ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 8, August 2015

Modified Hamming Codes with Double Adjacent Error Correction along with Enhanced Adjacent Error Detection

Babitha Antony, Divya S

Dept. of Electronics and Communication, Sree Narayana Gurukulam College of Engineering, Ernakulam, India

Assistant Professor, Department of ECE, Sree Narayana Gurukulam College of Engineering, Kadayiruppu, Kerala, India

ABSTRACT: Error correction codes are used in semiconductor memories to protect information against soft errors. Soft error is major concern for memory reliability especially for memories that are used in space applications. In such cases simple error correction codes are preferred due to their simple encoding/decoding logic, low redundancy and low encoding/decoding latency. Hamming codes are attractive as they are simple to construct for any word length and the encoding/decoding can be done with low delay. But they only allow single error correction or double error detection, so a multiple error can lead to a wrong decoding. Nowadays multiple errors are becoming more frequent as integration scale increases. Multiple errorsoccurs mainly to adjacent bits. In this paper modified Hamming codes to enhance adjacent error detection along with double adjacent error correction is presented. The enhanced detection of adjacent error can be achieved by modifying the Hamming matrices. The double adjacent error correction can be achieved by adding few redundant bits. The modified Hamming codes can correct single & double adjacent errors and can detect double errors & triple adjacent errors. The double and triple adjacent errors are precisely the types of errors that an MCU would likely cause, and therefore, the modified Hamming codes will be useful to provide error detection and correction for MCUs in memory designs.

KEYWORDS: Hamming codes;Multiple cell upsets; enhanced adjacent error detection; double adjacent error correction; Hamming matrix

I. Introduction

The memories play an important role in the semiconductor market because the system-on-chip market is booming and almost every system chip contains some type of embedded memory. The memories will be reliable only when the data stored in it is error free. If there is error then the memory will be unreliable and sometimes this can make the whole system unreliable. For memories that are used in radiation environments especially those that are used in space applications the main cause of error is the radiation induced soft error [1]. The radiation induced soft error occurs when a radiation particle hits the device and changes the logic value. Error Correction Codes (ECCs) are used to prevent soft errors from causing data corruption in memories and registers [2], [3]. The codes used range from simple codes such as Hamming codes [4] to more powerful and complex codes like Bose–Chaudhuri–Hocquenghem (BCH) codes, matrix codes and Euclidean Geometry (EG) codes [5], [6], [7]. In all cases, the data is encoded when it is written into the memory and decoded when it is read. Therefore, the encoding and decoding latency directly impact the memory access time. To minimize this effect ECCsfor which decoding is simple are used in most cases. Oneexample of codes for which decoding can be done with lowdelay is Single Error Correction (SEC) codes. Hamming codes are Single error correction codes. Hamming codes are attractive as they are simple to construct for any word length and theencoding and decoding can be done with low delay. SEC codes havea minimum distance of three and therefore a double error canbe mistaken for a single error and erroneously corrected. Toavoid this issue Single Error Correction Double Error Detection(SEC-DED) codes are preferred in memory applications[2], these codes have a minimum distance of four. Hamming codes can be extended with a parity bitcovering all bits to implement a SEC-DED code [4]. Therefore,they are suitable for memory applications and also to protectregisters in digital circuits.

A SEC-DED code [4] is capable of correcting one error and detecting all possible double errors. It is commonly used in memories and caches, but cannot correct more than a 1-bit error in a word. As technology scales, it is more likely that a radiation particle upsets more than one memory cell or register causing multiple errors [9]. This is known as a Multiple Cell Upset [10]. Recent studies characterizing different bit errors arising from an SEU suggest that 1–5% of the SEUs can cause multiple bit upsets (MBUs) [8]. The cells affected by the MCU are physically close and in many cases adjacent [11]. This is because errors are created along the path that the particle traverses. MCUs can therefore cause multiple errors on a given word causing a failure even when a SEC-DED code is used.

In order to correct the most commonly occurring MBUs, this paper proposes a low cost ECC methodology to correct double adjacent bit errors and to detect triple adjacent errors. It involves constructing a single-error-correcting, double-error-detecting, double adjacent-error-correcting, triple adjacent error detecting (SEC-DED-DAEC-TAED) code by selectively avoiding certain types of linear dependencies in the parity check matrix and by selectively shortening & reordering the Hamming matrices. A key feature of the proposed SEC-DED-DAEC- TAED code is that it uses only 6 more additional check bits as the conventional SEC-DED codes and they have nearly identical timing overhead as the conventional SEC-DED codes.

