A Single-Bit and Double-Adjacent Error Correcting Parallel Decoder for Multiple-Bit Error Correcting BCH Codes

Kazuteru Namba, *Member, IEEE*, Salvatore Pontarelli, Marco Ottavi, *Senior Member, IEEE*, and Fabrizio Lombardi, *Fellow, IEEE*

***Abstract*—This paper presents a novel high-speed BCH decoder that corrects double-adjacent and single-bit errors in parallel and serially corrects multiple-bit errors other than double-adjacent errors. Its operation is based on extending an existing parallel BCH decoder that can only correct single-bit errors and serially corrects double-adjacent errors at low speed. The proposed decoder is constructed by a novel design and is suitable for nanoscale memory systems, in which multiple-bit errors occur at a probability comparable to single-bit errors and double-adjacent errors occur at a higher probability (nearly two orders of magnitude) than other multiple-bit errors. Extensive simulation results are reported. Compared with the existing scheme, the area and delay time of the proposed decoder are on average 11% and 6% higher, but its power consumption is reduced by 9% on average. This paper also shows that the area, delay, and power overheads incurred by the proposed scheme are significantly lower than traditional fully parallelized BCH decoders capable of correcting any double-bit errors in parallel.**

***Index Terms*—Error correcting code (ECC), double-adjacent error correction (DAEC), BCH codes, parallel decoder.**

I. Introduction

ERROR control codes (also known as error correcting codes, ECCs) have been frequently used to improve the dependability of a memory system [1], [2]. However, the dependability of a memory system still remains a concern due to neutron-induced single event upsets (SEUs) [3] and the occurrence of multiple-bit errors. Maiz *et al.* [4] have reported that 1–5% of SEUs cause the change of data in multiple cells. Furthermore, Ibe *et al.* [5] have provided simulation evidence that nearly half of the SEUs change the contents of multiple cells at a feature size of 22 nm. Therefore, ECCs dealing with multiple-bit errors are becoming more and more important. The BCH code is one of the best-known and widely used multiple-bit error correcting codes [1], [2]. Multiple-bit error correction of a BCH code needs a low-speed serial decoding process. BCH codes can be decoded faster by parallelizing the serial operations [6], [7], but parallelization incurs in a large hardware overhead, particularly for long information bit length. Moreover, it is well known that the BCH code is less efficient for short information bit lengths [8]. There are few multiple-bit error correcting codes that can be decoded in parallel, e.g., product codes and some low-density parity-check (LDPC) codes, such as orthogonal Latin square (OLS) codes [1], Euclidean geometry LDPC (EG-LDPC) codes [9] and difference-set cyclic codes (DSCC) [10]. However, they require longer check bits than BCH codes. To resolve these issues, Wilkerson *et al.* [8] have proposed a high speed decoding scheme for the BCH code. This scheme utilizes parallel decoding when no error or a single-bit error occurs, and serial decoding when multiple-bit errors occur. As single-bit errors occur more often, this scheme achieves high-speed decoding for most errors. However, at nanoscale feature sizes, multiple-bit errors occur with a significantly high probability due to the high integration density of these memories. Reviriego *et al.* [11] and Wang [12] have presented BCH decoders improving on the decoder of Wilkerson *et al.* [8]. Reviriego's design [11] is a parallel decoder (similar to [8]) that detects but does not correct single errors in parallel. Wang has proposed a decoder that is smaller than Wilkerson's, but it is only suitable for hierarchical double-error correcting (HDEC) codes, not for BCH codes. HDEC codes have a worse code rate than BCH, so Wang's decoder [12] requires a larger memory for the check bits than Wilkerson's decoder [8]. Hence in many cases, it will incur in a larger hardware overhead because the area of the additional memory is significantly larger than the decoder.

An *adjacent error* is a specific type of multi-bit error that changes (or flips) the contents of several adjacent cells. Adjacent errors are caused by a particle hitting a memory array that releases enough energy to affect the value of multiple adjacent cells. This effect occurs with a higher probability than other multiple-bit errors [13]–[15]. Radaelli *et al.* in [15] have experimentally shown that double-adjacent errors occur at a higher probability than other multiple-bit errors by nearly two orders of magnitude. In general, the occurrence of adjacent errors is mitigated by utilizing an interleaving scheme, such that adjacent cells keep values in different words [15]. However, interleaving schemes are not always possible due to design features, such as negative impact on floorplanning, access time, and/or power consumption. Moreover, interleaving requires a long check bit length. So, error correction requires both ECCs

Manuscript received July 9, 2013; revised November 29, 2013, January 15, 2014, and February 19, 2014; accepted February 28, 2014. Date of publication March 5, 2014; date of current version June 3, 2014.

K. Namba is with the Graduate School of Advanced Integration Science, Chiba University, Chiba 263-8522, Japan (e-mail: namba@ieee.org).

S. Pontarelli is with the National Inter-University Consortium for Telecommunications (CNIT), 43124 Parma, Italy (e-mail: pontarelli@ing.uniroma2.it).

M. Ottavi is with the University of Rome "Tor Vergata," 00173 Rome, Italy (e-mail: ottavi@ing.uniroma2.it).

F. Lombardi is with the Department of Electrical and Computer Engineering, Northeastern University, Boston, MA 02115 USA (e-mail: lombardi@ece.neu.edu).

