Hardware Implementation of Single Bit Error Correction and Double Bit Error Detection through Selective Bit Placement for Memory

Lankesh
M. Tech student,
Dept. of Telecommunication Engineering,
Siddaganga Institute of Technology,
Tumkur, Karnataka, India-572103

K.C.Narasimhamurthy, PhD
Professor,
Dept. of Telecommunication Engineering,
Siddaganga Institute of Technology,
Tumkur, Karnataka, India-572103

ABSTRACT
Hamming codes are widely used for the single bit error correction double bit error detection (SEC-DED) which occurred during data transmission process. This paper presents an enhanced detection of double adjacent bit errors and correcting all possible single bit errors in Hamming codes through selective bit placement technique for memory application. Soft errors occur due to the radiation particles which affects the memory elements. These soft errors cause flipping of single bit or more often adjacent bits in the memory. Proposed technique improves the probability of detecting double adjacent bit errors and also provides a simple method of detecting double adjacent bit errors as compared to convolution coding through interleaving which is complex and requires higher memory. Hamming (12, 8) is implemented on hardware using RL 78 (G13) Microcontroller.

Keywords
Hamming code, SEC-DED, soft error.

1. INTRODUCTION
Hamming code is most widely used for the detection and correction of single bit error till today even though it was introduced long ago in the year 1950 [1]. One of its applications includes mitigation of soft errors in memory devices. Whenever a radiation particle hits the memory device, if the logical value of stored data is changed then it is referred as soft error. In this case only the stored data is flipped which can be replaced by correct data. There exists one more type of error called "hard error", in which the hardware circuitry is damaged permanently and there is no possibility of retrieving back the original data. Single event upset (SEU) is the most commonly used word for soft error, in which only single bit of the data is affected, if the energy of the radiation particle is high enough to flip more than one data bits then it is called as multiple bit upset (MBU).

The major sources for soft errors are as follows a) Alpha particles and b) Cosmic rays [2, 3]. Alpha particles emitted by traces of uranium or thorium impurities in packaging materials are emitted as a result whenever the nucleus of an unstable isotope decays to a lower energy state [4]. As compared to the other radio- active isotopes, uranium and thorium possesses the maximum activity among the naturally occurring materials. Radio-active impurities contributes as the major sources of alpha particle emission which includes lead based isotopes in solder bumps of the flip-chip technology, aluminum in ceramic packages, lead based alloys and inter connect metallization.

Cosmic rays are another major source for the soft errors. In order to mitigate the effect of soft errors, prevention and recovery techniques are used. Error detection and correction (EDAC) is one of recovery technique used for the mitigation of soft errors in memory devices. Hamming code can correct all the single bit errors without any miscreation, but if there exists more than one bit error then Hamming code fails. For the SCE-DED extended Hamming code or shortened Hamming code is used by including an additional parity bit to the Hamming code output, through which double bit errors can also be detected but not corrected. The proposed scheme increases the probability of detecting double and triples adjacent bit errors using the Hamming code as compared to the conventional method. The rest of the paper is organized as follows, section 2 contains brief explanation of Hamming code, details of double bit error detection is provided in section 3, section 4 contains the proposed selective bit placement strategy, in section 5 simulation results are discussed, hardware implementation details are given in section 6 and conclusion and future scope is given section 7.

2. HAMMING CODE
Most commonly used channel encoding techniques are block code and convolution code. Construction of block code is simple as compared to convolution code. Hamming code is one of the linear block codes, and is capable of correcting single bit errors [5]. It converts a k-bit message into n-bit code word. For any positive integer $m \geq 3$,

$$n = 2^m - 1 \quad (1)$$

$$k = n - m \quad (2)$$

Where 'n' is the length of the code word, 'k' is the length of the message bits and m is the number of parity check bits to be added to the message bit.