The rest of the paper is organized as follows section 2 presents related work. In section 3 an overview of Hamming codes is presented. Section 4 describes the enhanced adjacent error detection. Section 5 describes the double adjacent error correction. Section 6 explains about the proposed code and in Section 7 the simulation results are presented. Section 8 provides the conclusion of the paper.

II. Related work

A number of approaches for extending the basic SEC-DED Hamming code [4] have been previously proposed. A special class of SEC-DED codes known as Hsiao codes [12] was proposed to improve the speed, cost, and reliability of the decoding logic. The codes constructed in the proposed methodology can be thought of as a special class of Hsiao codes. Another class of SEC-DED codes [13], [14] was proposed to detect any number of errors affecting a single byte. These codes are known as single-error-correcting double-error-detecting single-byte-error-detecting (SEC-DED- SBD) codes. For protecting byte-organized memories, SEC-DED-SBD codes are more suitable than the conventional SEC-DED code.

To provide complete double error correction capability, a double-error-correcting triple-error-detecting (DEC-TED) code may be used at the cost of much larger overhead in terms of both the check bits and more complex hardware to implement the error correction and detection [15], [16], [17].

The Reed-Solomon (RS) code and Bose-Chaudhuri- Hocquenghem (BCH) codes are able to detect and correct multiple bytes of errors with very low overhead in terms of additional check bits required. However, these codes typically work at the block level and are applied to multiple words at a time. The general drawbacks with these methods are latency and speed. Most of these codes require several cycles to correct the first error. Moreover, the encoding and decoding are much more complex and require several table lookups for multiplication in higher order fields.

Another class of multiple error-correcting approaches combines coding with circuit level techniques to sense multiple errors in a memory. In [18] and [19], an asynchronous built in current sensor (BICS) on the vertical power lines of a memory along with a parity bit per memory word is used. A conventional SEC-DED code and the BICS approach are combined in [20] to detect multiple bit upsets affecting the same memory word.

Even though several powerful error correcting codes exist, the SEC-DED code has remained an attractive choice mainly because of its fast and simple encoding/ decoding and low hardware overhead. One of the most commonly used techniques to minimize the probability of multiple bit upsets in a single word is bit interleaving which is a memory layout architecture in which physically adjacent bits are assigned to different logical words. For k-way interleaving, k adjacent failing bits appear as k single bit errors in k different logical words rather than as a k-bit error in a single logical word. A simple SEC-DED code can be used along with bit interleaving to help protect from multiple bit upsets. However, there can be some limitations/drawbacks for bit interleaving. In some cases, it may negatively impact floor

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 8, August 2015

planning, access time, and/or power consumption. The proposed SEC-DED-DAEC-TAED code requires very little overhead and can be used instead of or in addition to bit interleaving to provide greater flexibility for optimizing a memory design.

The Selective bit placement strategy is proposed in [21] to maximize the probability of detecting adjacent errors in Hamming codes. The enhanced detection is achieved by selectively placing the bits in the memory such that adjacent errors produce a syndrome that does not match any of those that correspond to a single error. In order to further enhance the adjacent error detection capability the Selective shortening and reordering strategy is proposed in [22]. The enhanced detection of adjacent errors is achieved by reordering the Hamming matrices.

The ECC methodology proposed in this paper constructs a different SEC-DED-DAEC-TAED code from the ones described in [22] and [23]. The proposed SEC-DED-DAEC-TAED codes are targeted for memories that are used in space applications. They have additional six check bits and nearly identical encoding and decoding latency as conventional SEC-DED codes. The proposed codes are constructed by selectively avoiding certain type of cycles in the parity check matrix and by selectively shortening &reodering the Hamming matrices.

