www.ijecs.in
International Journal Of Engineering And Computer Science ISSN: 2319-7242
Volume 4 Issue 10 Oct 2015, Page No. 14670-14676

Analytical Approach For Decoder Delay Reduction Sec-DED-DAEC Codes Derived From Orthogonal Latin Square Codes

B.Swarupa (Pg Scholar) [1] ***Y. Subhash Kumar Assistant Professor***[2]

Department of ECE, Lords Institute of Engineering and Technology, Hyderabad, INDIA

Abstract

Although tremendous progress has done in past years on memory designing but still Radiation-induced soft errors is concerned area in the field of soft memories and the single error correction double error detection (SEC-DED) codes are commonly used to give assured memory contents with absence of corrupted scenario. Since SEC-DED codes cannot correct multiple errors, they are often combined with interleaving. Interleaving, however, impacts memory design and performance and cannot always be used in small memories. This limitation has spurred interest in codes that can correct adjacent bit errors. In particular, several SEC-DED double adjacent error correction (SEC-DED-DAEC) codes have recently been proposed. Implementing DAEC has a cost as it impacts the decoder complexity and delay. Another issue is that most of the new SEC-DED-DAEC codes miscorrect some double nonadjacent bit errors. In this brief, a new class of SEC-DED-DAEC codes is derived from orthogonal Latin squares codes. The new codes significantly reduce the decoding complexity and delay. In addition, the codes do not miscorrect any double nonadjacent bit errors. The main disadvantage of the new codes is that they require a larger number of parity check bits. Therefore, they can be useful when decoding delay or complexity is critical or when miscorrection of double nonadjacent bit errors is not acceptable. The proposed codes have been implemented in Hardware Description Language and compared with some of the existing SEC-DED-DAEC codes. Finally the experimental results confirm the reduction in decoder delay.

KEYWORDS: Error correction codes, Orthogonal Latin square codes. Single error correction double error detection (SEC-DED), Double adjacent error correction (DAEC), Memory

1. INTRODUCTION

Memories are very dense structures that are especially susceptible to defects. Transient errors due to radiation, power supply noise, etc., can cause bit-flips in a memory. To protect the data integrity of the memory during runtime operations, error correcting codes (ECC) of various class and strength is generally employed A soft error occurs when a radiation event causes enough of a charge disturbance to reverse or flip the data state of a memory cell, register, latch, or flip-flop. The error is "soft" because the circuit/device itself is not permanently damaged by the radiation—if new data are written to the bit, the device will store it correctly.

Recently, research has shown that commercial static random access memories (SRAMs) are now so small and sufficiently sensitive that single event upsets (SEUs) may be induced from the electronic stopping of a proton. This sensitivity appears near the 65 nm technology node as the critical charge to upset a cell is on the order of 1 fC; merely 6,000 electrons are required to cause a change in data state. The lower critical charge required to cause a bit-flip has more pronounced effects on space applications compared to terrestrial ones. Also low voltage operation can lead to greater number of failures, arising due to more pronounced effect of process variations. Voltage scaling, which is one of the most effective ways to reduce power consumption can lead to unreliable operations at lower voltages. Voltage scaling is

limited by a minimum value referred to as VCC min beyond which circuits may not function reliably. Voltage scaling beyond Vccmin gives rise to reliability issues, most notably for the memory sub-systems. In order for Vccmin to be reduced to enable ultra-low power modes in microprocessors and other circuits, some means for handling high memory bit failure rates is required.

To protect memories, error correction codes are commonly used. Traditionally, single error correction double error detection (SEC-DED) codes have been used. A SEC-DED code has a minimum Hamming distance of four and is able to correct single bit errors and detect double errors without miscorrection. This is important to avoid silent data corruption. SEC-DED codes are sufficient when errors affect only one bit, however, the percentage of soft errors affecting more than a single bit is increasing as technology scales. For memories implemented in 40 nm and below, multiple bit errors are a significant percentage of errors and thus SEC-DED codes alone are no longer sufficient to protect memories. One option is to combine SEC-DED codes with interleaving. Interleaving, places the bits that belong to the same logical word physically apart. As the errors caused by a radiation particle hit are physically close, this ensures that the errors affect at most one bit per logical word. Interleaving has an impact on the memory design.