The error correcting capacity mainly depends on the parameter known as Hamming distance (d_{min}), it indicates the number of positions by which two equal length code words differ. If the minimum distance between any two code words is ≥ 3, then such code can correct all single bit errors and if the $d_{min} \geq 4$, then it can correct the single bit errors and also can detect possible double bit errors, for example C1 = (1 0 1 0 1 01) and C2 = (1 1 0 0 1 0 0) then d_{min} (C1, C2) = 3. Encoding and decoding is done using the generator matrix and parity check matrices respectively. Table 1 varying length of codeword(n) for different length of message bits(k) of Hamming code.

Table 1. Hamming code parameters

k	n
4	7
11	15
26	31
57	63

Encoding can be done using systematic and non-systematic way, which depends on the type of generator matrix used and same applies for decoding purpose. Generator matrix and parity check matrix for (7, 4) (i. e. n, k) Hamming code in systematic form is as shown below

$$G = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (5)$$

$$H = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix} \quad (6)$$

The message 'u' is multiplied with the generator matrix to obtain the code word C, let u = (1 0 1 0) is multiplied with the generator matrix G then the code word obtained is C = (0 0 1 1 0 1 0). In the code word first three bits are parity check bits followed by four message bits, hence it is in systematic form. At the receiver side received code word is multiplied with the transpose of the parity check matrix (H^T) for the decoding purpose and the resulting vector is called as syndrome.

$$v = r \times H^T \quad (5)$$

Suppose fifth bit was flipped during the transmission and the received vector is r = (0 0 1 1 1 1 0) then the syndrome obtained is v = (0 1 1). Position of the syndrome is sorted in parity check matrix, (0 1 1) present in 5^{th} column, hence e = (0 0 0 0 1 0 0) is added (modulo 2 addition) to 'v' to obtain the correct code word, which has 1 at fifth position. Hence the erroneous data is corrected, if the syndrome is a null vector then it indicates received code word is correct and no error has occurred. For the double bit error detection shortened Hamming code is used which is also known as SEC-DED, in which overall parity bit is padded to the code word, hence (7,4) code word now becomes (8,4) with the extra parity bit.

Another approach of generating Hamming code is as follows: i) all the bit positions are numbered from 1 to 7(for (7, 4) code word), ii) position numbers are written in binary form (000, 010, 011, 100, etc.), iii) check bits are placed in 2^r bit positions. Let p1, p2, and p3 are the check bits d1, d2, d3, d4 are the data bits [6].

Table 2. Algorithm for Generation of (7, 4) Hamming Code

Position	c1	c2	c3	c4	c5	c6	c7
Binary	001	010	011	100	101	110	111
Content	p1	p2	d1	p3	d2	d3	d4
c1(p1)			0**1**1		1**0**1		1**1**1
c2(p2)			**0**11			1**1**0	1**1**1
C4(p3)					**1**01	**1**10	**1**11

Calculation of parity bits done as given below:

c1 =c3 + c5 + c7 or p1 = d1 + d2 + d4

c2 =c3 + c6 + c7 or p2 = d1 + d3 + d4

c4 =c5 + c6 + c7 or p3 = d2 + d3 + d4, i.e. c1 (or p1) is calculated by ex-or operation on the data bits whose binary representation of the bit position contains 1 as LSB (which is highlighted in bold in Table 2), hence p1 is calculated by doing ex-or operation on the data bits which are in position no. 3, 5, 7(d1, d2 and d4), c2(or p2) is calculated by ex-or operation on those data bits whose positions have a 1 in their second from least significant binary position number(3, 6, 7), procedure is repeated to calculate c3(or p3) but on those data bits whose positions have a 1 in their third from least significant binary position number(5, 6, 7). This approach is known as non-canonical representation of generator matrix. These parity bits are used for the detection and correction of errors. Minimum number of check bits to be included in the code word is depends on the following formula

$$2^m \geq m + k + 1 \quad (6)$$

Table 3 shows the minimum number of check bits to be included for different length of message bits.