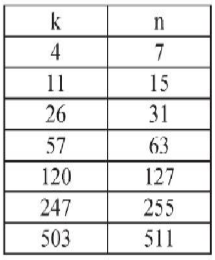
III. HAMMING CODES

Hamming codes are linear block error-correcting codes that were proposed by R.W. Hamming [4]. They provide single error correction or double error detection. For any positive integer m ⩾ 3 they have the following parameters :

$n = 2^m - 1;\ k = n - m;\ d_{min} = 3$ (1)

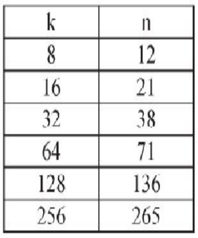
where n is the block size, m the parity check bits, k the number of information bits and d_{min} the minimum distance of the code. The parity bits are organized in a special way so different incorrect bits produce different error results when decoding. Some possible values of the parameters are illustrated in Table I. For memory applications, the number of information bits k is commonly a power of two and Hamming codes are shortened to fit that word length as illustrated in Table II.

Table I
Hamming Codes Parameters



k	n
4	7
11	15
26	31
57	63
120	127
247	255
503	511

Table II
Shortened Hamming Codes Parameters



k	n
8	12
16	21
32	38
64	71
128	136
256	265

Hamming codes are linear codes and, consequently, can be generated and decoded using the generator and parity-check matrices, respectively.An algorithm to generate Hamming code words from information bits is as follows: a) positions are numbered from 1 to n; b) positions are written in their binary form (1, 10, 11, etc.); c) bits in positions 2r are parity bits for those other positions where the binary form of those positions has the bit r + 1 set to one.
For instance, in the Hamming code (7,4) with n = 7, k = 4 and m = 3, positions c1, c2 and c4 are parity bits (p1, p2 and p3) and the information bits (d1, d2, d3 and d4) are placed in order in the rest of positions as shown in Table III.

Table III
Generation algorithm for Hamming Code (7,4)

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)	--		011		101		111
c2 (p2)		--	011			110	111
c3 (p3)				--	101	110	111

Parity bits are calculated as follows:

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

As an example, with this scheme the code word for data bits (1 0 1 0) is (1 0 1 1 0 1 0). In order to check the code word, the parity bits can be recalculated again from the information bits and compared to the original set of parity bits. If they match, then no error was introduced (or it is not detected), otherwise, an error is detected and the non-matching parity bits can provide us with the information of the bit that was flipped so that the error can be corrected.
From a linear algebra approach, the parity-check matrix can be used to detect an error. The product of this matrix by the current value of the code word results in a vector called syndrome. If this vector is the null vector, then the current value of the word is an actual code word. In any other case, an error occurred in the code word.

If a single bit error occurs in the code word, the syndrome vector that results from the product of the parity-check-matrix with the error code word gives the binary representation of the position where the error was inserted. A Hamming code can be used to correct single errors or, alternatively, to detect single and double errors. As the minimum distance between two words is three, it is not possible to distinguish between single and double errors. An option is to use an extended version of the Hamming code that includes an additional parity bit that covers all the bits in the code word. This solution increases the minimum distance to four and allows performing single error correction and double error detection (SEC-DED) simultaneously. Alternatively, it can be used to detect triple errors. With a minimum distance of 4, a triple error in the extended Hamming code can be miscorrected.

IV. ENHANCED ADJACENT ERROR DETECTION

The enhanced detection of adjacent errors is achieved by performing a selective shortening and reordering of the Hamming matrix so adjacent errors result in a syndrome that does not match that of any single error.
For a 16-bit data word (k = 16), shortening is applied to a (31, 26) Hamming code, producing a (21, 16) SEC code. Thus, 10 columns can be removed from the original matrix. The shortening process chosen for this paper is as follows:
- Fill the first 16 columns with the odd-weight values to maximize double-error detection. A double error affecting any of these columns will produce an even-weight syndrome. So, it will not correspond to any of these columns.

- Sort those columns trying to minimize the different even-weight syndromes generated. Adjacent errors on these 16 columns produce 15 syndromes. The goal is to maximize the coincidences between these syndrome values.
- The remaining 5 columns need to be filled by even-weight values. However, an adjacent error produced in the transition between the last odd-weight column and the first even-weight value would produce a miscorrection as it corresponds to a different existing odd-weight column. So a specific odd-weight column will be selected and removed from the matrix to provide for the identified odd-weight syndrome.
- The 5 columns plus the removed one (6 in total) are filled with even-weight values placing them in the appropriate order and excluding those which coincide with a previous double adjacent error syndrome.

