International Journal of Research Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 07
March 2018

An Efficient Single and Double-Adjacent Error Correcting Parallel Decoder for the (24,12) Extended Golay Code

Mr. P. SATISH BABU, Assistant Professor
S. Anusha, V. Naga Jyothi, T. Vijaya Kumar, Md. Hazi Moinuddin Basha
B.Tech Students , Department of Electronics & Communication Engineering, Ramachandra College of Engineering, Eluru, Andhra Pradesh, India.

Abstract— Memories that operate in harsh environments, like for example space, suffer a significant number of errors. The error correction codes (ECCs) are routinely used to ensure that those errors do not cause data corruption. However, ECCs introduce overheads both in terms of memory bits and decoding time that limit speed. In particular, this is an issue for applications that require strong error correction capabilities. A number of recent works have proposed advanced ECCs, such as orthogonal Latin squares or difference set codes that can be decoded with relatively low delay. The price paid for the low decoding time is that in most cases, the codes are not optimal in terms of memory overhead and require more parity check bits. On the other hand, codes like the (24,12) Golay code that minimize the number of parity check bits have a more complex decoding. A compromise solution has been recently explored for Bose–Chaudhuri–Hocquenghem codes. The idea is to imple-ment a fast parallel decoder to correct the most common error patterns (single and double adjacent) and use a slower serial decoder for the rest of the patterns. In this brief, it is shown that the same scheme can be efficiently implemented for the (24,12) Golay code. In this case, the properties of the Golay code can be exploited to implement a parallel decoder that corrects single- and double-adjacent errors that is faster and simpler than a single-error correction decoder. The evaluation results using a 65-nm library show significant reductions in area, power, and delay compared with the traditional decoder that can correct single and double-adjacent errors. In addition, the proposed decoder is also able to correct some triple-adjacent errors, thus covering the most common error patterns.

Index Terms— Double adjacent error correction (DAEC), error correction codes (ECCs), Golay code, memory, single error correction (SEC), triple-adjacent error correction.

I. INTRODUCTION

Harsh environments, like space, are a challenge for electronic circuits in general and for memories in particular. For example, radiation causes several types of errors that can disrupt the circuit functionality [1]. One common error for SRAM memories is soft errors that change the value of one or more memory cells [2]. To avoid corruption in the data stored in the memory, error correction codes (ECCs) are commonly used [3].

ECCs add parity check bits to each memory word to detect and correct errors. This requires an encoder to compute those bits when writing to the memory and a decoder to detect and correct errors when reading from the memory. These elements increase the memory area and the power consumption, and can also reduce the access speed. These overheads increase with the error correction capability of the ECC. Traditionally, codes that can correct a single bit error per word have been used. In particular, single error correction–double error detection (SEC–DED) codes that can also detect double errors are commonly used [4].

Manuscript received May 4, 2015; revised July 8, 2015; accepted August 5, 2015.

In recent years, the number of errors that affect more than one memory cell has increased significantly. This is due to the scaling of the memory cells and is projected to grow further [5]. These errors, known as multiple cell upsets (MCUs), pose a challenge for SEC–DED codes. One solution to ensure that the MCU errors can be corrected is to interleave the bits of different logical words so that an MCU affects one bit per word [6]. This is based on the observation that the cells affected by an MCU are physically close [7]. Interleaving, however, has a cost as it complicates the memory design [8]. In some space applications, there is an additional issue as the number of errors is high, and SEC–DED codes may not be sufficient when errors accumulate over time [9]. These issues have led to an increased interest on the use of more advanced ECCs to protect SRAM memories.

As MCUs affect cells that are close together, a number of codes that can correct double-adjacent or triple-adjacent errors have been recently proposed [8], [10]–[12]. These codes, in many cases, do not require additional parity check bits and in the rest require only one or two additional bits. The decoding complexity increases but in many cases can still be implemented with limited impact on the memory speed. These codes are useful for applications in which the error rate is low, however, when the error rate is large, codes that can correct errors on multiple independent bits are needed [9].

Research for multibit ECCs has focused on reducing the decoding latency as in many cases, the traditional decoders are serial and require several clock cycles. To some extent this can be done for some traditional ECCs by using a parallel syndrome decoder [13] but the decoder complexity explodes as the error correction capability or the word size increases. Another approach is to use codes that can be decoded with low delay, such as orthogonal Latin squares (OLSs) or difference set (DS) codes [14], [15]. In the case of OLS codes, the main issue is that they are not optimal in terms of the number of parity check bits and thus require more memory overhead. The DS codes are more competitive in terms of parity check bits but are still not optimal for some word lengths. For example, the (21, 10) DS code can correct 2-bit errors while a code with a similar block size and code rate, and the (24,12) extended Golay code [16] can correct 3-bit errors. However, the Golay code requires a more complex decoder that needs several clock cycles [17].

Namba *et al.* [18] have proposed a compromise solution for Bose–Chaudhuri–Hocquenghem codes. The idea is that the most common error patterns are decoded in parallel and the rest serially.

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

Fig. 1. Parity check matrix of the (24,12) Golay code.