Table 3. Check Bits for Error Correction/Detection

Number of information bits k	m for SEC	m for SEC-DED
1	2	3
2 to 4	3	4
5 to 11	4	5
12 to 26	5	6
27 to 57	6	7
58 to 120	7	8
121 to 247	8	9
248 to 502	9	10

The proposed technique consists of a special parity check matrix called as lexicography matrix [7]. Each column j of the matrix contains the binary representation of j. The shortened Hamming code lexicography matrix for code length 12 bits is as shown below

$$H_{Lex} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{pmatrix} (7)$$

If there is a single bit error in the code word, when it is multiplied with the transpose of Lexicographic matrix (H_{Lex}) it will result in the binary representation of the position of the bit. For example if the message bits are (01010100) then it will be encoded as (000010110100) using (12, 8) Hamming code, suppose error is introduced in the fifth bit then the encoded data will change to (000000110100), when this erroneous encoded data is multiplied with the $(H_{Lex})^T$ it will result the syndrome as (0101) which is the binary representation of position five, which is present at the fifth column of Lexicographic check matrix, hence the single bit

error is corrected. Suppose there are two bits in error at position 3 and 4, then the encoded data will become (001110110100) when it is multiplied with $(H_{Lex})^T$ it will result in a syndrome (0111), which assumes that there is a single bit error at position 7 and the code word would be corrected into (001110010100) instead of the right word.

Extended Hamming code is used to detect the double bit errors, an overall parity bit is padded to the code word which is responsible for the detection of double bit error in the code word, as given in Table 4. By padding the overall parity bit all double bit errors can be detected, which was miscorrected earlier. The errors at position 3 and 4 generate a nonzero syndrome with no error in the overall parity bit, hence a double bit error is detected but not corrected.

Table 4. Adding a parity bit to make a SEC-DED code

Possibilities			Receiver Conclusion
Errors	Overall parity	Syndrome	
0	Even	0	No error.
1	Odd	0	Overall parity bit is in error.
		≠ 0	Syndrome indicates the bit in error.
2	Even	≠ 0	Double error (not correctable).

3. DOUBLE BIT ERROR DETECTION

The lexicographic check matrix consists of 12 columns as given in Eq. 7 and each column is represented in the binary form starting from number 1 to 12. It can be observed that 13, 14 and 15 are missing in the matrix (as 15 is the maximum value in 4 bit binary representation). If any syndrome value exceeds 12(or 1100 in binary) then it indicates there is at least one double bit error, hence 13, 14 and 15 are reserved for the double bit error detection, which cannot be corrected as the syndrome does not match with any of the column in the lexicographic check matrix. By using extended Hamming code (padding overall parity bit) all double bit errors can be detected and also increases the probability of detecting triple bit errors. The proposed algorithm improves the probability of detecting double bit error without using overall parity bit, which is discussed in section 4.

4. SELECTIVE BIT PLACEMENT TECHNIQUE

The main aim of this technique is to maximize the detection of double and triple bit errors using Hamming and parity extended Hamming code respectively by rearranging the bits randomly. The procedure is as follows:

- All possible double and triple bit errors are calculated using a software program, such that the resulting syndrome doesn't match with any of that is resulted for single bit errors.
- Words are reordered manually in order to maximize the detection of adjacent bit errors.

The complete procedure is as shown in Fig. 1.

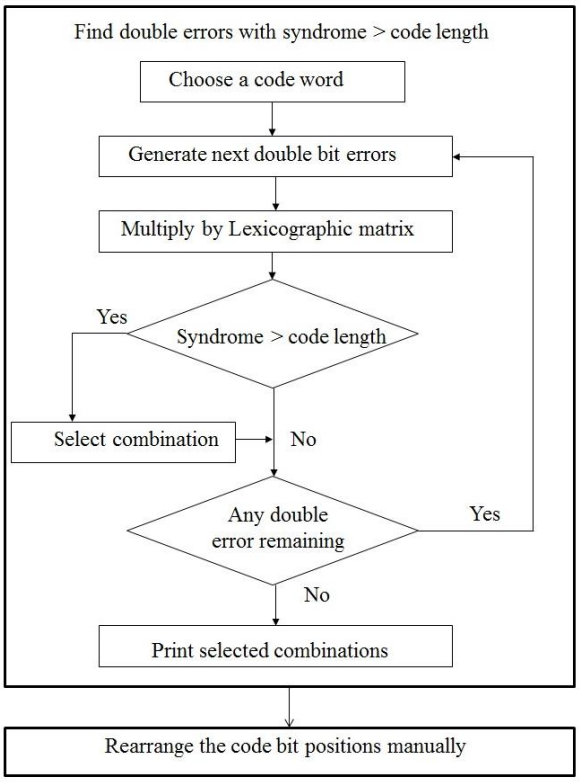

Fig.1: Selective bit placement technique procedure.