Digital Object Identifier 10.1109/TDMR.2014.2309935

TABLE I
NUMBER OF CORRECTABLE ERRORS FOR (n, k)-BCH CODES

Code function	number of correctable errors	information bit length(k)			
		256	512	1,024	2,048
SEC-DAEC	$2n$–1	547	1,063	2,091	4,143
SEC	n	274	532	1,046	2,072
DEC	$\binom{n}{2}$	37,401	141,246	546,535	2,145,556

with high error correction capabilities as well as interleaving (even if a high degree interleaving is possible [16]). Multiple-bit error correcting codes such as BCH codes can be used for this purpose. However, this multiple-bit error correction incurs in a considerable overhead (for example, in the complex hardware required for the decoders). Research on adjacent error correcting (AEC) codes and in particular double-adjacent error correcting (DAEC) codes has been pursued in the technical literature [14]–[18]. Table I summarizes different numbers of errors that can be corrected by different (n, k) configurations of BCH codes. Note that the number of double-adjacent errors is $n - 1$, which is almost the same as the number of single-bit errors, n [19], and significantly less than the number of double-bit errors, given by $\binom{n}{2}$.[1] So as a first estimate, the hardware penalty incurred for double-adjacent error correction (DAEC) should be comparable to single-bit error correction, and moreover, it should be significantly less than double-error correction [14]. It is also well-known that the odd-weight column SEC-DED (single-error correcting, double-error detecting) code (also referred as the Hsiao code) [20] can be used as an SEC-DAEC code [2]. Recently, a few DAEC codes have been proposed [16], [18]. Generic multi-bit errors (not necessarily double-adjacent errors) occur at a lower probability than double-adjacent error by two orders of magnitude [15], but they still cannot be ignored. Some SEC-DAEC codes can be provided with the capability of detecting multiple-bit errors in addition to double-adjacent errors [14], [16], [21]. However, these codes are not capable of correcting multiple-bit errors, so they are of limited use at nano scales.

This paper presents a high-speed BCH decoder. The proposed decoder resembles Wilkerson's design [8] with high-speed single-bit error correction. Wilkerson's decoder corrects single-bit errors in parallel and multiple-bit errors serially. Instead, we propose a decoder that corrects single-bit errors and double-adjacent errors in parallel and corrects other multiple-bit errors serially. In the proposed decoding scheme, the error pattern generator for SEC is also used for error generation of double-adjacent errors. So, the area for the proposed decoder is comparable to Wilkerson *et al.* [8]. Extensive simulation results are provided to substantiate the viability of the proposed design.

This paper is organized as follows: Section II reviews the BCH code and the existing high-speed decoding scheme [8]. The proposed decoding scheme is described in detail in Section III. Section IV presents the evaluation of the proposed decoding scheme. Section VI concludes this manuscript.

[1]Let u and v be a code word and a received word. Its error pattern e is $v - u$. If no error occurs, $e = (0 \ldots 0)$. For single-bit errors $e = (10 \ldots 0)$, $(010 \ldots 0), (0010 \ldots 0), \ldots, (0 \ldots 01)$. The number of single-bit error patterns is equal to the codeword length n. Similarly, for adjacent errors $e = (110 \ldots 0), (0110 \ldots 0), (00110 \ldots 0), \ldots, (0 \ldots 011)$. Their number is $n - 1$. The number of double-bit errors is given by $\binom{n}{2}$ and is equal to the number of combinations of two bits in a codeword.

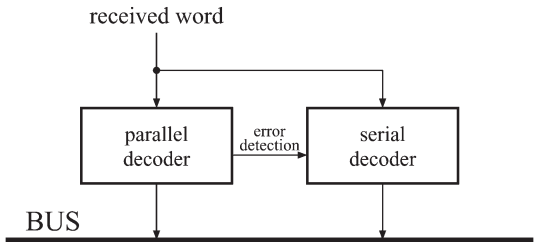


Fig. 1. Diagram of the Wilkerson's BCH decoder.

II. REVIEW OF BCH CODES

BCH (Bose, Chaudhuri, Hocquenghem) codes are one of the most well known binary multiple-error detecting and correcting codes. The BCH code is a cyclic code, and can be decoded serially. However, the high-speed parallel decoding of a BCH code incurs in a large hardware overhead. Jang *et al.* [6] have proposed a BCH decoding scheme in which only some operations are partially parallelized, and overall, it is slower than fully parallelized decoders. Chen *et al.* [7] have shown a fully parallelized BCH decoder. However, the parallelization of BCH decoders for long information bit length requires a significant overhead in hardware. Chen *et al.* [7] have provided evaluation results only at an information bit length of $k = 256$. In a subsequent section of this manuscript, extensive evaluation results are given for larger values of k. They confirm the difficulty of parallelizing high speed decoding for large values of k. This is a major concern, because BCH codes are more efficient at a long information bit length.

Wilkerson *et al.* [8] have presented a high speed BCH decoding scheme in which code words are decoded in parallel if no error or a single bit error occurs and they are decoded serially only if multiple-bit errors occur. As the probability of occurrence of a multiple-bit error is lower than a single-bit error, this decoding scheme represents a good compromise, because it achieves high-speed operation for the most likely cases of error occurrence. Fig. 1 shows the block diagram of Wilkerson's BCH decoder. It consists of a parallel decoder and a serial decoder. The parallel decoder decodes the received word. When the parallel decoder detects multiple-bit errors, it generates an error signal that starts the operation of the serial decoder. The parallel decoder detects multiple-bit errors in a single clock cycle, and the serial decoder requires n iterations for an (n, k) BCH code to find the error location (when using the Berlekamp–Massey algorithm).

While the algorithm of the serial decoder is conventional, the parallel decoder shows significant originality. The following is an H matrix of a t-bit error correcting and $(t + 1)$-bit error detecting BCH code:

$$H = \begin{bmatrix} H_{\text{parity}} \\ H_1 \\ H_3 \\ \vdots \\ H_{2t-1} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ \alpha & \alpha^2 & \alpha^3 & \cdots & \alpha^n \\ \alpha^3 & \alpha^6 & \alpha^9 & \cdots & \alpha^{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \alpha^{2t-1} & \alpha^{2(2t-1)} & \alpha^{3(2t-1)} & \cdots & \alpha^{n(2t-1)} \end{bmatrix}$$

where α is a primitive root.