Using the procedure described above, the following matrix (2) is generated for Hamming (21,16) code. The parity bit columns (those that have a single one) are emphasized using bold face.

$$\begin{pmatrix} 0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&0&0&0&1\\ 1&1&1&1&1&1&0&0&1&0&0&0&0&0&0&1&1&0&1&0&1\\ 1&1&1&0&0&0&1&1&0&1&0&0&0&0&1&0&0&1&0&1&0\\ 1&0&0&1&1&0&1&1&0&0&1&0&0&1&0&1&1&1&0&0&0\\ 0&0&1&0&1&1&1&0&0&1&0&0&1&1&0&1&0&0&1&1&0 \end{pmatrix} \quad (2)$$

A similar approach can be applied to Extended Hamming Codes adding the Triple Adjacent Error detectioncapability. Extending Hamming codes include an additional parity bit over the whole word. If we consider initially only the original Hamming matrix, selective shortening is, this time, performed by taking into account the weight properties of columns being added. The objective is to get an even-weight syndrome for any triple adjacent error. To fulfill this goal, columns are arranged placing them with the following weight order (odd odd even odd odd even . . . and so on).

An algorithm is designed to place the smallest available column value with the expected weight for a specific position. It excludes the previously selected columns and, if the expected weight is even, it also ignores those values that result from the combination of any previous consecutive three columns.Theresult is a set of matrices where any three adjacent columns produce a value not included in the same matrix, so a triple adjacent error is detected. The matrices for (22,16) code, which do not consider the parity bit yet, are shown in (3).

$$\begin{matrix}
\boldsymbol{0} & \boldsymbol{0} & 0 & \boldsymbol{0} & 0 & 0 & \boldsymbol{0} & 0 & 0 & 0 & 0 & 1 & \boldsymbol{1} & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\boldsymbol{0} & \boldsymbol{0} & 0 & \boldsymbol{0} & 0 & 1 & \boldsymbol{1} & 1 & 1 & 1 & 1 & 0 & \boldsymbol{0} & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
\boldsymbol{0} & \boldsymbol{0} & 1 & \boldsymbol{1} & 1 & 0 & \boldsymbol{0} & 0 & 1 & 1 & 1 & 0 & \boldsymbol{0} & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\
\boldsymbol{0} & \boldsymbol{1} & 0 & \boldsymbol{0} & 1 & 0 & \boldsymbol{0} & 1 & 0 & 0 & 1 & 0 & \boldsymbol{0} & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
\boldsymbol{1} & \boldsymbol{0} & 1 & \boldsymbol{0} & 1 & 1 & \boldsymbol{0} & 1 & 0 & 1 & 0 & 1 & \boldsymbol{0} & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1
\end{matrix} \qquad (3)$$

The additional parity bit is easily generated as the combination of all bits and checked by comparing it to the parity of the code word excluding the bit itself. This process can be done independently or matrices can be modified. Considering the first option, double errors in the message (not affecting the parity bit) are detected because they will produce a Hamming Syndrome different from zero with no error in the recalculation of the parity bit.All triple adjacent errors are detected due to the special configuration of the check matrix, except the one affecting the additional parity bit and its two adjacent code word bits.To detect this error, the additional parity bit is placed prepending the less significant bit. Any error affecting the two first bits of the code word will be always detected because they produce a syndrome (with a value of 3) which is not a valid column value. Thus, a miscorrection is never performed. So if the parity bit is affected too, it will be detected by the previous condition without any additional modification.

V. DOUBLE ADJACENT ERROR CORRECTION

The characteristics of a linear block code are completely determined by its H-matrix. To detect all single bit errors, the corresponding error syndromes should be unique. Note that the syndrome for a single bit error at the bit position p is the same as the p-th column of the H-matrix. To uniquely identify all the single bit errors, all the columns of the H-matrix must be unique. To detect all the double bit errors, the corresponding syndromes should be different from all the single bit error syndromes. To be able to correct all the adjacent double bit errors, the syndromes for the adjacent double bit errors should be different from each other and also different from all the single-error syndromes. Next we define the conditions that must be satisfied by the H-matrix to achieve single error correction, double adjacent error correction and double error detection:

1) No all 0 columns.2) All columns are distinct. 3) No linear dependency involving 3 or less columns.4) No linear dependency involving columns Ci,Cj,Ck,Cmwhere m>k>j>i, such that j=i+1 and m=k+l.
Condition 1 ensures that no single bit error case match the error free case.Condition 2 ensures that all the single error syndromes are unique. Every single error syndrome matches one of the columns of the H-matrix. Since all the columns of the H-matrix are distinct, the single bit errors are uniquely identifiable and hence correctable.
Condition 3 ensures that the syndromes for all double bit errors are different from that of the single bit errors. The syndrome for a double bit error is determined by the XOR of the columns corresponding to the erroneous bit positions. If the H-matrix is free of 3-cycles then the XOR of any two columns of the H-matrix is not identical to any of the columns of the H-matrix. This ensures that the syndromes of all the double bit errors are different from the single bit error syndromes, and condition 2 ensures that the double bit error syndromes are non-zero. Hence all the double bit errors are detectable.
Condition 4 along with condition 2, ensures that a syndrome for an adjacent double bit error is different from all other adjacent double bit error syndromes. If we assume that the only errors are single bit errors or adjacent double bit errors, then with an H-matrix satisfying conditions 1 through 4, we can uniquely identify the syndromes for all single bit errors and adjacent double bit errors and hence can correct all single bit errors and all double adjacent bit errors and detect all double bit errors.
The H-matrix for the (22,16) code to to achieve single error correction, double adjacent error correction and double error detection is shown below in (4)

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 8, August 2015

$$\begin{pmatrix} 1100110001110100100000 \\ 0001010011011001010000 \\ 1010100111100011001000 \\ 1001001110010110000100 \\ 0110101100001101000010 \\ 0111011000101010000001 \end{pmatrix} \quad (4)$$

VI. PROPOSED CODE

The proposed SEC-DED-DAEC-TAED code has the following properties: 1) All single bit errors can be corrected.2) All double bit errors can be detected.3) All adjacent double bit errors can be corrected.4) All triple adjacent bit errors can be detected.

The proposed code is constructed from the ones described in [22] and [23]. For a 16 bit data word the proposed code has 28 bit code word, i.e., 16 bit data and 12 bit check bits. It uses two H-matrices, one for triple adjacent error detection & one for double adjacent error correction. In order to generate the 28 bit code word from a 16 bit data, it first generate the 22 bit code word after multiplying the 16 bit data with the H-matrix for enhanced adjacent error detection and then it generates the 6 check bits using an XOR network corresponding to the G-matrix for double adjacent error correction. The 6 check bits are then appended to the 22 bit code word forming 28 bit code word.

During encoding, the data bits can be directly copied and the check bits are generated. The decoding algorithm is as follows:

1) Generate the syndrome using an the H-matrices.
2) If the syndrome is the all zero vector, then no error is detected, otherwise one or more errors occurred.
3) If the syndrome matches any of the H-matrix columns, then a single error is detected and the error position is the corresponding column position. The corresponding bit should be flipped to correct the error.
4) Else if the syndrome matches any of the n-1 adjacent double error syndromes, then a double adjacent error is detected and the corresponding bit positions are generated using the error correction logic.
5) Else if the syndrome is other than the previous syndromes then a triple adjacent error is detected.

Fig. 1 shows the basic error detection and correction block diagram. If a non-zero syndrome is encountered, then the OR gate flags an error indication. If the syndrome matches any of the single error syndromes then the syndrome decoder generates a 1 in the erroneous bit position. Otherwise, if the syndrome matches any of the adjacent double error syndromes, then the decoder generates 1's at the erroneous adjacent bit positions. Otherwise the output of the syndrome decoder is the all zero output. The syndrome decoder consists of 3-input OR gates whose inputs are driven by outputs of r-input AND gates. The i-th output of the decoder is 1 if and only if a single error occurred at the i-th bit or a double adjacent error occurred at <i,i+1> bits or <i-1,i> bits. The outputs of the decoder are used to generate the corrected word, by using n 2-input XOR gates. If the syndrome is non-zero and does not match any of the single or double adjacent error syndromes, then a triple adjacent error is detected and UE signal is flagged.