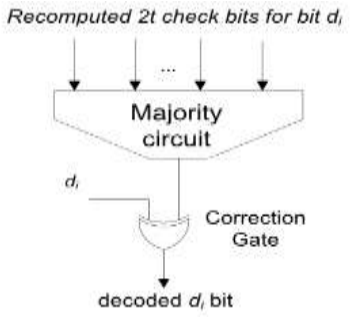

Figure 1: Illustration of OS-MLD decoding for OLS codes

Another alternative is to use error correction codes that can correct adjacent bits. In many cases, directly adjacent bits account for over 90% of the observed multiple bit errors. Several codes have been recently proposed to this end. For example, a code that can correct double and triple adjacent errors for words of 16 bit was presented. A technique to design SEC-DED double adjacent error correction (SEC-DED-DAEC) codes was introduced. The extension of SEC-DED-DAEC codes to also detect larger burst errors has also been recently considered. One issue with those SEC-DED-DAEC codes is that they can miscorrect some double nonadjacent bit errors. The reduction of the miscorrection probability has been considered. The algorithm tries to minimize the number of 4 cycles. It was shown that miscorrection can be avoided for the most common error patterns and in some cases for all patterns at the cost of adding additional parity check bits. Another issue with SEC-DED-DAEC codes is that their decoding complexity and latency are larger than those of SEC-DED codes.

The main limitation for these codes is that they require a number of parity check bits equal to the number of data bits. The use of more advanced codes such as difference set and orthogonal Latin squares (OLS) codes to correct adjacent errors has also been considered. Those codes are one-step majority logic decodable (OS-MLD) and therefore, can be decoded with low latency. In this brief, a new class of SEC-DED-DAEC codes is presented. The proposed codes are derived from OLS codes. They require fewer parity check bits than double error correction (DEC) OLS codes and are simpler to decode. Compared with existing SEC-DED-DAEC codes, the new codes have two main advantages: first, there is no miscorrection for double nonadjacent errors and second, the decoding is much simpler and faster. The main drawback for the proposed codes is that they require more parity check bits than existing SEC-DED-DAEC codes. Therefore, the proposed codes can be useful to protect memories in which decoding latency is critical or miscorrection cannot be tolerated. The rest of this brief is organized as follows.

DOI: 10.18535/ijecs/v4i10.22

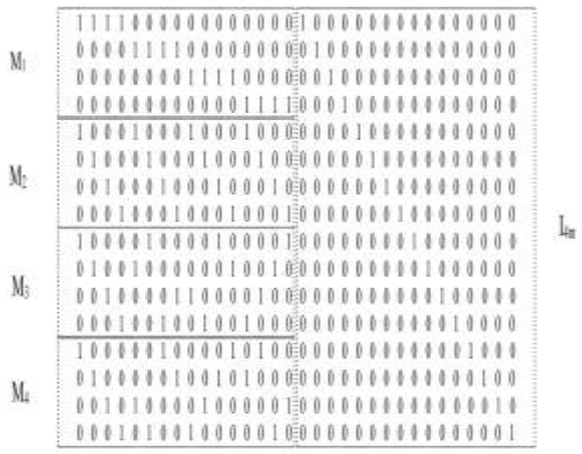

Figure 2: Parity check matrix H for the OLS code with k=16 and t =2

2. ERROR CORRECTING CODES

The most common error correcting code that is used is single-error-correcting, double-error-detecting (SEC-DED) codes. These codes can correct single bit errors in any word of the memory and can detect double bit errors, have moderate redundancy in terms of check bits and are relatively easy to decode. Decoding and correction are done via syndrome method which takes single cycle. A special class of SEC-DED codes known as Hsiao codes [Hsiao 70] was proposed to improve the speed, cost, and reliability of the decoding logic. However some situations demand more stringent reliability requirements, thus necessitating error correction stronger than normal SEC-DED.

Stronger error correcting codes includes single byte-error-correcting, double-by te error-detecting (SBC-DBD) codes. These codes perform at a higher order Galois field and consequently the encoding and decoding are more complex. Moreover, they require more check bits thereby increasing the size of the memory. There are also the double-error-correcting triple-error-detecting (DEC-TED) codes, which come 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 The general drawbacks 3 with these methods are latency and speed. Most of these codes require several cycles to correct the first error unlike the SEC-DED codes. Moreover, the encoding and decoding are much more complex and require several table lookups for multiplication in higher order fields. However in spite of their low check bits overhead and single cycle decoding, SEC-DED codes are not able to provide requisite reliability under certain conditions.