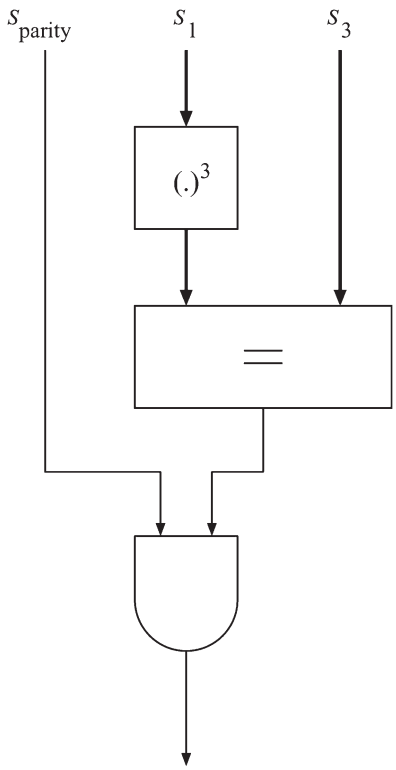


Fig. 2. Single-bit error detector for double error correcting BCH code.

Let $s = (s_{\text{parity}}\ s_1\ s_3 \ldots s_{(2t-1)})$ be a syndrome where s_{parity} and s_j $(j = 1, \ldots, 2t-1)$ correspond to H_{parity} and H_j in H, i.e., $s_{\text{parity}} = vH^T_{\text{parity}}$ and $s_j = vH^T_j$, where v is the received word. If a single-bit error occurs at the i-th bit, the syndrome s is as follows:

$$s_{\text{parity}} = 1 \tag{1}$$

$$\begin{aligned} s_1 &= \alpha^i \\ s_j &= \alpha^{ij} \\ &= (s_1)^j. \end{aligned} \tag{2}$$

The occurrence of a single-bit error is detected by verifying the validity of (1) and (2). Note that (1) and (2) do not include the variable i. So, verification needs to be established only once for every syndrome. Fig. 2 shows the circuit for detecting the occurrence of single-bit errors using a double-bit error correcting BCH code. If $s_{\text{parity}} = 0$ (i.e., (1) is false), the AND gate outputs a 0 regardless of the output value of the comparator. If $s_3 \neq (s_1)^3$ (i.e., (2) is false), then the comparator outputs a 0 and the AND gate outputs a 0. If and only if both (1) and (2) are true, then the AND gate outputs a 1, so detecting the occurrence of a single-bit error.

Since no two columns of H_1 are identical, error location of a single-bit error is established by comparing s_1 and all columns in H_1. The comparison of every column in H must be performed to find the error location and only s_1 (and not the entire syndrome) must be compared. Therefore, r_1-input (not r-input) gates are required, where r_1 and r are the number of rows in H_1 and H, respectively. Therefore, this scheme reduces the area overhead and delay time of the decoder compared to a general decoder design that must compare the entire syndrome.

III. Proposed Scheme

This section presents the parallel SEC-DAEC decoder for the BCH code. The proposed decoder corrects double-adjacent as well as single-bit errors. The double-adjacent error is a special case of a double-bit error, in which the values at two adjacent bits (i.e., i-th and $(i+1)$-th bits) erroneously change. For example, $(110\ldots0)$ is an error pattern of a double-adjacent error; $(1010\ldots0)$ is not an error pattern of a double-adjacent error.

The outline of the proposed decoder resembles the high-speed decoder of [8] (shown in Fig. 1). The proposed scheme however differs from [8] in the algorithm and construction of the parallel decoder. If a double-adjacent error occurs at the i-th and $(i+1)$-th bits, the syndrome appears as

$$s_{\text{parity}} = 0 \tag{3}$$

$$\begin{aligned} s_1 &= \alpha^i + \alpha^{(i+1)} \\ &= \alpha^i(\alpha + 1) \end{aligned} \tag{4}$$

$$\begin{aligned} s_j &= \alpha^{ij} + \alpha^{(\alpha+1)j} \\ &= \left(\frac{s_1}{\alpha+1}\right)(\alpha^j + 1). \end{aligned} \tag{5}$$

The occurrence of a double-adjacent error is detected by verifying that (3) and (5) are valid. In addition, the location of the error is established from (4). s_{parity} is the signal indicating the error type for diagnostic purposes: $s_{\text{parity}} = 0$ for double-adjacent errors, while $s_{\text{parity}} = 1$ for single-bit errors. If there is no error, then $s_{\text{parity}} = 0$ and $s_1 = s_j = 0$.

Next, it is shown that the detection of a double-adjacent error is possible by verifying that (3) and (5) are true. The syndrome of a double-adjacent error differs from that of another correctable or detectable error due to the conditions in the ECC. In a traditional non-shortened BCH code, H_1 consists of all non-zero column vectors; any non-zero vector can appear as s_1. In addition $s_1 = 0$ when no error occurs. Therefore, every syndrome satisfying these two equations appears as a correct codeword, or as a double-adjacent error; the syndrome of any correctable and detectable error except double-adjacent errors does not satisfy these two equations.

Fig. 3 illustrates an example of the construction of the parallel decoder for double-error correcting BCH codes. It consists of a syndrome generator, an error pattern generator and an error detector.

- The syndrome generator generates the syndrome $s = (s_{\text{parity}}\ s_1\ s_3)$ from a received word v.
- The error pattern generator generates the error pattern e, and the decoder then outputs $v + e$ as a decoded word.
- The error detector detects uncorrectable errors, i.e., errors that are neither single-bit nor double-adjacent.

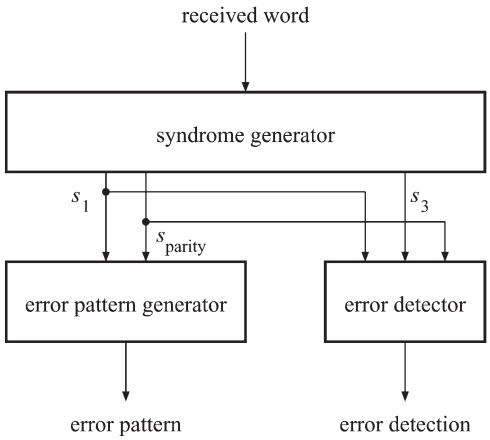


Fig. 3. Diagram of parallel decoder for double error correcting BCH code.

