N modular redundancy with majority voting [16]. Figure 4 shows a 3-input majority voter circuit, where if one of the three inputs change, the state for the output remains true. This approach can provide very high reliability. However, they are too expensive to be applied to microprocessors or embedded in systems such as IMD. The dual classical modular redundancy can provide SER stability by detecting errors. Modifying this mechanism may increase its capabilities as shown in [16-19]. For memory devices, many field studies show that their reliability depends upon device physics and design as well as error correction codes (ECC). SEUs and MBUs have been addressed (e.g. in static RAMs) by using a simple single-error-correcting-double-error-detecting (SEC-DED) ECC, such as Hamming code [20] with minimum Hamming distance D equal to four. Depending on the value of D, which is used to define some fundamental concepts in ECC, the binary code can detect d_t bit errors and correct d_c bit errors:

$$D \geq dt + 1 \quad (1)$$

$$D \geq 2dc + 1 \quad (2)$$



(a) (b)

Figure 4. Majority voter: (a) schematic and (b) standard-cell circuit

On the other hand, the continued rising effect of cosmic radiation upon these devices limits the efficiency of such SEC-DED ECCs, especially for MBUs. Several approaches have been proposed for modifying the Hamming code with D = 4. In Hsiao [21] a special class Hamming code was presented to improve the speed, cost, and reliability of the decoding logic. In Kazeminejad [22], by adding one extra bit, the author improved the Hamming code in terms of area, speed, and power. And in [23] the Hamming code with different parameters (22 bits), (107 bits) and (248 bits) of information data (dataword) was implemented and debugged using field programmable gated array kit with integrated software environments for simulation and testing the results of the hardware system. Such system has the same ability to correct single bit error and detect two bits error. To correct two bits error in memory devices, double adjacent error correction (DAEC) codes can be used. Because of the negative impact on decoder complexity and the probabilities of giving incorrect decoding in some double nonadjacent bit errors, however, the implementation of DAEC is limited. To improve the double errors correction of DAEC, the authors in [24] proposed a new method to mitigate this disadvantage based upon unequal error protection (UEP) codes. Another problem that stands in the way of achieving reliable fault tolerance is the miss-detection probability and false-detection probability of ECCs. In [12] the authors offer a new design solution for dynamic RAM manufacturers to employ ECC to tolerate unrepaired weak memory cells in order to decrease the probabilities of miss-detection and false-detection. On the other hand, the Reed-Solomon code and Bose-Chaudhuri Hocquenghem (BCH) codes are capable of detection and correction of multi-byte errors with very low overhead in terms of additional check bits required [25, 26]. However, applying these codes for memory devices (dynamic/static RAM and EEPROM) to correct MBU results in high encoding and decoding complexity as these codes typically work at the block level and are applied to multiple words at a time. In the next two sections of this paper, we modify Hamming code with D = 4 to increase its capabilities to correct and detect multiple-bit errors, and we combine the modified code with replication methods to achieve a high level of reliability.

2. RESEARCH METHOD

2.1. Proposed technique

In this section, we describe our approach to efficiently protect against soft errors that mostly affect memory. The proposed technique targets the protection of codeword in memory against triple soft errors, and

in some cases, the capability of detecting four errors regardless of the errors' pattern. The technique exploits the replication method and the extended Hamming code with D = 4 (with additional parity), insuring that only correct values get passed to the system output or are processed in spite of the presence up to three-bit errors ($d \leq 3$) in codeword. The practical implementation of the proposed technique involves the inclusion in its structure of one coder to minimize the overheads of the error correction resources and two decoders as shown in Figure 5.

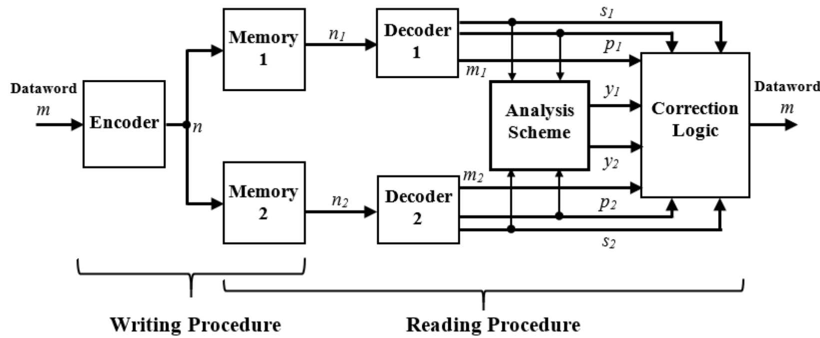


Figure 5. Block diagram of proposed technique based on a combination of replication method (dual modular redundancy) and modified Hamming code