3. OLS CODES

The OLS codes were introduced decades ago to protect memories. On the end simultaneously have recently been proposed to protect caches and interconnects. The block sizes for OLS codes are k=m 2 data bits and 2tm parity bits. Where t is the number of errors that the code can correct and mis an integer. For memories, the word sizes are typically a power of two and therefore mis commonly also power of two. The main advantage of OLS codes is that their decoding is simple and fast. This is because, as mentioned in the introduction, OLS codes can be decoded using OS-MLD. In OS-MLD, each bit is decoded by simply taking the majority value on the set of the recomputed parity check equations (or syndrome bits) in which it participates. The idea behind OS-MLD is that when an error occurs in bit di, the recomputed parity checks in which it participates will take a value of one unless there are errors in other bits. Therefore, a majority of ones in those recomputed checks is an indication that the bit is in error and therefore needs to be corrected. If the code is such that two bits share at most one parity check, thent−1 errors on other bits will not affect the majority of the 2t vote and therefore, the error will be corrected. Only a few codes have this property and can be decoded using OS-MLD. This is the case for difference set codes and for OLS codes, as mentioned in the introduction.

More formally, the construction of OLS codes is such that:

1) Each data bit participates in exactly 2t parity check bits;

2) Each other data bit participates in at most one of those parity check bits.

Therefore, for a number of errorst or smaller, when one error affects a given bit, the remaining t −1 errors can, in the worst case affect t −1 check bits on which that bit participates. Therefore, still a majority oft +1 will trigger the correction on the erroneous bit. Conversely, when a given bit is correct ,t errors on other bits will not cause miscorrection as a majority

of t +1 is needed. As shown in Fig. 1, the use of OS-MLD enables a simple and fast decoding that is attractive to protect memories when decoding latency is critical. Another characteristic of OLS codes is that they correct only errors on the data bits. No correction is done for the parity bits. This is not an issue as in memories; the goal is to recover the stored data correctly.

The proposed codes are derived from DEC OLS codes. These are block linear codes that are defined by their parity generating G and parity check H matrixes. The parity check matrix is used to detect errors by computing the syndrome s that is obtained by multiplying the stored word by the H matrix

k	n-k	m
16	16	4
64	32	8
256	64	16

TABLE I: PARAMETERS OFSOMEDEC OLS CODES

The parity check matrix H for a DEC OLS code with $k=m^2$ is constructed as follows:

$$H = \begin{bmatrix} M_1 \\ M_2 \\ M_3 \\ M_4 \end{bmatrix} I_{4m} \quad (1)$$

Where I_{4m} is the identity matrix of size 4mand M1, M2, M3, M4 are matrices with size $m \times m^2$ derived from OLS of size m×m. The weight or the number of ones, of all the columns, in the M_i matrices must be one. Therefore, the first $k = m^2$ columns in H have a number of one's equal to 2t (four for DEC codes). In addition, any pair of columns has at most a position with a one in common. This as discussed before enables the use of OS-MLD for decoding. As an example, the H matrix for a code with $k = m^2 = 16$ data bits and 2tm=16 parity bits that can correct double errors is shown in Fig. 2. The parameters of some DEC OLS codes are summarized in Table I.

4. PROPOSEDSEC-DED-DAEC CODES AND ANALYSIS

The proposed codes are derived from DEC OLS codes. Taking the parity check matrix in (1) as a starting point, the first step is to remove the m parity check bits that correspond to one of the Mi matrices. As an example, consider removing theM1matrix from the matrix in Fig. 2 as shown in Fig. 3. The data bits that participated in each of the removed parity check equations will not share any parity check in the reduced matrix. This is a direct consequence from the property of OLS codes that any two data bits share (that is have a one in the same row in the H matrix) at most one parity check bit. This can be clearly observed in Fig. 2. In addition, those groups of m bits are marked asg1,g2, g3,andg 4in Fig. 3.

For example, the first four data bits share the first parity check bit in theM1 matrix and form the first group g1. It can be observed that they do not share any other parity check bits. Therefore, whenM1is removed they do not share any parity check bit. The same occurs for the other groups of bits 5–8 (g2), 9–12 (g3), and 13–16 (g4). In the reduced matrix, each data bit participates in three parity checks. Therefore, if a majority vote is used to decode the bits, single and double errors can be corrected.