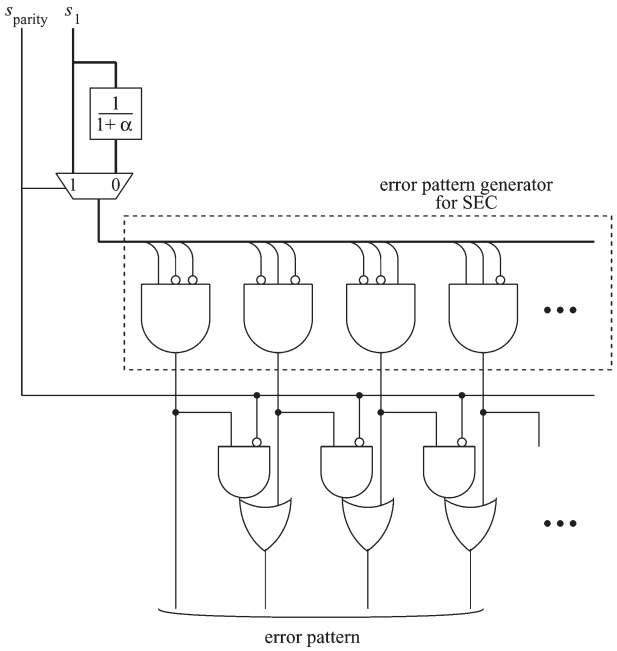


Fig. 4. Error pattern generator for the proposed parallel decoder.

Fig. 4 shows the structure of the error pattern generator for the proposed parallel decoder. It includes the error pattern generator for single-error correction that outputs the correct error pattern if s_1 is received as input for a single-bit error. Specifically, if α^i is the input, the output vector $e = (e_0, \dots e_{(k-1)})$ satisfies the following conditions: $e_i = 1$ and $e_j = 0$ for $i \neq j$. The input value of the generator for SEC is given as follows:

$$\begin{cases} s_1 & (s_{\text{parity}} = 1) \\ \frac{1}{1+\alpha} s_1 & (s_{\text{parity}} = 0). \end{cases}$$

The input value of the generator for SEC is α^i as per (4), and so the generator for SEC outputs e, such that $e_i = 1$ and $e_j = 0$ for $i \neq j$ for double-adjacent errors on the i-th and $(i+1)$-th bits.

The decoder also includes AND–OR gates. The output vector of the generator for SEC $e = (e_0, \dots e_{(k-1)})$ and the output of the AND–OR gates, i.e., the output vector of the entire error pattern generator $e' = (e'_0, \dots e'_{(k-1)})$, satisfies the following condition:

$$e'_j = \begin{cases} e_j & (s_{\text{parity}} = 1) \\ e_j + e_{j-1} & (s_{\text{parity}} = 0). \end{cases}$$

Note that for SEC-DAEC decoding, the error pattern can be computed by checking if the syndrome vector corresponds to either h_i, the i-th column of the H matrix (for the case of SEC), or the sum of two adjacent columns h_{i-1} and h_i, or h_i and h_{i+1} (for the case of DAEC). This "standard" SEC-DAEC decoder requires to compute a comparison against $3 \times n$ columns ($3 \times n$ AND gates with $n - k$ inputs and n 3-inputs OR gates), while the proposed decoder reuses the same circuitry for locating the erroneous bit(s) for both the SEC and DAEC cases, thus reducing the complexity of the decoder. A comparison between this "standard" SEC-DAEC decoder and the one proposed in this section is included in the evaluation section.

Next, a few examples are provided for the operation of the circuit in Fig. 4.

- As a first example, consider a single-bit error occurring on the first bit. A syndrome is given by $s = (1\ \alpha\ \alpha^3)$. Since $s_{\text{parity}} = 1$, the MUX outputs $s_1 = \alpha$. The error pattern generator for SEC outputs $e = (10 \dots 0)$; as the output of the entire error pattern generator, the AND–OR gates generate $e' = e$.
- As a second example, consider a double-adjacent error occurring on the first and second bits. The syndrome is given by $s = (0(\alpha + \alpha^2)(\alpha^3 + \alpha^6))$. As $s_{\text{parity}} = 0$, the MUX outputs $s_1/(1+\alpha) = \alpha$. The error pattern generator for the SEC generates as output $e = (10 \dots 0)$, and thus the entire error pattern generator outputs $e' = (100 \dots 0) + \overline{s_{\text{parity}}}(010 \dots 0) = (110 \dots 0)$.
- As a third example, consider the case when no error occurs; then, $s_{\text{parity}} = 0$, and the MUX outputs $s_1/(1+\alpha) = 0$. The error pattern generator for SEC outputs $e = (0 \dots 0)$, while the entire error pattern generator outputs $e' = (0 \dots 0) + \overline{s_{\text{parity}}}(0 \dots 0) = (0 \dots 0)$.

Fig. 5 shows the design in block form of the uncorrectable error detector. This circuit verifies (2) and (5). This detector includes two comparators, namely the left and right comparators for verifying (2) and (5) respectively. The detector outputs a (detection) signal selecting either of the comparators according to s_{parity}. For example, consider the case in which a single-bit error occurs on the 0th bit. The syndrome is given by $s = (1\ \alpha\ \alpha^3)$. The inputs of the left comparator in Fig. 5 are $s_1^3 = \alpha^3$ and $s_3 = \alpha^3$. The left comparator outputs a zero (i.e., the two inputs match). Since $s_{\text{parity}} = 1$, the MUX selects the output of the left comparator as the output of the detector, and thus, the detector outputs a 0 (i.e., no uncorrectable error is detected). As a further example, consider a double-bit error occurring on the 0th and second bits. The syndrome is now given by $s = (0(\alpha + \alpha^3)(\alpha^3 + \alpha^9))$. This error is not correctable. So, the error pattern generator in Fig. 5 outputs the wrong error patterns. The inputs of the right comparator are $s_1^3/(1+\alpha)^3$ and $s_3/(1+\alpha^3)$. For example, assume that the minimal polynomial is given by $m_1(x) = x^3 + x + 1$. The two inputs are equal to α^5 and α^4 and therefore, the comparator outputs a 1 (the two