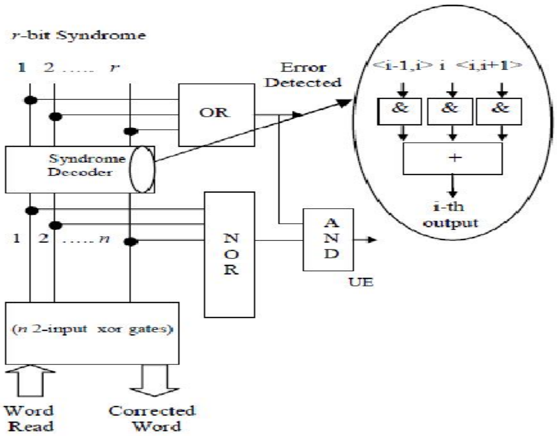


Fig. 1. Error detection and correction block diagram

DOI: 10.15680/IJIRCCE.2015. 0308056

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

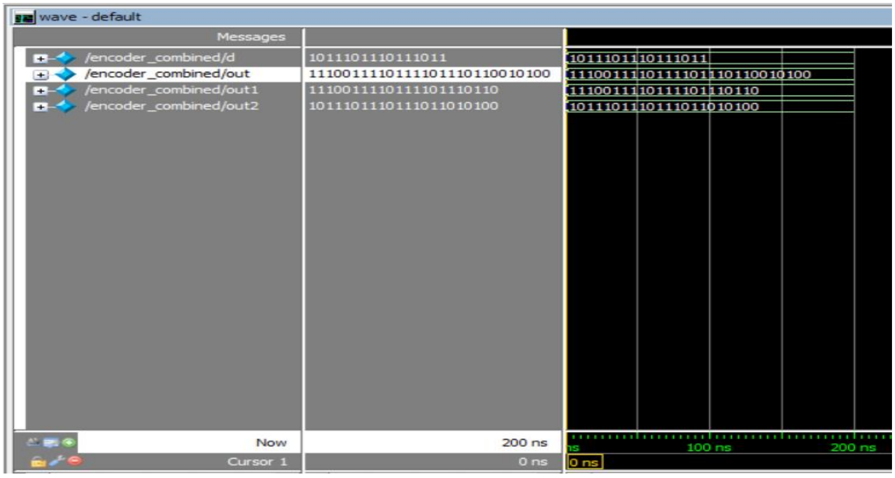
International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 8, August 2015