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

Fig. 2. Parity check matrix of the (24,12) Golay code with the proposed bit placement.

traditional SEC decoder but that can also correct all double-adjacent errors and some triple-adjacent errors. The proposed decoder has been implemented in hardware description language and mapped to a 65-nm technology to show its benefits. The main contribution of this brief is to enable a fast and efficient parallel correction of the single and double-adjacent errors in the (24,12) Golay code.

The rest of this brief is organized as follows. Section II provides an introduction to the (24,12) Golay code. The proposed decoder is presented in Section III. The evaluation results are summarized and discussed in Section IV. Finally, this brief ends with the conclusions in Section V.

The requirement for SEC is that the columns must be different. Therefore, it would seem possible to use a subset of the parity bits to decode single errors. However, since the code can correct three errors, we need to ensure that the single-error parallel decoder does not introduce erroneous corrections in the presence of multiple bit errors. For example, if we use an SEC-DAEC code with a minimum distance of four, a triple error can cause a miscorrection in the SEC-DAEC decoding phase. A 4-bit error may not be even detected by the SEC-DAEC decoder. Therefore, the full syndrome is used for comparisons in all the cases to ensure that triple errors do not trigger miscorrections and 4-bit errors are detected.

II. (24,12) EXTENDED GOLAY CODE

The (24,12) extended Golay code is obtained by adding an overall parity check bit to the (23, 12) Golay code [16]. This code is a perfect code with a minimum distance of seven and has been widely studied [19]. The extended code has a minimum distance of eight, and therefore can correct 3-bit errors and detect 4-bit errors. It has been used in many applications including space missions that require strong error correction capabilities [19]. The decoding of the Golay code is done in a series of steps [17], [20] and requires several clock cycles. For example, 27 clock cycles are needed in the implementation presented in [17]. This, as discussed before, is not suitable for SRAM protection. To the best of our knowledge, no SEC-DAEC parallel decoder optimized for the Golay code has been proposed in the literature.

The parity check matrix of the (24,12) Golay code is shown in Fig. 1. The 12 first bits correspond to the parity check bits and last 12 to the data bits. A single-error correcting parallel decoder can be implemented by computing the syndrome and comparing in parallel with the 12 data bit and the 12 check bit columns. When there is a match that bit is corrected [4].

$$\begin{array}{cccc}
1 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 \\
1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1
\end{array}$$

Fig. 3. Example of syndrome values for errors that affect the first data bit in the proposed bit placement.

TABLE I
NUMBER OF COMPARATORS REQUIRED FOR THE DECODER

	Comparators	*Bits*
SEC	24	12
SEC-DAEC	47	12
Proposed SEC-DAEC	24	10/11/12

that change from one pattern to another and that the values cover the four possible combinations of the first two bits. This means that the decoding can be done by simply comparing the remaining ten bits with the last ten bits of the syndrome. If they match, then the second bit (first data bit) has to be corrected. It can be observed that the same reasoning applies to the rest of the data bits, except the last one. For the last bit, there are only two values to check (single and double adjacent with bit 23). In this case, it is easy to see that this can be done by checking the first 11 bits only.

The previous discussion shows how parallel decoding can be efficiently implemented. In fact, the proposed parallel decoder will be simpler than an SEC decoder. Table I summarizes the comparators needed for each of the different decoders. A comparator is needed for each syndrome value that triggers a correction. For an SEC code, this is simply 24 while for a traditional SEC-DAEC code is 47. In the case of the proposed decoder, 12 comparators cover both single bit errors on the data bits and double adjacent bit errors, and

TABLE II
DECODER AREA (μm^2)

	Area optimized	Delay optimized
SEC	556	2770
SEC-DAEC	902	5036
Proposed SEC-DAEC	493	2399

TABLE III
DECODER DELAY (ns)

another 12 are needed to cover single errors on the check bits giving a total of 24. It can be observed that the proposed decoder needs less comparator and also less bits in some of them. Both factors help to reduce the decoder complexity. In Section IV, the benefits will be evaluated for a design mapped to a 65-nm technology.

The proposed parallel decoder also has to detect errors that it cannot correct. In those cases, the serial decoder must be used to correct the error. The logic needed to detect those errors is simply a check for a no zero syndrome and a check that none of the comparators has detected a match. The first part can be implemented with a 12-input OR gate and the second with another 24-input OR gate.

It should be noted that the same idea can be partly applied to other triple ECCs even if the number of parity check and data bits is not the same. In more detail, when there are more data bits, the first data bits can also be interleaved with parity bits and decoded with the proposed scheme, while for the rest, a traditional SEC-DAEC decoding can be used. The application of the proposed scheme to other codes is left for future work.

TABLE IV
DECODER POWER (mW)

	Area optimized	Delay optimized
SEC	0.82	2.36
SEC-DAEC	1.06	3.80
Proposed SEC-DAEC	0.76	2.60

IV. EVALUATION

The proposed parallel SEC-DAEC decoder has been implemented in HDL and mapped to a TSMC 65-nm technology library using Synopsys Design Compiler. The traditional SEC and SEC-DAEC decoders have also been implemented to show the benefits of the new decoder.