The procedure for writing the dataword m into memory is summarized as following: in the encoder, the dataword is encoded by computing its parities. For instance, if the dataword consists of 16 bits ($m= a_0, a_1, a_2, \ldots, a_{15}$), the encoder generates 6 checkbits ($k=6$), then the codeword ($n=m+k=22$) is written into memory 1 as n_1 and in memory 2 as n_2. The values of parity bits for n_1 (when $m=16$) are determined as:

$$p10 = a0 \oplus a1 \oplus a3 \oplus a4 \oplus a6 \oplus a8 \oplus a10 \oplus a11 \oplus a13 \oplus a15$$
$$p11 = a0 \oplus a2 \oplus a3 \oplus a5 \oplus a6 \oplus a9 \oplus a10 \oplus a12 \oplus a13$$
$$p12 = a1 \oplus a2 \oplus a3 \oplus a7 \oplus a8 \oplus a10 \oplus a14 \oplus a15$$
$$p13 = a4 \oplus a5 \oplus a6 \oplus a7\ a8 \oplus a9 \oplus a10$$
$$p14 = a11 \oplus a12 \oplus a13 \oplus a14 \oplus a15$$
$$p15 = a0 \oplus a1 \oplus a2 \oplus \ldots \oplus a15 \oplus p10 \oplus p11 \oplus p12 \oplus p13 \oplus p14$$

The same structure of generating the parity bits p_{20}, p_{21}, p_{22}, p_{23}, p_{24} and p_{25} is implemented for n_2.

The generation of parity bits is done by a parity generator that uses exclusive-OR (XOR) gates. After that, the codeword is written into memory 1 and memory 2 as shown in Figure 5. For the procedure of reading, firstly, the decoders form the values of syndromes S and parities P (s_1 , s_2 , p_1 and p_2). The value of S is generated by taking an XOR of the data bits and recomputed check bit. Secondly, these values are examined by the analysis scheme that generates control signals. This way, it is possible to distinguish between single-bit errors, two-bit errors, and three-bit errors with only a minimum impact on performance. As a result, the dataword is read from the memory, which does not contain any errors or contains fewer errors, and is corrected by the corresponding decoder. For example, assuming that in memory 1 a three-bits error occurs in the codeword n_1, the signal y_1 prevents reading data from that device in order not to get passed to the system output. Meanwhile, the signal y_2 allows codeword n_2 to get passed from memory 2. The analysis of the list of possible error combinations and the indication of how to respond in each of the possible situations is shown in Table 1. From Table 1, we obtain (3) and (4) for the control signals y_1 and y_2 that determine which memory the data should be read from:

$$\begin{aligned} y_1 &= \bar{s}_1\bar{p}_1\bar{s}_2\bar{p}_2 + \bar{s}_1\bar{p}_1 s_2 p_2 + \bar{s}_1\bar{p}_1 s_2\bar{p}_2 + \\ &+ s_1 p_1 s_2 p_2 + s_1\bar{p}_1\bar{s}_2 p_2 + s_1 p_1 s_2\bar{p}_2 = \\ &= \bar{s}_1\bar{p}_1 + s_1 p_1 s_2 \end{aligned} \quad (3)$$

$$\begin{aligned} y_2 &= s_1 p_1\bar{s}_2\bar{p}_2 + s_1\bar{p}_1\bar{s}_2\bar{p}_2 + \bar{s}_1 p_1\bar{s}_2\bar{p}_2 + s_1\bar{p}_1 s_2 p_2 \\ &= p_1\bar{s}_2\bar{p}_2 + s_1\bar{s}_2 p_2 + s_1\bar{p}_1 s_2 p_2 \end{aligned} \quad (4)$$

Table 1. The decoding algorithm (procedure of reading)

Number of errors			Reaction of Decoder				Performing the Reading process is realized from	
			Decoder 1		Decoder 1		Memory 1	Memory 2
Total Errors	Memory 1	Memory 2	S_1	P_1	S_2	P_2	$(y1 = 1 \& y2 = 0)$	$(y1 = 0 \& y2 = 1)$
0	0	0	0	0	0	0	+	
1	1	0	1	1	0	0		+
	0	1	0	0	1	1	+	
2	2	0	1	0	0	0		+
	0	2	0	0	1	0	+	
	1	1	1	1	1	1	+	
3	3	0	0	1	0	0		+
	0	3	0	0	0	1	+	
	1	2	1	1	1	0	+	
	2	1	1	0	1	1		+