Using this approach, with the help of software program all possible combinations of two bit errors are obtained for the message bits (01010100) and they are as follows: 1–12, 2–12, 3–12, 4–9, 4–10, 4–11, 5–8, 5–10, 5–11, 6–8, 6–9, 6–11, 7–8, 7–9 and 7–10[7]. The only adjacent combination is 7–8, hence only 1 out of 11 adjacent pairs is correctly detected. By using selective bit placement 9 out of 11 combinations are detected correctly, which is tabulated in Table 5.

Table 5. Double adjacent error detection for Hamming (12, 8)

Bit placement												Detection probability	
1	2	3	4	5	6	7	8	9	10	11	12	1/11	0.09
1	12	2	3	6	8	7	9	4	10	5	11	9/11	0.82

In the proposed work same procedure is carried out for 64 bit message, the suggested bit position for 64 bit message is proposed in Table 6.

Table 6. New bit positions for Hamming (71, 64)

Code	New position Order
Hamming (71, 64)	{8}{64}{9}{65}{10}{66}{11}{67}{12}{68}{13} {69}{14}{70}{15}{71}{1}{63}{2}{62}{3}{61} {4}{60}{5}{59}{6}{58}{7}{57}{16}{56}{17} {55}{18}{54}{19}{53}{20}{52}{21}{51}{22} {50}{23}{49}{24}{48}{25}{47}{26}{46}{27} {45}{28}{44}{29}{43}{30}{42}{31}{41}{32} {40}{33}{39} {34}{38}{35}{37}{36}

5. SIMULATION RESULTS

The proposed method is same as that of discussed for Hamming (12, 8) but it is extended up to 64 bits which consists of 71 columns. By trial and error all possible combinations of double bit errors are calculated, and the bit positions which produces the syndrome greater than (1000111) (binary equivalent of 71) are placed adjacently as mentioned in Table 6.

Simulation is carried out using Matlab for the message bits (0101010001010100010101000101010001010100010101000 101010001010100). If there is no error then the encoded data (1000101101000100101000101010001101010001010100010 1010001010100011010100) is multiplied with the Lexicographic check matrix and the syndrome obtained is (0000000), hence it displays as "No error", as shown in fig. 1(line no. 8 in fig.2), and if there are two adjacent bit errors which was miscorrected previously, is now detected after selective bit placement (line no. 20 in Fig. 2). Suppose error occurs at 25th position then encoded data with error is (1000101101000100101000100100011010100010101000010 1010001010100011010100). When this erroneous encoded data is multiplied with Lexicographic check matrix, the resulting syndrome is (0011001) which is present at 25th column of Lexicographic check matrix, and displays as "Error is in bit position: 25" as shown in Fig. 3(line no. 7 in fig. 3) and the erroneous bit is corrected. Suppose there exists double adjacent bit errors at positions 25 and 26 of encoded data then the conventional method will fails, which detects as "Error is in bit position : 3" instead detecting double bit errors and miscorrect it(as indicated in line no.10 to line no. 14 in Fig. 3). Miscorrection of double adjacent bit errors can be avoided using selective bit placement technique, which detects as "There is a double bit error" (line no. 20 in Fig. 3). Result is tabulated in Table 7.