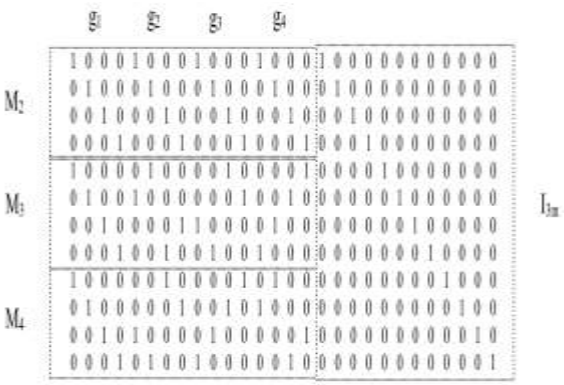

Figure 3: Reduced parity check matrix H after the removal of M_1

DOI: 10.18535/ijecs/v4i10.22

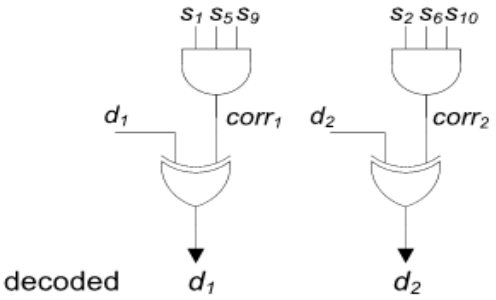

Figure 4: Illustration of the proposed decoder for data bits 1 and 2

However, double errors can also cause miscorrection on other bits. Therefore, the modified matrix, when a majority vote is used, is only effective in correcting single errors. However, let us consider that instead of a majority vote, the logical AND of the three parity checks is used. In Fig. 4, this is shown for the first two data bits where the s i values correspond to bits of the syndrome vector obtained by multiplying the word by the H matrix. In this case, the code will obviously not miscorrect when there are two errors. Single errors on data bits will also be corrected. Double errors affecting data bits will also be corrected as long as the data bits do not share any parity check bit.

However, some double adjacent errors may affect bits on different groups. For example, an error on bits 8 and 9 affects it in g_2 and another in g_3. These bits share parity check bit 7 and therefore, will not be corrected as that recomputed parity bit will take a value of zero in the syndrome as it has two bits in error. This effect can be avoided by carefully placing the bits in the memory. For example, the bits within each group can be reordered to ensure that the ones at the borders does not share any parity check bit with the adjacent bit on the other group. Another issue that can occur is that a double adjacent error affects two parity bits and the error is confused with a double nonadjacent error. For example, an error on parity check bits 4 and 5 produces the same syndrome as an error that affects data bit 16 and parity check bit 11. This can lead to silent data corruption leaving an error on data bit 16 undetected. However, this issue can also be solved by carefully placing the bits into the memory.

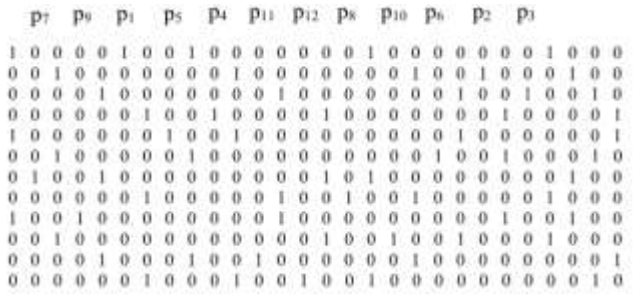

Figure 5: Reduced parity check matrix H after the removal of M1 with the proposed bit placement

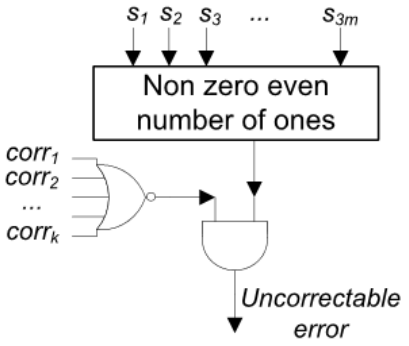

Figure 6: Detection of double uncorrectable errors in the proposed scheme

The proposed bit placement is as follows: 1) ensure that the bits at the borders of the groups do not share any parity check bits and 2) interleave the parity check bits with the data bits so that no double adjacent error affects two parity bits. An example of this bit placement for the code with k =16 is shown in Fig. 5. The parity bits are marked in the figure and obviously, they can only be placed such that the adjacent columns do not participate in the parity bit. With this bit placement, all double adjacent errors affect at least a data bit and that data bit is corrected.

In addition, for nonadjacent errors that affect two bits, if any bit is corrected it means that the error is correctable. When the error affects two data bits, either they are both corrected or there is no correction. Obviously, when the error affects a data bit and a parity bit, if the data bit is corrected the error was correctable.

This enables a simple method to detect uncorrectable errors. The proposed scheme to detect the uncorrectable errors is shown in Fig. 6. It is based on

detecting a nonzero even number of ones in the syndrome that can only be caused by a multiple bit error and checking if any correction has been made.

k	n-k	m
16	12	4
64	24	8
256	48	16

TABLE 2: PARAMETERS OF THEPROPOSEDSEC-DED-DAEC CODES

The proposed scheme can be summarized as follows:

1) Reduce the H matrix of the DEC OLS code by eliminatingM1;

2) Place the bits in the groups ofmbitsg_1, g2,..., gm such that the bits at the borders of the groups do not share any parity check;

3) Interleave the parity bits with the data bits such that two adjacent bits never participate in the same parity bit;

4) Instead of majority voting, decode based on unanimity (three-way AND) to correct errors;

5) Implement the circuit of Fig. 6 to detect uncorrectable errors.