After their simplification, the given Boolean expressions are used to design the analysis scheme as shown in Figure 6. From Figure 6 it follows that this scheme consists of four inverters, five AND gates, and two OR gates. Knowing the characteristics of these logic elements, and also that the typical value of the signal propagation delay time is usually 5-10 ns, we can conclude that the proposed scheme does not complicate the implementation of the decoding algorithm and does not impair its speed. The set of admissible states s_1, s_2, p_1, and p_2, at any moment in time at which the error correction model produces the correct result, is described by the following Boolean expression:

$$\begin{aligned} TRUE = &((s_1 \wedge p_1) \wedge (s_2 \wedge p_2)) \vee ((\bar{s}_1 \wedge \bar{p}_1) \wedge (s_2 \wedge p_2)) \vee \\ &\vee ((s_1 \wedge p_1) \wedge (\bar{s}_2 \wedge \bar{p}_2)) \vee ((\bar{s}_1 \wedge p_1) \wedge (s_2 \wedge p_2)) \vee \\ &\vee ((s_1 \wedge p_1) \wedge (\bar{s}_2 \wedge p_2)) \vee ((\bar{s}_1 \wedge \bar{p}_1) \wedge (\bar{s}_2 \wedge \bar{p}_2)) \vee \\ &\vee ((s_1 \wedge \bar{p}_1) \wedge (s_2 \wedge p_2)) \vee ((s_1 \wedge p_1) \wedge (s_2 \wedge \bar{p}_2)) \vee \\ &\vee ((\bar{s}_1 \wedge \bar{p}_1) \wedge (\bar{s}_2 \wedge p_2)) \vee ((\bar{s}_1 \wedge p_1) \wedge (\bar{s}_2 \wedge \bar{p}_2)). \end{aligned} \quad (5)$$

The set of states s_1, s_2, p_1, and p_2, at any given time at which the error correction model generates a signal of uncorrectable errors is described by the following Boolean expression:

$$FALSE = (\bar{s}_1 \wedge p_1) \wedge (\bar{s}_2 \wedge p_2). \quad (6)$$

We assume that when working with data storage devices, the time is measured in cycles; then, if the data was transferred to the input of the memory device in t cycle, the encode delay which indicates the time required to calculate the check-bits for a write operation should be equal to $t + t_{cd}$ cycle (where t_{cd} is the required time to encode the data). It should be noted that the decode delay t_{dc} depends on the reliability state of the stored data (i.e. the multiplicity of errors and their distribution). If there are no errors, then the value of t_{dc} will be the minimum.

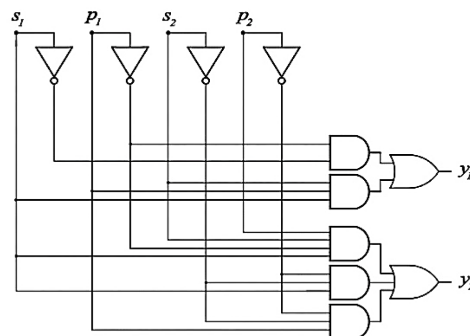


Figure 6. The analysis schemes

2.2. Case study: Decreasing probability of miss-detecting 4-bit errors

The objective of the proposed technique is to enhance the reliability of data against soft errors without compromising performance. To achieve this goal, we have to explore the extended Hamming code to

answer the following important questions. Using Hamming code with a minimum Hamming distance of four ($D = 4$), how do we reduce the probability of false decoding of three-bit errors? Additionally, is it possible to reduce the probability of miss-detecting four-bits errors? If we assume that three bits could have been corrupted using the Hamming code with $D = 4$, then in some cases there is going to be a probability of false decoding of three-bit errors and single errors. In this situation, the value of the syndrome and parity may have the same value ($S \neq 0, P \neq 0$). In addition, in the case of the occurrence of four-bit errors in the codeword, the decoder of Hamming code with $D = 4$ generates the following values: $S = 0$ and $P = 0$. If no errors are found, the values of the syndrome and parity will be zero, and as a result, a four-bit error is skipped, remaining undetected.