Table 7. Double adjacent bit error detection for Hamming (71, 64)

Code	Number of parity bits used	Probability of double adjacent bit errors			
		Original sequence		Selective bit placement	
Hamming (71,64)	7	1/63	0.01	15/73	0.21

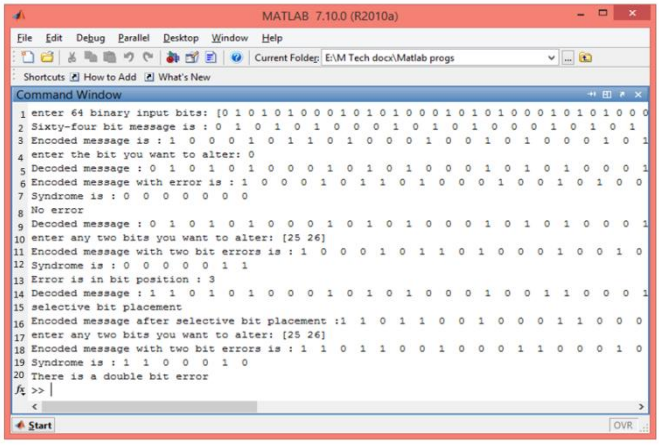

Fig.2: Simulation result for no error and double adjacent bit errors.

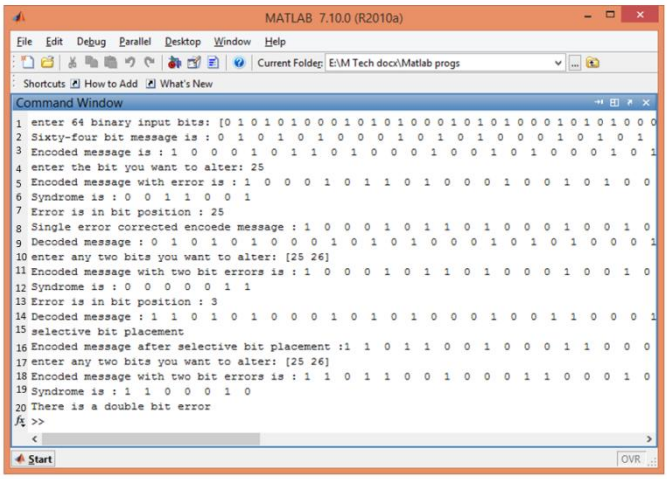

Fig.3: Simulation result for one bit error correction and double adjacent bit error detection.

6. HARDWARE IMPLEMENTATION

Hardware implementation is done using RL 78(R5F100LEA) Microcontroller. (Specifications of the microcontroller is as follows, operating frequency 16 MHz, ROM 512 KB, RAM 32KB, Data flash of 8 KB) [8]. Simulation of Hamming (12, 8) is carried out using C coding through CodeBlocks. UART serial mode of communication is done between the personal computer and the microcontroller for the error detection and correction purpose using E1 emulator as a debugging tool. Result of Hamming (12, 8) is as shown in Fig. 4. Result from the microcontroller is obtained using Terminal v1.39b. Eight bit message (10101010) is encoded as (111101001010), suppose error occurs at 5th position encoded data changes to (111111001010), after multiplication with Lexicographic

check matrix, syndrome obtained is 0101, single bit error will be detected and displayed as "Error is in bit position : 5", as shown in Fig. 4 and the error is corrected.

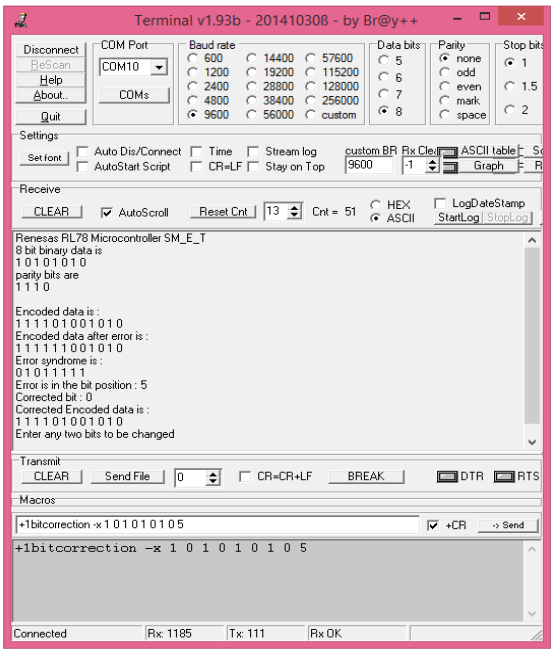

Fig.4: Single bit error correction for Hamming (12, 8).