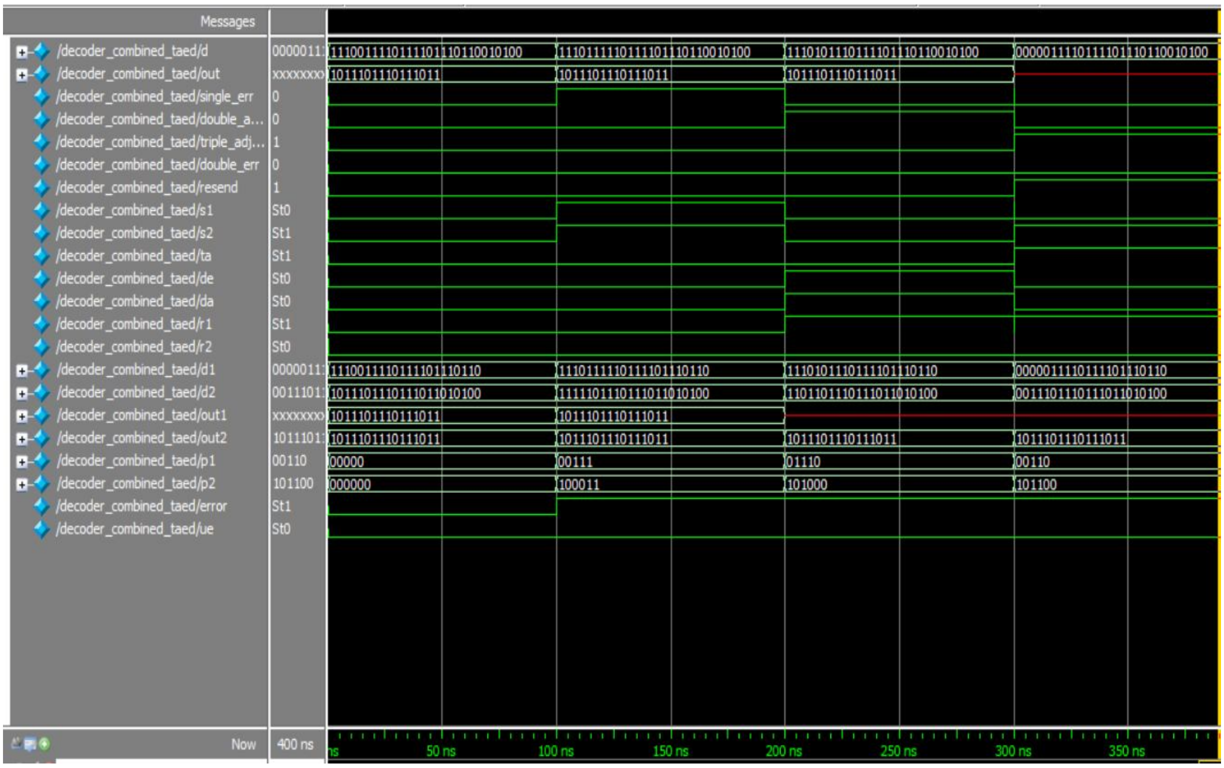
VII. FPGA IMPLEMENTATION & SIMULATION RESULTS

The code for the proposed Hamming code was written in Verilog Hardware Description Language and the code was simulated on ModelSim SE 6.3f for a 16 bit data. It was tested for correct functionality by giving various inputs The proposed code is synthesized on Xilinx 13.3 version and has been implemented on Spartan 3E FPGA kit. The simulation results of the encoder and decoder section of the proposed code is shown in Fig. 2 and Fig. 3. The input to the encoder is a 16 bit data and the output is a 28 bit code word. The 28 bit code word is the input to the decoder.



```
wave - default
Messages
/encoder_combined/d       1011101110111011
/encoder_combined/out     1110011110111101110110010100
/encoder_combined/out1    111001111011110111110110
/encoder_combined/out2    101110111011101110100
Now       200 ns
Cursor 1  0 ns
```

Fig. 2. Simulation result for encoder of the proposed code



```
Messages
/decoder_combined_taed/d            00000111
/decoder_combined_taed/out          xxxxxxxx
/decoder_combined_taed/single_err   0
/decoder_combined_taed/double_a...  0
/decoder_combined_taed/triple_adj...1
/decoder_combined_taed/double_err   0
/decoder_combined_taed/resend       1
/decoder_combined_taed/s1           St0
/decoder_combined_taed/s2           St1
/decoder_combined_taed/ta           St1
/decoder_combined_taed/de           St0
/decoder_combined_taed/da           St0
/decoder_combined_taed/r1           St1
/decoder_combined_taed/r2           St0
/decoder_combined_taed/d1           00000111
/decoder_combined_taed/d2           00111011
/decoder_combined_taed/out1         xxxxxxxx
/decoder_combined_taed/out2         10111011
/decoder_combined_taed/p1           00110
/decoder_combined_taed/p2           101100
/decoder_combined_taed/error        St1
/decoder_combined_taed/ue           St0
Now  400 ns
```

Fig. 3. Simulation result for decoder of the proposed code

VIII. CONCLUSION AND FUTURE WORK

The ECC methodology described in this paper adds the ability to correct double adjacent errors along with the enhanced detection of triple adjacent errors. The proposed code can correct single error, detect double errors, correct double adjacent errors and detect triple adjacent errors. The main drawback is that it adds some area to the existing SEC-DED code because of the additional check bits required for the double adjacent error correction. The proposed methodology is flexible and can be used to construct codes for correcting any subset of double errors. It can help in the detection and correction of adjacent errors which are the ones usually caused by Multiple Cell Upsets (MCUs) in semiconductor memories.