To improve the probability of detecting three-bit ($d_t = 3$) and four-bit errors ($d_c = 4$), we begin the study of the Hamming code with the minimum Hamming distance $D = 3$. Binary Hamming codes have parameters: $n = 2k - 1, m = 2k - 1 - k$ and $D = 3$ [20], which are specified using a check matrix, the columns of which are all non-zero binary vectors of length k. For example, for the code (7, 4), the parity check matrix is presented in initial form as shown in Figure 7 (a). Hamming codes with $D = 3$ can either correct single errors or detect double errors. For a single error, syndrome S uniquely indicates the location of the error and is equal to the corresponding column of the check matrix. The code can be shortened by excluding parts of information symbols, which corresponds to crossing out a number of columns in submatrix I. The code can be extended by inserting a general (whole) parity check. In this case, the minimum Hamming distance $D = 4$ makes it possible to correct all single-bit errors with simultaneous detection of double-bit errors. For example, if we extend the code (7, 4), then we get the code matrix (8, 4) as shown in Figure 7 (b). In many cases, it is advisable to modify the Hamming codes to give them additional useful properties that facilitate practical implementation or expand the circle of potential consumers. The main objectives of the modifications are:

- Simplification of coding and decoding devices.
- Ensuring, the same signal delay when encoding and decoding in all bits (structure homogeneity).
- Ensuring the adequacy of byte organization of memory and the ability to increase them.
- Reducing the probability of incorrect decoding for multiple errors.

$$\mathbf{H} = \left|\begin{array}{c:c} \mathbf{C} & \mathbf{I} \\ 0\,1\,1\,1 & 1\,0\,0 \\ 1\,0\,1\,1 & 0\,1\,0 \\ 1\,1\,0\,1 & 0\,0\,1 \end{array}\right| \qquad \mathbf{H} = \left|\begin{array}{c:c} \mathbf{C} & \mathbf{I} \\ 0\,1\,1\,1 & 1\,0\,0\,0 \\ 1\,0\,1\,1 & 0\,1\,0\,0 \\ 1\,1\,0\,1 & 0\,0\,1\,0 \\ 1\,1\,1\,1 & 1\,1\,1\,1 \end{array}\right|$$

(a) (b)

Figure 7. Example (a) code (7,4) with $D = 3$, (b) code (8,4) with $D = 4$

Simplification of coding and decoding devices may be achieved by minimizing the number of ones (positions with values of one) in the check matrix. For the convenience of detecting double errors, the matrix H^T with $D = 4$ is transformed (built) by adding the sum of all the remaining rows to the last row (the code properties do not change from this). As a result, the total number of ones in the matrix decreases and the following useful properties appear: all columns acquire odd weight, therefore, syndromes of all single-bit errors have an odd weight, and double-bit error syndromes have an even weight; the rows contain the same number of ones, which ensures the uniformity of the structure of the coding and decoding schemes of the syndrome when each syndrome is implemented separately. For example, shortening a code reduces the number of columns, and for the code (8, 4) we get:

$$\mathbf{H^T} = \left|\begin{array}{c:c} \mathbf{C} & \mathbf{I} \\ 0\,1\,1\,1 & 1\,0\,0\,0 \\ 1\,0\,1\,1 & 0\,1\,0\,0 \\ 1\,1\,0\,1 & 0\,0\,1\,0 \\ 1\,1\,1\,0 & 0\,0\,0\,1 \end{array}\right|$$

Accordingly, we should exclude the columns in the check matrix H^T with the maximum number of ones, since each one in the matrix is realized by an XOR gate. Consequently, reducing the number of ones in the H^T matrix leads to a decrease in complexity and an increase in the speed of encoding and decoding circuits. Secondly, to increase the probability of correct decoding, it is necessary to exclude columns in the H^T matrix in such a way as to reduce the probability of false correction of three errors:

$$P_3 = \frac{4 \cdot B_4}{C_n^3}, \tag{7}$$

It also shows the probability of miss-detecting four errors:

$$P_4 = \frac{B_4}{C_n^4}, \tag{8}$$

Here, n is the length of the shortened code. B_4 - the number of words that weights 4 in the shortened code. This is the total number of four errors with zero syndrome, and each four-error case that characterizes four types of three errors is falsely decoded. This is due to the fact that some errors of multiplicity of 3, 5, 7, etc. are taken as single errors because the multiple errors syndrome may coincide with the *i-th* column of the matrix H^T, and some errors of multiplicity 4, 6, 8, etc. are not detected at all since their syndrome is equal to zero. Therefore, by exclusion of certain columns, one can increase the probability of correct decoding. For this, it is necessary that the shortened matrix H^T in each row should have the same number of ones if possible, and also in submatrix C, in each column there should be an odd number of ones. Accordingly, using the mentioned results as criteria to construct the optimal weight of columns in the proposed technique for enhancing the data reliability against soft errors, we obtain the following equation for the probability of miss-detecting four-bit errors:

$$P_4 = \frac{2B_4}{C_{2n}^4} = \frac{2B_4}{C_{2(m+2+[log_2 m])}^4} \tag{9}$$

3. RESULTS AND ANALYSIS

The evaluation of the efficiency of the proposed technique is done depending on the following three criteria:

- The number of bit errors (d) that can be detected and corrected, reflecting the fault tolerance capabilities of the technique and misdetection probability.
- The information rate m / n reflecting the amount of information redundancy added.
- The complexity of encoding and decoding schemes, reflecting the amount of hardware, software, and time redundancy added.

As mentioned, the proposed technique gives the ability of a system (computer, embedded system, etc.) to continue operating without interruption when one-, two-, or three-bit errors occur in the codeword. The following investigation shows the capabilities of the proposed technique to detect four-bit errors $(d = 4)$. A disadvantage of many of the SEC-DED codes proposed in the literature is the miss-correction and miss-detection of some bit errors, effectively reducing the reliability of these codes in a memory system or in data transmission. For example, Hamming code with $D = 4$ is an error-detecting and error correcting binary code that satisfies the equation:

$$2^k \geq m + k + 1 \tag{10}$$

where m is the number of data bits and k is the number of parity bits, depending on (1) and (2), such code can correct all single-bit errors and detect double-bit errors in any codeword. But in cases of errors occurring greater than two-bits (i.e $d > 2$) the result of decoding operations using Hamming code may provide an incorrect output. To decrease the probabilities of allowing wrong values get passed to the system output, the proposed technique uses the replication method and Hamming code by constructing its weight column depending on the criteria mentioned above. Suppose that $m = 16$; then, the parity check matrix is:

$$\mathbf{H} = \left|\begin{array}{c:c} \overset{\mathbf{C}}{00000000000111111111111111} & \overset{\mathbf{I}}{100000} \\ 00001111111000000011111111 & 010000 \\ 01110001111000111100001111 & 001000 \\ 10110110011011001100110011 & 000100 \\ 11011010101101010101010101 & 000010 \\ 11111111111111111111111111 & 111111 \end{array}\right|$$

If we replace the last row (containing all ones) with the sum of all ones from each column of the matrix H, then we get the following matrix H^T:

$$\mathbf{H^T} = \left|\begin{array}{c:c} \mathbf{C} & \mathbf{I} \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1 & 1\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,1\,1\,1\,1\,1\,1\,1\,0\,0\,0\,0\,0\,0\,0\,1\,1\,1\,1\,1\,1\,1\,1 & 0\,1\,0\,0\,0\,0 \\ 0\,1\,1\,1\,0\,0\,0\,1\,1\,1\,1\,0\,0\,0\,1\,1\,1\,1\,0\,0\,0\,0\,1\,1\,1\,1 & 0\,0\,1\,0\,0\,0 \\ 1\,0\,1\,1\,0\,1\,1\,0\,0\,1\,1\,0\,1\,1\,0\,0\,1\,1\,0\,0\,1\,1\,0\,0\,1\,1 & 0\,0\,0\,1\,0\,0 \\ 1\,1\,0\,1\,1\,0\,1\,0\,1\,0\,1\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1 & 0\,0\,0\,0\,1\,0 \\ 1\,1\,1\,0\,1\,1\,0\,1\,0\,0\,1\,1\,1\,0\,1\,0\,0\,1\,1\,0\,0\,1\,0\,1\,1\,0 & 0\,0\,0\,0\,0\,1 \end{array}\right|$$