Suppose there are two bits in error at positions 1 and 2 respectively, the erroneous encoded data is (011101001010), and then conventional Hamming code fails to detect the double bit error and miscorrect it as 3rd bit is in error. But the proposed method detects it as "There is a double bit error", and miscorrection is avoided, as shown in Fig. 5.

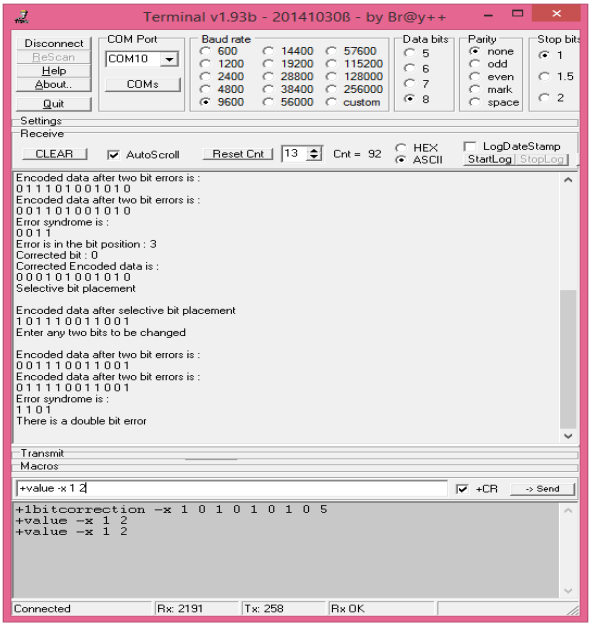

Fig.4: Double bit error detection through selective bit placement for Hamming (12, 8).

7. CONCLUSION AND FUTURE SCOPE

In this paper a strategy which is used to maximize the probability detecting double adjacent bit errors is presented for different length of message bits. The strategy used is selective bit placement, which provides improved result for double adjacent bit error detection as compared to conventional Hamming code with lesser number of check bits. Results are obtained for different length of message bits using Matlab software and compared with the conventional Hamming code. Further, work presented in [7] is extended to 64 bit message sequence, which was limited to 32 bit message sequence. Hamming (12, 8) is implemented on hardware using RL 78 (G13) Microcontroller. In future, the proposed strategy can be enhanced for the detection and correction of double and triple adjacent bit errors, with less number of check bits.

8. REFERENCES

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

[2] R. C. Baumann, "Radiation-induced soft errors in advanced semiconductor technologies," *IEEE Trans. Device Mater. Rel.*, vol. 5, no. 3, pp. 301–316, Sep. 2005

[3] R. C. Baumann, "Soft errors in advanced semiconductor devices - part I: the three radiation sources," *IEEE Trans. Nucl. Sci.*, vol. 1, no. 1, pp. 17–22, Mar. 2001.

[4] L. Lantz, "Soft errors induced by alpha particles," *IEEE Trans. Reliab.*, vol. 45, no. 2, pp. 174–179, Dec. 1996.

[5] S. Lin and D. J. Costello, *Error Control Coding*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 2004.

[6] P C Gupta, *Data communication and networking*, 2nd edition PHI Learning Pvt. Ltd., 02-Nov-2013

[7] Alfonso Sanchez - Macian "Enhanced detection of double and triple adjacent errors in hamming codes through selective bit placement" *IEEE Transactions on device and materials*, vol. 12, no. 2, pp. 357– 362, Jun. 2012.

[8] RL78-Microcontroller http://documentation.renesas.com/doc/products/mpumcu/doc/rl78/r01ds0131ej0310_rl78g13.pdf