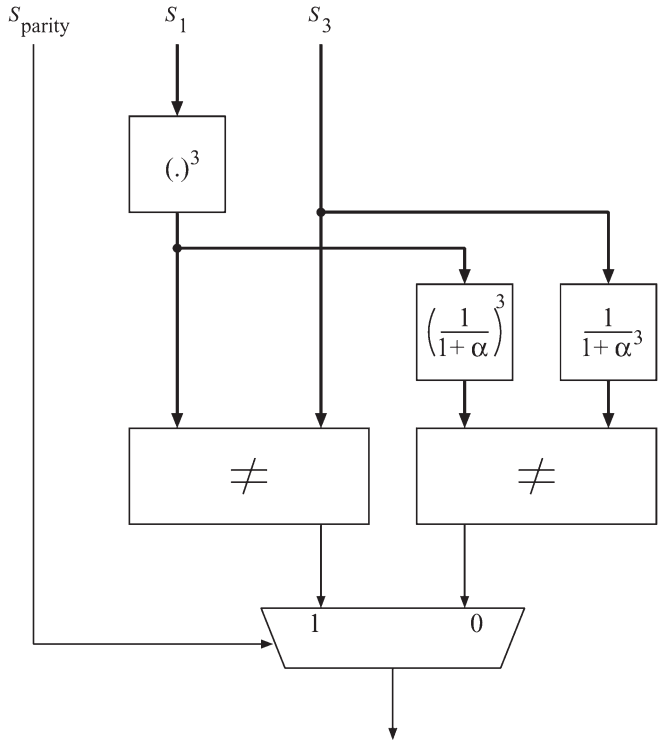


Fig. 5. Uncorrectable error detector for double error correcting BCH code.

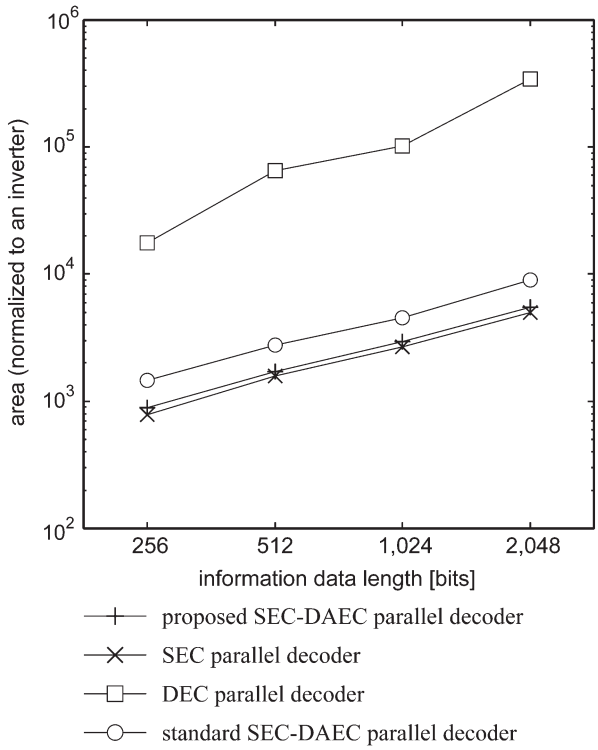


Fig. 6. Area of parallel decoders (normalized to an inverter).

inputs do not match for any $m_1(x)$ due to the features of the BCH code.) Since $s_{\text{parity}} = 0$, the MUX selects the output of the right comparator as the output of the detector. Therefore, the detector outputs a 1; the error is detected to be uncorrectable and the serial decoder of Fig. 1 is activated.

IV. Evaluation

This section evaluates the proposed parallel decoder and compares it with the decoder of [8] for double-bit error correcting (DEC) BCH codes with information bit lengths of $k = 256, 512, 1024,$ and 2048. In addition, the proposed scheme is also compared with the double-bit error correcting BCH parallel decoder of [7], whose design is considered the best among all double-bit error correcting BCH parallel decoders as well as for multiple-error correcting BCH parallel decoders found in the technical literature. All the decoders evaluated in this paper are designed using Verilog-HDL (RTL-level) and synthesized by using the Synopsys Design Compiler. These circuits are combinational and the presented evaluation considers as figures of merit area, power consumption and gate depth (delay time) normalized to those of an inverter (thus making it feature size independent). They are denoted by the ratios of A_C/A_I, P_C/P_I and D_C/D_I where A_C and A_I are the areas of the evaluated circuit and an inverter, P_C and P_I are the power consumptions of the evaluated circuit and an inverter, D_C and D_I are the delay time of the evaluated circuit and an inverter. Note that P_I and D_I have been obtained by connecting the output of the inverter under consideration to another inverter.

The following simulation-based results confirm that, as briefly discussed in Section 1 in terms of many figures of merit, double-adjacent error correction is *comparable* to single-bit error correction, and therefore it incurs in *significantly less* penalties than double-error correction.

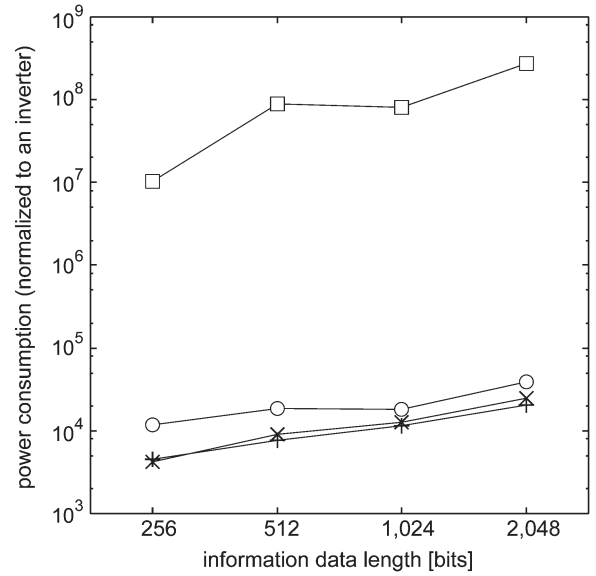


Fig. 7. Power consumption of parallel decoders (normalized to an inverter).