Now shorten this matrix to transform the current code to (22, 16) code by removing the last 10 columns in submatrix C, we get the following matrix:

$$\mathbf{H^{(22,12)}} = \left|\begin{array}{c:c} \mathbf{C} & \mathbf{I} \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,1\,1\,1 & 1\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,1\,1\,1\,1\,1\,1\,1\,0\,0\,0\,0\,0 & 0\,1\,0\,0\,0\,0 \\ 0\,1\,1\,1\,0\,0\,0\,1\,1\,1\,1\,0\,0\,0\,1\,1 & 0\,0\,1\,0\,0\,0 \\ 1\,0\,1\,1\,0\,1\,1\,0\,0\,1\,1\,0\,1\,1\,0\,0 & 0\,0\,0\,1\,0\,0 \\ 1\,1\,0\,1\,1\,0\,1\,0\,1\,0\,1\,1\,0\,1\,0\,1 & 0\,0\,0\,0\,1\,0 \\ 1\,1\,1\,0\,1\,1\,0\,1\,0\,0\,1\,1\,1\,0\,1\,0 & 0\,0\,0\,0\,0\,1 \end{array}\right|$$

For the matrix $H^{(22,16)}$ we obtain the following results $B_4 = 263$, therefore, $P_3 = 0{,}683$ and $P_4 = 0{,}036$. Further, shorten the matrix $H^{(22,16)}$ according to the above criteria, i.e. so that each column of the submatrix C contains the same number of 1's (that is, 3 ones) and each row of the matrix contains the same number of 1's, then we obtain the $H^{Optimal}$ matrix:

$$\mathbf{H^{Optimal}} = \left|\begin{array}{c:c} \mathbf{C} & \mathbf{I} \\ 0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,1\,1\,1\,1\,1\,1 & 1\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,1\,1\,1\,1\,0\,0\,0\,0\,1\,1\,1\,1 & 0\,1\,0\,0\,0\,0 \\ 0\,1\,1\,1\,0\,0\,1\,1\,0\,0\,1\,1\,0\,0\,0\,1 & 0\,0\,1\,0\,0\,0 \\ 1\,0\,1\,1\,0\,1\,0\,1\,1\,1\,0\,0\,0\,0\,1\,0 & 0\,0\,0\,1\,0\,0 \\ 1\,1\,0\,1\,1\,1\,0\,0\,0\,1\,0\,1\,0\,1\,0\,0 & 0\,0\,0\,0\,1\,0 \\ 1\,1\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,0\,0 & 0\,0\,0\,0\,0\,1 \end{array}\right|$$

For the $H^{Optimal}$ matrix, we obtain the following results $B_4 = 250$, therefore, $P_3 = 0{,}649$ and $P_4 = 0{,}034$. The $\mathbf{H^{Optimal}}$ matrix has been implemented in the proposed technique. Suppose that sixteen data bits (m = 16) is protected against soft errors by using the proposed technique. Then the probability of miss-detecting four-bit errors when m = 16 is:

$$P_4 = \frac{2B_4}{C_{2n}^4} = \frac{2B_4}{C_{2(m+2+[log_2 m])}^4} = \frac{2 \cdot 250}{C_{44}^4} = \frac{500}{814506} = 0.0006.$$

These results as shown in Table 2 show the superiority of this technique in increasing reliability compared to other codes for the same decoding complexity.

Table 2. Performance evaluation with respect to time delay, false-correction probability and miss-detection probability

Code	Time Delay (ns)		False-Correction Probability		Misdetection Probability
	Coding	Decoding	$P(d = 2)$	$P(d = 3)$	$P(d = 4)$
SEC-DED	0.208	0.298	0.0	65.2	100
DAEC	0.230	0.325	64.0	65.2	100
BCH DEC	0.238	0.413	0.0	4.6	100
Proposed	0.227	0.340	0.0	0.0	0.0006

The encode delay equals the total time for calculating the check-bits for writing the data bits, while the decode delay is the time required to generate the syndromes S and parities P to correct the errors (when S ≠ 0 and P ≠ 0) on a read operation. The analysis scheme has been simplified for minimizing the decoding delay (consists of four inverters, five AND gates and two OR gates). The proposed technique adds no more than a 5% increase in decoder delay relative to the DAEC code (22, 16) and 14% comparing with SEC-DED (22, 16) Hamming code. The increase can be attributed to the analysis scheme required for the additional three adjacent bit error correcting syndromes. These results are shown in Table 2. Further,

the false-correction probabilities of three-bit errors for SEC-DED, DAEC, and BCH DEC codes range from 4.6 to 65.2, while the proposed technique shows zero probability of false-correction three-bit errors. With respect to four-bit errors, the mentioned codes cannot detect these errors, and the proposed technique may detect most cases ($P(d_t = 4) = 0.9994$) without affecting the performance. The information rate is represented by R as shown in Table 2 and it is given as,

$$R = \frac{m}{2n} = \frac{m}{2\cdot(m+2+[log_2 m])}. \tag{11}$$

It can be noted that the value of R for the proposed technique, when protecting large amount of data (big datawords), approaches 0.5 ($R \approx 0.5$). Table 3 compares the information rate, error detection, and correction capabilities of proposed technique with other ECCs.