The synthesis was done twice, one with maximum effort to reduce area and another with maximum effort to reduce delay. The first one shows the benefits in area when delay is not an issue, and the second shows the maximum speed that can be achieved by the decoder.

The results for decoder area, delay, and power are shown in Tables II–IV. It can be observed that optimizing for delay leads, in all cases, to a large increase in circuit area and power. The proposed SEC-DAEC decoder requires a less circuit area than both the traditional SEC-DAEC decoder and an SEC decoder. In the case of the area optimized synthesis, the reductions are over 45% and 11%, respectively. This clearly confirms that the proposed SEC-DAEC decoder is much simpler than the traditional SEC-DAEC decoder and more interestingly, and is also simpler than the SEC decoder. In terms of delay, the proposed decoder is also faster than both the traditional SEC-DAEC decoder and an SEC decoder. For the delay optimized synthesis, the reductions are 28% and 21% compared with the traditional SEC-DAEC decoder and the SEC decoder, respectively. Finally, the power consumption is significantly smaller than for the traditional SEC-DAEC decoder and similar to that of the SEC decoder (slightly lower for area optimized and slightly larger for the delay-optimized synthesis).

In summary, the evaluation for a commercial library confirms that the proposed SEC-DAEC parallel decoder reduces significantly the cost of implementing DAEC compared with the traditional SEC-DAEC decoder. In fact, the area and delay are also lower than for a parallel SEC decoder.

V. Conclusion

In this brief, a single and double-adjacent error correcting parallel decoder for the (24,12) extended Golay code has been proposed. The decoder uses the properties of the code to achieve an efficient implementation. In fact, the proposed decoder is not only much simpler than a traditional SEC-DAEC decoder, but also simpler than a standard SEC decoder for the Golay code. To evaluate the benefits of the new decoder, it has been implemented in HDL and mapped to a 65-nm library. The results confirm that significant reductions in area, delay, and power consumption can be obtained compared with the traditional SEC-DAEC decoder.

The new SEC-DAEC parallel decoder can be used in conjunction with a serial decoder so that the most common error patterns are corrected in one clock cycle.

References

[1] R. D. Schrimpf and D. M. Fleetwood, *Radiation Effects and Soft Errors in Integrated Circuits and Electronic Devices*. Singapore: World Scientific, 2004.

[2] R. C. Baumann, "Soft errors in advanced computer systems," *IEEE Des. Test. Comput.*, vol. 22, no. 3, pp. 258–266, May/Jun. 2005.

[3] 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.

[4] M. Y. Hsiao, "A class of optimal minimum odd-weight-column SEC-DED codes," *IBM J. Res. Develop.*, vol. 14, no. 4, pp. 301–395, Jul. 1970.

[5] 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.

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

[7] 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.

[8] 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.

[9] M. A. Bajura *et al.*, "Models and algorithmic limits for an ECC-based approach to hardening sub-100-nm SRAMs," *IEEE Trans. Nucl. Sci.*, vol. 54, no. 4, pp. 935–945, Aug. 2007.

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

[11] Z. Ming, X. L. Yi, and L. H. Wei, "New SEC-DED-DAEC codes for multiple bit upsets mitigation in memory," in *Proc. IEEE/IFIP 20th Int. Conf. VLSI Syst.-Chip*, Oct. 2011, pp. 254–259.

[12] L.-J. Saiz-Adalid, P. Reviriego, P. Gil, S. Pontarelli, and J. A. Maestro, "MCU tolerance in SRAMs through low-redundancy triple adjacent error correction," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, to be published.

[13] R. Naseer and J. Draper, "DEC ECC design to improve memory reliabil-ity in sub-100 nm technologies," in *Proc. IEEE ICECS*, Aug./Sep. 2008, pp. 586–589.

[14] M. Y. Hsiao, D. C. Bossen, and R. T. Chien, "Orthogonal Latin square codes," *IBM J. Res. Develop.*, vol. 14, no. 4, pp. 390–394, Jul. 1970.

[15] S.-F. Liu, P. Reviriego, and J. A. Maestro, "Efficient majority logic fault detection with difference-set codes for memory applications," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 1, pp. 148–156, Jan. 2012.

[16] M. J. E. Golay, "Notes on digital coding," *Proc. IEEE*, vol. 37, p. 657, Jun. 1949.

[17] S. Sarangi and S. Banerjee, "Efficient hardware implementation of encoder and decoder for Golay code," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, to be published.

[18] K. Namba, S. Pontarelli, M. Ottavi, and F. Lombardi, "A single-bit and double-adjacent error correcting parallel decoder for multiple-bit error correcting BCH codes," *IEEE Trans. Device Mater. Rel.*, vol. 14, no. 2, pp. 664–671, Jun. 2014.

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

[20] E. R. Berlekamp, "Decoding the Golay code," Jet Propulsion Lab., Pasadena, CA, USA, Tech. Rep. 32-1526, 1971, pp. 81–84.