- Fig. 6 shows the area of the parallel decoders. The proposed decoder has an area comparable to a SEC decoder and significantly less than a DEC decoder. Our proposed SEC-DAEC decoder achieves an area saving of 37.6% compared to the "standard" SEC-DAEC decoder.
- Fig. 7 shows the power consumption of the proposed as well as the SEC and DEC schemes. Our SEC-DAEC procedure allows a power saving of 51.2% when compared

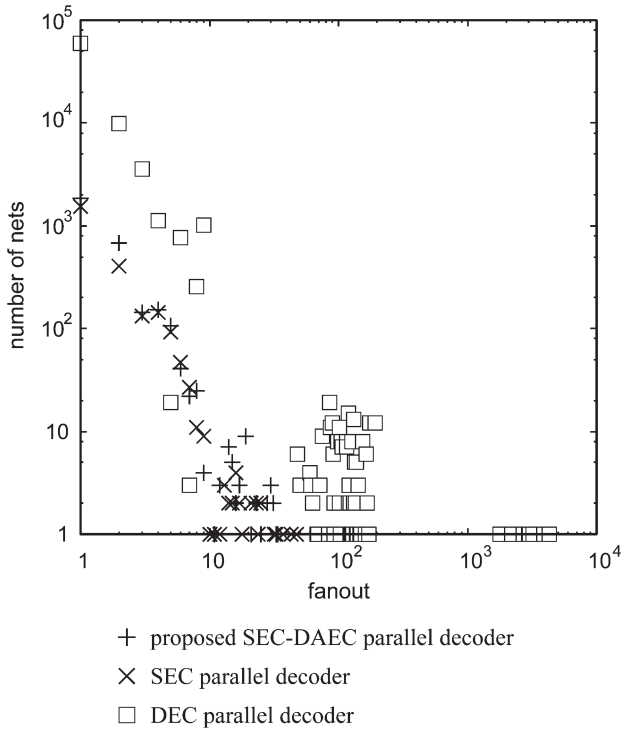


Fig. 8. Fanout versus number of nets for information bit length of 256 bits.

to the "standard" SEC-DAEC decoding. Compared to the area analysis, the difference between the DEC and the other three schemes is large. This is due to the high-fanout nets between the syndrome generators and the error pattern generator. This is shown in Fig. 8 in which a point is plotted at (x, y) to denote that the number of nets with a fanout of x gates is given by y. This figure shows that the DEC decoder has many high-fanout nets (i.e., a fanout of nearly 10^2), unlike the SEC and SEC-DAEC decoders.

- There is no comparison result for the DEC decoder in terms of gate depth (delay time), because the Design Compiler could not find the delay time of DEC due to the high-fanout nets required in this design.

A detailed comparison between the proposed SEC-DAEC and existing SEC decoders has also been pursued.

- Fig. 9 shows the area of the parallel decoders. The area of the proposed decoder is on average 11% larger than the existing decoder of [8]. Hence, the 11% increase in area allows correcting double-adjacent errors at high speed.
- Fig. 10 shows the dissipation due to switching power (power for charging or discharging of the output load external to every gate) and internal power (power dissipated in gates due to charging of internal loads and the short-circuit current between activated N and P transistors). The switching power consumption is increased by 6% on average; however, the internal power is reduced by 18% on average. This occurs due to the high-fanout and the corresponding larger internal power dissipation. The proposed decoder has a more complex design construction than [8], but it also requires a lower high-fanout. Hence, the total dissipation is reduced by 8%.

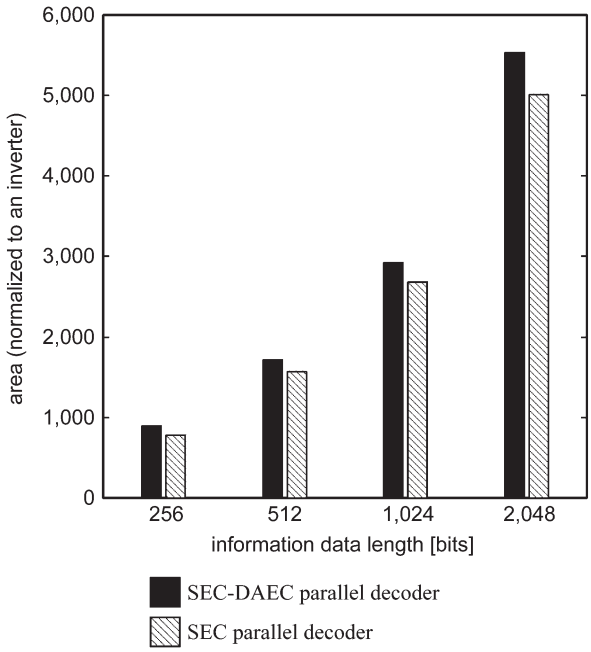


Fig. 9. Area of parallel decoders (normalized to an inverter) (comparison to SEC).

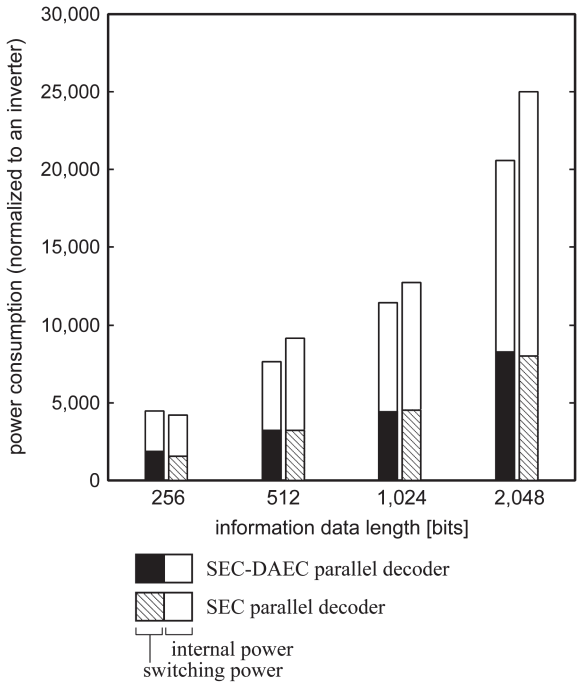


Fig. 10. Power consumption of parallel decoders (normalized to an inverter) (comparison to SEC).