Table 3. A comparison of codes with similarity decoding complexity in regard to information rate and the numbers of correctable and detectable errors

Code	Information Rate (R)	Number of Correctable Errors (d_c)	Number of Detectable Errors (d_t)
SEC-DED	0.77	1	2
DAEC	0.77	2	2
BCH DEC	0.62	2	2
Proposed	0.36	3	4

A memory simulation program was built to simulate the functionality of the proposed technique. We assumed that every codeword n contains 16 data bits and 6 check bits ($n = 22$). The program artificially inserts errors into the codeword. When the reading operation starts, the decoding function generates the values of syndromes S and parities P for the first and the second codewords. Depending on the Boolean (5), the simulation process confirms that all cases of three-bit error patterns in the data segment will not affect the data reliability. Consequently, the correct values get passed to the system output. In comparison with SEC-DED, DAEC and BCH DEC codes, the proposed technique shows relatively excellent multiple errors detection and correction capabilities, and at the same time, the structural simplification and the reduced gate delays in analysis schemes allows the processing of data at a faster rate and improved overall performance.

4. CONCLUSION

The impacts of technology scaling are leading to decreases in reliability against soft errors caused by cosmic radiation. For ensuring reliability under such scenarios, this paper presents a technique that provides high error-correction performance, high speed, and low complexity. The proposed technique ensures that only correct values get passed to the system output or are processed, in spite of the presence of up to three-bit errors ($d \leq 3$) in codewords, and provides a high probability of multiple-bit soft error detection (for $m = 16$, $P(d_t = 4) = 0.9994$). Such a solution has the potential to increase both protection capability and performance. The analysis of simulation results indicates that using a modified Hamming code (depending on above criteria), with a simple scheme of logic gates for generating control signals, can achieve good error detection and correction capabilities. At the same time, it can save a high-level performance value.

REFERENCES

[1] T. Heijmen, “Radiation Induced Soft Errors in Digital Circuits: A Literature Survey,” Philips Electronics Natl. Lab., Netherlands. Report number: 828, 2002.

[2] C. Mertens, B. Kress, M. Wiltberger, W. Tobiska, B. Grajewski, X. Xu, “Atmospheric ionizing radiation from galactic and solar cosmic rays,” Current Topics in Ionizing Radiation Research, InTech, pp. 683-738, 2012.

[3] M. Johnson-Groh, “NASA Studies Cosmic Radiation to Protect High-Altitude Travelers,” NASA's Goddard Space Flight Center, 2017. [Online]. Available: https://www.nasa.gov/feature/goddard/2017/nasa-studies-cosmic-radiation-to-protect-high-altitude-travelers/.

[4] E. Normand, “Single-Event Effects in Avionics,” *IEEE Transactions on Nuclear Science,* vol. 43, no. 2, pp. 461-474, 1996.

[5] T. Phillips, “Rads on a Plane: New Results,” 2018. [Online]. Available: https://spaceweatherarchive.com/2018/10/04/rads-on-a-plane-new-results/.

[6] P. Bradley, “The effects of cosmic radiation on implantable medical devices,” *Radiation'96 Conference handbook*, Sydney, vol. 56, 1996.