5. EXPERIMENTAL RESULTS

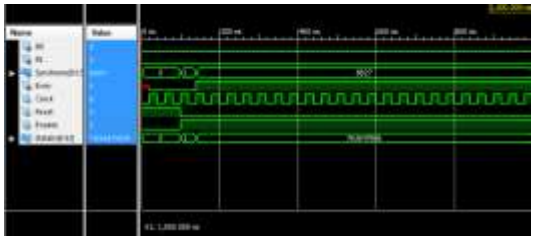

Figure 7: SIMULATION RESULT OF Error MLD test

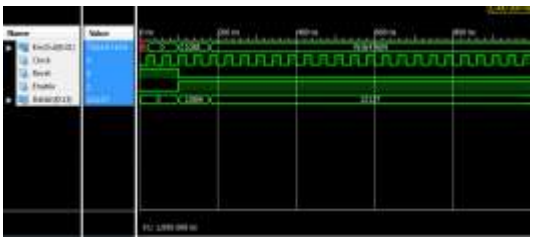

Figure 8: SIMULATION RESULT OF OLS Encoder test

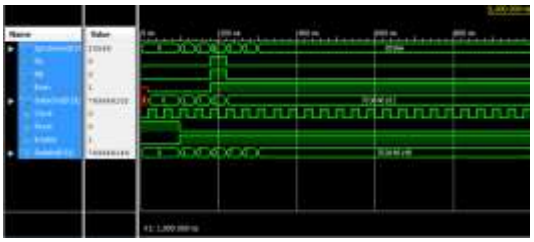

Figure 9: SIMULATION RESULT OF OLS Decoder test

6. CONCLUSION

In this brief, a new class of SEC-DED-DAEC codes has been presented. The codes are derived from DEC OLS codes and can be decoded with low latency. Another interesting feature is that the codes do not experience miscorrection when double nonadjacent error occurs. This is interesting to minimize silent data corruption. The codes can also correct some nonadjacent double errors.

Compared with existing SEC-DED-DAEC codes, they require a larger number of parity check bits, therefore, they are attractive when low latency decoding is a required. The codes have been implemented in HDL and the resulting implementations compared with existing SEC-DED-DAEC codes to put the reductions in decoding latency in perspective. The ideas used to derive the proposed SEC-DED-DAEC can also be used to derive burst error correction codes from OLS codes that can correct multiple errors. The key observation is that the structure of OLS codes is such that the data bits can be divided in groups of m bits that do not share any parity check. Therefore, any error affecting up to 2t−1 bits in one of these groups can be corrected. This can be exploited by carefully placing the data and parity check bits so that, in the best case, up to 2t−1 adjacent bit errors can be corrected. The development of burst error correction codes is an interesting

avenue to continue and extend the work presented in this brief.

REFERENCES

[1] R. C. Baumann, "Soft errors in advanced computer systems,"IEEE Des. Test. Comput., vol. 22, no. 3, pp. 258–266, May/Jun. 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] M. Y. Hsiao, "A class of optimal minimum odd-weight-column SEC-DED codes,"IBM J. Res. Develop., vol. 14, no. 4, pp. 395–301, Jul. 1970.

[4] 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.

[5] P. Reviriego, J. A. Maestro, S. Baeg, S. Wen, and R. Wong, "Protection of memories suffering MCUs throughthe selection of the optimal interleaving distance," IEEE Trans. Nucl. Sci., vol. 57, no. 4, pp. 2124–2128, Aug. 2010

[6] 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.

[7] S. Baeg, S. Wen, and R. Wong, "Minimizing soft errors in TCAM devices: A probabilistic approach to determining scrubbing intervals," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 57, no. 4, pp. 814–822, Apr. 2010.

[8] X. She, N. Li, and D. W. Jensen, "SEU tolerant memory using error correction code,"IEEE Trans. Nucl. Sci., vol. 59, no. 1, pp. 205–210, Feb. 2012.

[9] A. Dutta and N. A. Touba, "Multiple bit upset tolerant memory using a selective cycle avoidance based SEC-DED-DAEC code," inProc. 25th IEEE VLSI Test Symp., May 2007, pp. 349–354.

[10] A. Neale and M. Sachdev, "A new SEC-DED error correction code subclass for adjacent MBU tolerance in embedded memory," IEEE Trans. Device Mater. Rel., vol. 13, no. 1, pp. 223–230, Mar. 2013.