- Fig. 11 shows the delay time. The gate depth of the proposed decoder is on average 6% larger than for the decoder design of [8] and on average 4% larger than for the "standard" SEC-DAEC decoder.

Consider the decoding time reduction technique in which the syndrome is calculated first and then the received data is output without error correction if the syndrome is all zeros, i.e., no error occurs [11]. The proposed method can be used also with this technique. Fig. 12 shows the gate depth for detecting the

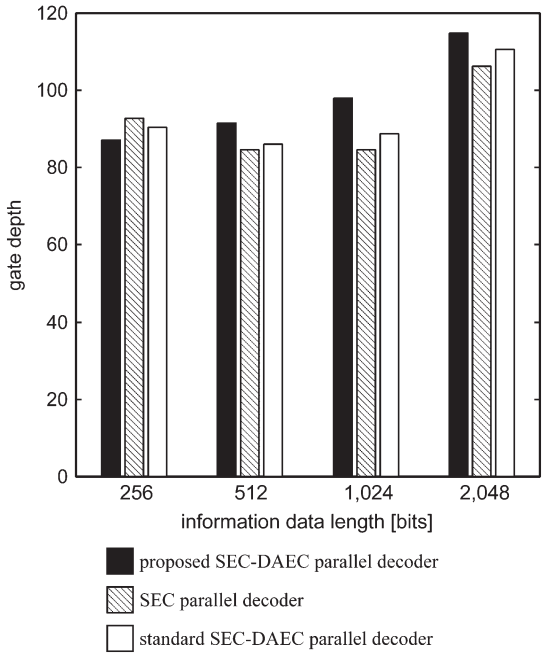


Fig. 11. Gate depth of parallel decoders (delay time normalized to an inverter) (comparison to SEC).

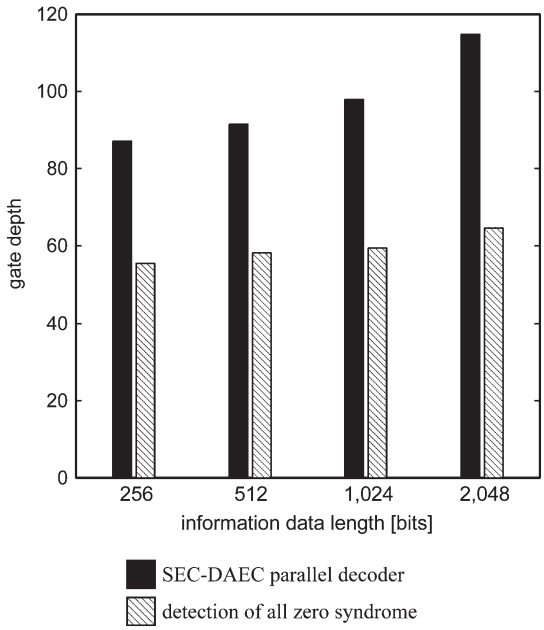


Fig. 12. Gate depth of proposed parallel decoder (delay time normalized to an inverter) (comparison to detection of all zero syndrome).

all-zeros syndromes for BCH codes as well as the proposed decoding scheme. The use of this technique with the proposed scheme reduces the gate depth by almost half for the error-free cases.

V. Conclusion

This paper has presented a high-speed BCH decoder for correcting double-adjacent as well as single-bit errors in parallel. The proposed decoder resembles Wilkerson's parallel BCH decoder [8] that can correct only single-bit errors. The decoding scheme of [8] operates serially (and hence at low speed) when a multiple-bit error occurs (including a double-adjacent error). High-speed correction of double-adjacent errors is included in the proposed scheme, because double-adjacent and single-bit errors are much more frequent in a memory system.

The proposed scheme corrects double-adjacent errors in parallel by using a novel decoder design. The hardware overhead of the proposed decoder is comparable to the decoder of [8]. In particular, its power consumption is 9% lower than the decoder of [8] although the proposed decoder is capable of parallel DAEC decoding (unlike [8]). The power saving is due to the number of high-fanout nodes (that are power hungry) that are less than those required for the Wilkerson's decoder. The proposed scheme incurs in overheads that are significantly lower than traditional fully parallelized BCH decoders capable of correcting any double-bit errors in parallel. The area and power consumption of the proposed decoder is 3.87×10 and 8.55×10^3 smaller than those of the best double-bit error correcting BCH parallel decoder presented in the literature [7].

References

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

[2] E. Fujiwara, *Code Design for Dependable Systems: Theory and Practical Applications*. Hoboken, NJ, USA: Wiley-Interscience, 2006.

[3] M. Violante, L. Sterpone, A. Manuzzato, S. Gerardin, P. Rech, M. Bagatin, A. Paccagnella, C. Andreani, G. Gorini, A. Pietropaolo, G. Cardarilli, S. Pontarelli, and C. Frost, "A new hardware/software platform and a new 1/E neutron source for soft error studies: Testing FPGAs at the ISIS facility," *IEEE Trans. Nucl. Sci.*, vol. 54, no. 4, pp. 1184–1189, Aug. 2007.

[4] J. Maiz, S. Hareland, K. Zhang, and P. Armstrong, "Characterization of multi-bit soft error events in advanced SRAMs," in *IEDM Tech. Dig.*, 2003, pp. 21.4.1–21.4.4.

[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] S.-C. Jang, J.-H. Lee, W.-C. Lee, and K.-R. Cho, "Design of parallel BCH decoder for MLC memory," in *Proc. IEEE Int. SoC Des. Conf.*, 2008, pp. III-46–III-47.