[7] L. Claro, A. Silva, J. Santos, S. Nogueira, A. Barrios, "Evaluation of soft errors rate in a commercial memory EEPROM," *2011 International Nuclear Atlantic Conference - INAC 2011,* 2011.
[8] G. Li, S. Hari, M. Sullivan, T. Tsai, K. Pattabiraman, J. Emer, S. Keckler, "Understanding error propagation in deep learning neural network (DNN) accelerators and applications," *SC '17 Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, Denver, 2017.
[9] R. Baumann, "Soft errors in advanced computer systems," *IEEE Design and Test of Computers,* vol. 22, no. 3, pp. 258– 266, 2005.
[10] I. Zaczyk, "Impact of cosmic radiation on aviation reliability and safety," *Journal of Applied Engineering Science*, vol. 11, no. 4, pp. 217-223, 2013.
[11] F. Shuguang, G. Shantanu, A. Ansari, S. Mahlke, "Shoestring: Probabilistic soft-error resilience on the cheap," *Proc. 15th Int. Conf. on Architectural Support for Programming Languages and Operating Systems. ACM SIGPLAN Notices,* Pittsburgh, 2010.
[12] H. Wang, K. Zhao, M. Lv, X. Zhang, H. Sun, T. Zhang, "Improving 3D DRAM Fault Tolerance Through Weak Cell Aware Error Correction," *IEEE Transactions on Computers*, vol. 66, no. 5, pp. 820–833, 2017.
[13] K. Zhang, Y. Manzawa, K. Kobayashi, "Impact of body bias on soft error tolerance of bulk and Silicon on Thin BOX structure in 65-nm process," *2014 IEEE International Reliability Physics Symposium*, Waikoloa, 2014.
[14] D. Kobayashi, K. Hirose, T. Ito, Y. Kakehashi, O. Kawasaki, T. Makino, T. Ohshima, D. Matsuura, T. Narita, M. Kato, S. Ishii, K. Masukawa, "Heavy-Ion Soft Errors in Back-Biased Thin-BOX SOI SRAMs: Hundredfold Sensitivity Due to Line-Type Multicell Upsets," *IEEE Transactions on Nuclear Science*, vol. 65, no. 1, pp. 523-532, 2018.
[15] K. Zhang, S. Umehara, J. Yamaguchi, J. Furuta, K. Kobayashi, "Analysis of soft error rates in 65- and 28-nm FD-SOI processes depending on BOX region thickness and body bias by Monte-Carlo based simulations," *IEEE Transactions on Nuclear Science*, vol. 63, no. 4, pp. 2002–2009, 2016.
[16] W. Zhang, "Replication Cache: A Small Fully Associative Cache to Improve Data Cache Reliability," *IEEE Transactions on Computers,* vol. 54, no. 12, pp. 1547-1555, 2005.
[17] N. Oh, P. Shirvani, E. McCluskey, "Error detection by duplicated instructions in super-scalar processors," *IEEE Transactions on Reliability*, vol. 51, no. 1, pp. 63-75, 2002.
[18] W. Zhang, S. Gurumurthi, M. Kandemir, and A. Sivasubramaniam, "ICR: In-cache replication for enhancing data cache reliability," *2003 International Conference on Dependable Systems and Networks*, San Francisco, 2003.
[19] J. Hu, S. Wang, S. Ziavras, "On the exploitation of narrow-width values for improving register file reliability," *IEEE Transactions on Very Large-Scale Integration (VLSI) Systems*, vol. 17, no. 7, pp. 953– 963, 2009.
[20] R. W. Hamming, "Error Correcting and Error Detecting Codes," *Bell System Technical Journal*, vol. 29, no. 2, pp. 147-160, 1950.
[21] M. Hsiao, "A Class of Optimal Minimum Odd-weight column SEC-DED codes," *IBM Journal of Research and Development*, vol. 14, no. 4, pp. 395-401, 1970.
[22] A. Kazeminejad, "Fast, Minimal Decoding Complexity, Systematic (13, 8) Single-error-correcting codes for On-chip DRAM applications," *IEEE Electronics Letters*, vol. 37, no. 7, pp. 438-440, 2001.
[23] A. Abdullah Hamdoon, Zaid Ghanim Mohammed, Emad A. Mohammed, "Design and implementation of single bit error correction linear block code system based on FPGA," *TELKOMNIKA Telecommunication Computing Electronics and Control*," vol. 17, no. 4, pp. 1785-1795, 2019.
[24] M. Demirci, P. Reviriego, J. Maestro, "Unequal error protection codes derived from double error correction orthogonal Latin square codes," *IEEE Transactions on Computers*, vol. 65, no. 9, pp. 2932–2938, 2016.
[25] A. Dutta, N. Touba, "Multiple Bit Upset Tolerant Memory Using a Selective Cycle Avoidance Based SEC-DED-DAEC Code," *25th IEEE VLSI Test Symmposium*, pp. 349–354, 2007.
[26] G. Nguyen, "Error-Detection Codes: Algorithms and Fast Implementation," *IEEE Transactions on Computers*, vol. 54, no. 1, pp. 1– 11, 2005.