REFERENCES

1. 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.
2. C. L. Chen and M. Y. Hsiao, "Error-correcting codes for semiconductor memory applications: A state-of-the-art review," IBM J. Res. Develop., vol. 28, no. 2, pp. 124–134, Mar. 1984.
3. R. Leveugle, "Optimized state assignment of single fault tolerant FSMs based on SEC codes," in Proc. 30th Conf. Des. Autom., Jun. 1993, pp. 14–18.
4. R. W. Hamming, "Error detecting and error correcting codes," Bell Syst. Tech. J., vol. 29, no. 2, pp. 147–160, Apr. 1950.Anjum Asmaand GihanNagib,'Energy Efficient Routing Algorithms for Mobile Ad Hoc Networks–A Survey', International Journal of Emerging Trends & Technology in computer Science, Vol.3,Issue 1, pp. 218-223, 2012.
5. P. Ankolekar, S. Rosner, R. Isaac, and J. Bredow, "Multi-bit error correction methods for latency-constrained flashmemory systems," IEEE Trans. Device Mater. Rel., vol. 10, no. 1, pp. 33–39, Mar. 2010.
6. Costas Argyrides, Dhiraj K. Pradhan and TaskinKocak, "Matrix Codes for Reliable and Cost Efficient Memory Chips", IEEE Transactions On Very Large Scale Integration (Vlsi) Systems, Vol. 19, No. 3, pp.420-428, March 2011.
7. H. Naeimi and A. DeHon, "Fault secure encoder and decoder for nanoMemory applications," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 17, no. 4, pp. 473–486, Apr. 2009.
8. Maiz, J., S. Hareland, K. Zhang, and P. Armstrong, "Characterization of Multibit Soft Error Events in Advanced SRAMs", Proc. of IEEE Int'l Electronic Device Meeting, pp. 519-522, Dec. 2003.
9. E. Ibe, H. Taniguchi, Y. Yahagi, K. Shimbo, and T. Toba, "Impact of scaling on neutron-induced soft error in SRAMs from a 250 nm to a 22 nm design rule," IEEE Trans. Electron Devices, vol. 57, no. 7, pp. 1527–1538, Jul. 2010.
10. R. K. Lawrence and A. T. Kelly, "Single event effect induced multiple-cell upsets in a commercial 90 nm CMOS digital technology," IEEE Trans. Nucl. Sci., vol. 55, no. 6, pp. 3367–3374, Dec. 2008.
11. S. Satoh, Y. Tosaka, and S. A. Wender, "Geometric effect of multiple-bit soft errors induced by cosmic ray neutrons on DRAM's," IEEE Electron Device Lett., vol. 21, no. 6, pp. 310–312, Jun. 2000.
12. Hsiao, M. Y., "A Class of Optimal Minimum Odd-weightcolumn SEC-DED codes", IBM Journal of Research and Development, Vol. 14, pp. 395-401, 1970.
13. Reddy, S.M., "A Class of Linear Codes for Error Control in Byte-per-Package Organized Memory Systems", IEEE Trans. On Computers, Vol. C-27, pp. 455-458, May. 1978.
14. Chen, C. L.," Error Correcting Codes with Byte Error Detection Capability", IEEE Trans. On Computers, Vol. 32, pp. 615-621, May 1983
15. Lin, S., and D. J. Costello, Jr., Error Control Coding: Fundamentals and Applications, Prentice-Hall, 1983.
16. Berlekamp, E. R., Algebraic Coding Theory, McGraw-Hill, New York, 1968.
17. Lala,P.K.,"An Adaptive Double Error Correction Scheme for Semiconductor Memory Systems,"Digital Processes,Vol.4, pp 237-243, 1978.
18. Vargas, F. L., and M. Nicolaidis," SEU-Tolerant SRAMDesign Based On Current Monitoring", Proc. Int. Symposium on Fault Tolerant Computing, pp. 106-115, June 1994.
19. Calin, Th., F. L. Vargas, and M. Nicolaidis,"Upset TolerantCMOS Using Current Monitoring: Prototype and Test Experiments", Proc. Int. Test Conference, pp. 45-53, 1995.
20. Gill, B., M. Nicolaidis, and C. Papachristou," RadiationInduced Single-Word Multiple-bit Upsets Correction in SRAM" Proc. of Int. Online Test Symposium, pp. 266-271, Jul. 2005.
21. Sanchez-Macian, P. Reviriego, and J. A.Maestro, "Enhanced detection of double and triple adjacent errors in hamming codes through selective bit placement," IEEE Trans. Device Mater. Rel., vol. 12, no. 2, pp. 357– 362, Jun. 2012.
22. Sanchez-Macian, P. Reviriego and J. A. Maestro, "Hamming SECDAED and Extended Hamming SEC-DED-TAED codes through selective shortening and bit placement," IEEE Transactions on Device and Materials Reliability, vol. 14 No.1,pp-574-576, March2014.
23. Dutta and N. A. Touba, "Multiple bit upset tolerant memory using a selective cycle avoidance based SEC-DED-DAEC code," in Proc. 25th IEEE VLSI Test Symp., May 2007, pp. 349–354.

BIOGRAPHY

Babitha Antony is doing MTech in VLSI & Embedded Systems in Electronics & Communication Department, Sree Narayana Gurukulam College of Engineering, Ernakulam, India. Her research interests are Error correction codes, Fault tolerance etc.