[7] Y.-H. Chen, C.-H. Yang, and H.-C. Chang, "A fully-parallel step-by-step BCH decoder over composite field for NOR flash memories," in *Proc. IEEE Int. Symp. VLSI Des. Autom. Test*, 2012, pp. 1–4.

[8] C. Wilkerson, A. R. Alameldeen, Z. Chishti, W. Wu, D. Somasekhar, and S. Lu, "Reducing cache power with low-cost, multi-bit error-correcting codes," in *Proc. Annu. Int. Symp. Comp. Archit.*, 2010, pp. 83–93.

[9] H. Naeimi and A. DeHon, "Fault secure encoder and decoder for memory applications," in *Proc. IEEE Int. Symp. Defect Fault Tolerance VLSI Syst.*, 2007, pp. 409–417.

[10] 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.

[11] P. Reviriego, C. Argyrides, and J. A. Maestro, "Efficient error detection in double error correction BCH codes for memory applications," *Microelectron. Reliab.*, vol. 52, no. 7, pp. 1528–1530, Jul. 2012.

[12] Z. Wang, "Hierarchical decoding of double error correcting codes for high speed reliable memories," in *Proc ACM/EDAV/IEEE Des. Autom. Conf.*, 2013, pp. 1–7.

[13] 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 Dev. Lett.*, vol. 21, no. 6, pp. 310–312, Jun. 2000.

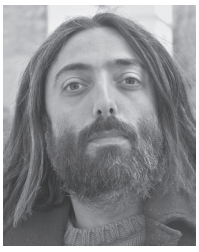
[14] A. Dutta, "Low cost adjacent double error correcting code with complete elimination of miscorrection within a dispersion window for multiple bit upset tolerant memory," in *Proc. IEEE/IFIP Int. Conf. VLSI-SoC*, 2012, pp. 287–290.

[15] D. Radaelli, H. Puchner, S. Wong, and S. Daniel, "Investigation of multi-bit upsets in a 150 nm technology SRAM device," *IEEE Trans. Nucl. Sci.*, vol. 52, no. 6, pp. 2433–2437, Dec. 2005.

[16] 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.*, 2007, pp. 349–354.
[17] A. Neale and M. Sachdev, "A new SEC-DED error correction code subclass for adjacent MBU tolerance in embedded memory," *IEEE Trans. Device Mater. Reliab.*, vol. 13, no. 1, pp. 223–230, Mar. 2013.
[18] P. Reviriego, S. Pontarelli, J. A. Maestro, and M. Ottavi, "Low-cost single error correction multiple adjacent error correction codes," *IET Electron. Lett.*, vol. 48, no. 23, pp. 1470–1472, Nov. 2012.
[19] N. M. Abramson, "A class of systematic codes for non-independent errors," *IRE Trans. Inf. Theory*, vol. 5, no. 4, pp. 150–157, Dec. 1959.
[20] M. Hsiao, "A class of optimal minimum odd-weight-column SEC-DED codes," *IBM J. Res. Dev.*, vol. 14, no. 4, pp. 395–401, Jul. 1970.
[21] M. Richter, K. Oberlaenderz, and M. Goessel, "New linear SEC-DED codes with reduced triple bit error miscorrection probability," in *Proc. IEEE Int. On-Line Testing Symp.*, 2008, pp. 37–42.

Kazuteru Namba (M'04) received the B.E., M.E., and Ph.D. degrees from Tokyo Institute of Technology, Yokohama, Japan, in 1997, 1999, and 2002, respectively.

In 2002, he joined Chiba University, Chiba, Japan, where he is currently an Assistant Professor with the Graduate School of Advanced Integration Science. His current research interests include dependable computing. He is a member of the IEICE and the IPSJ.



Salvatore Pontarelli received the Master's degree in electronic engineering from the University of Bologna, Bologna, Italy, in 2000 and the Ph.D. degree in microelectronics and telecommunications from the University of Rome "Tor Vergata," Rome, Italy, in 2003.

He was with the National Research Council (CNR) and the Italian Space Agency (ASI) and has been a Consultant for various Italian and European companies for projects related to digital design and fault tolerance in digital systems. He is currently a Research Consultant with the National Inter-University Consortium for Telecommunications (CNIT), Parma, Italy. His research activities are mainly focused on the development of highly reliable systems, error detection and correction codes, fault detection and recovery for arithmetic circuits, and hardware for networking applications.

Marco Ottavi (M'03–SM'10) received the Ph.D. degree in microelectronics and telecommunications engineering from the University of Rome "Tor Vergata," Rome, Italy, in 2004.

He is currently an Associate Professor with the same university, and from 2009 to 2013, he was the recipient of a "Rientro dei Cervelli" Fellowship awarded by the Italian Ministry of Education, University and Research. Previously, he was a Senior Design Engineer with AMD, Boxborough, MA, USA, and held postdoctoral positions with Sandia National Laboratories, Albuquerque, NM, USA, and with the Department of Electrical and Computer Engineering, Northeastern University, Boston, MA. His research interests include yield and reliability modeling, test, design for testability, fault-tolerant architectures, and online testing and design of nanoscale circuits and systems. In these fields, he published five book chapters and more than 50 articles on archival journals and peer-reviewed conferences. Since December 2011, he has been the Chair of COST Action IC1103 "Manufacturable and Dependable Multicore Architectures at Nanoscale" (MEDIAN).

Fabrizio Lombardi (M'81–SM'02–F'09) received the B.Sc. (Hons.) degree in electronic engineering from the University of Essex, Colchester, U.K., in 1977; the Master's degree in microwaves and modern optics and the Diploma in microwave engineering from University College London, London, U.K., both in 1978; and the Ph.D. degree from the University of London, London, in 1982.

He is currently the holder of the International Test Conference Endowed Chair Professorship at Northeastern University, Boston, MA, USA. His research interests are bioinspired and nanomanufacturing/computing, VLSI design, testing, and fault/defect tolerance of digital systems. He has extensively published in these areas and coauthored/edited seven books.