

# Short Paper

## A Method to Construct Low Delay Single Error Correction Codes for Protecting Data Bits Only

Pedro Reviriego, Salvatore Pontarelli, Juan Antonio Maestro, and Marco Ottavi

**Abstract**—Error correction codes (ECCs) have been used for decades to protect memories from soft errors. Single error correction (SEC) codes that can correct 1-bit error per word are a common option for memory protection. In some cases, SEC codes are extended to also provide double error detection and are known as SEC-DED codes. As technology scales, soft errors on registers also became a concern and, therefore, SEC codes are used to protect registers. The use of an ECC impacts the circuit design in terms of both delay and area. Traditional SEC or SEC-DED codes developed for memories have focused on minimizing the number of redundant bits added by the code. This is important in a memory as those bits are added to each word in the memory. However, for registers used in circuits, minimizing the delay or area introduced by the ECC can be more important. In this paper, a method to construct low delay SEC or SEC-DED codes that correct errors only on the data bits is proposed. The method is evaluated for several data block sizes, showing that the new codes offer significant delay reductions when compared with traditional SEC or SEC-DED codes. The results for the area of the encoder and decoder also show substantial savings compared to existing codes.

**Index Terms**—Double error detection, error correction codes (ECCs), single error correction (SEC), soft errors.

### I. INTRODUCTION

Soft errors are an important issue for electronic circuits, and many different techniques are used to mitigate their effects [1]. To protect memories, error correction codes (ECCs) are widely used [2]. ECCs have an impact on circuit delay, area, and power consumption. The delay is added as data has to be encoded when writing into the memory and decoded when reading from it. The impact on area and power comes from the encoding and decoding circuitry and also from the redundant bits that the ECC adds to each data block. For memories, the number of redundant bits is typically the most critical factor as those bits are added to each memory word. This means that the impact on area scales with memory size.

Single error correction (SEC) codes are the ones most commonly used to protect standard memories and circuits [2],

Manuscript received June 29, 2012; revised September 7, 2012; accepted October 19, 2012. Date of current version February 14, 2013. This work was supported in part by the Spanish Ministry of Science and Education under Grant AYA2009-13300-C03. The work of M. Ottavi was funded by the Italian Ministry for University and Research, Program Incentivazione alla mobilità di studiosi stranieri e italiani residenti all'estero, D.M. 96, 23.04.2001. This paper is part of a collaboration in the framework of COST ICT Action 1103 "Manufacturable and Dependable Multicore Architectures at Nanoscale." This paper was recommended by Associate Editor J. Cortadella.

P. Reviriego and J. A. Maestro are with the Universidad Antonio de Nebrija, Madrid 28040, Spain (e-mail: previrie@nebrija.es; jmaestro@nebrija.es).

S. Pontarelli and M. Ottavi are with the University of Rome Tor Vergata, Rome 00133, Italy (e-mail: pontarelli@ing.uniroma2.it; ottavi@ing.uniroma2.it).

Digital Object Identifier 10.1109/TCAD.2012.2226585

while more sophisticated codes are used in critical applications such as space [3]. The main reason for this is that SEC codes can be encoded or decoded with simple circuitry and require a low number of redundant bits. SEC codes can correct a single bit error per block. For memory protection, SEC codes are commonly extended to also detect double errors. In this case, the codes are known as SEC double error detection (SEC-DED) codes [4].

A classical type of SEC codes is Hamming codes that can be constructed in a simple way [5]. Hamming codes can also be extended with a parity bit to obtain a SEC-DED code. More sophisticated SEC-DED codes have also been proposed as, for example, the ones described in [4], [6], and [7]. In those cases, the number of redundant bits is kept to the lowest achievable value, and the different codes optimize the area and delay of the encoder and decoder or the detection of triple errors. This is a reasonable approach for memories as the number of redundant bits has a direct impact on memory size.

As technology scales, soft errors also become an issue for registers used in digital circuits. SEC codes can also be used to protect those registers that may store, for example, the state of a finite state machine or data-path values in an arithmetic circuit. In those cases, the design constraints for the SEC code are different than in memories. For example, in circuits minimizing the encoding and decoding delay may be the most critical aspect. The impact on area is also different, as the redundant bits are only added to the register being protected such that their cost can be lower than that of the encoding and decoding circuitry. In memories, since the redundant bits are added to each word, their overall cost commonly has the largest impact on area. Another difference is that in memories it can be useful to correct errors on the parity bits when decoding. This is the case, for example, when scrubbing is used to periodically remove errors to prevent their accumulation [8]. In registers, the correction of parity bits has little interest as the register contents are in many cases updated frequently and the input data comes from other circuit elements. In spite of these differences, the SEC or SEC-DED codes that have been used or designed to protect memories are also used in registers. This clearly suggests the interest of designing SEC codes that are targeted to the needs of registers.

In this paper, a method to construct SEC and SEC-DED codes that have low encoding and decoding delay is proposed. The proposed scheme can be used to design codes for any data block size using a simple script. To illustrate the benefits of the method, the derived SEC codes are compared to Hamming SEC codes and the proposed SEC-DED codes with the optimized SEC-DED codes presented in [6]. The results show that they achieve a significantly lower delay and also a lower area for the encoder and the decoder. The proposed codes require, in most cases, more redundant bits than traditional codes. This limits their applications to large memories, but is not an issue for registers where the area of the encoder and decoder can be

larger than that of the register itself. Therefore, the proposed method can be used to design SEC or SEC-DED codes that are tailored for the protection of registers in circuits. Another application of the proposed codes is the protection of high-speed memories or caches in which speed is more important than area.

The rest of this paper is organized as follows. Section II describes single error correction codes analyzing, in detail, the encoder and decoder. In Section III, the proposed method to construct low delay SEC and SEC-DED codes is presented. In Section IV, the derived codes are evaluated in terms of area and delay and compared with existing codes. Finally, conclusions of this paper and some ideas for future work are summarized in Section V.

## II. SINGLE ERROR CORRECTION CODES

A linear block code takes  $k$  data bits and produces an  $n$ -bit block [9]. In many applications, systematic codes that preserve the original  $k$  data bits and simply add  $n-k$  parity bits are preferred. A given linear block code can be described by its generator matrix  $G$ . Given a block of  $k$  data bits, the  $n$  bits codeword is obtained by multiplying the data block by the generator matrix. As an example, the generator matrix for an SEC Hamming code for  $k = 8$  and  $n = 12$  is shown in (1). The last four columns define the added parity bits. The generator matrix is used to encode the data block

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}. \quad (1)$$

To decode a codeword, the parity check matrix  $H$  is used. This matrix when multiplied by a codeword will be an all zero vector if there are no errors. If there is an error, the value of that vector, usually called syndrome, will serve to detect the error and correct it. The  $H$  matrix for the SEC Hamming code previously considered is shown in (2). It can be observed that all columns in the matrix are different. This means that any single-bit error will produce a different syndrome, and therefore the error can be corrected

$$H = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (2)$$

The structure of the encoder and decoder can be explained using the  $G$  and  $H$  matrixes. Encoding is simply computing the multiplication of the input data block by the  $G$  matrix. This requires a number of XOR gates for each column in  $G$  that is proportional to the number of ones in that column. Decoding starts by multiplying the  $H$  matrix by the codeword. This requires a number of XOR gates for each row in  $H$  that is proportional to the number of ones in that row. Then, the

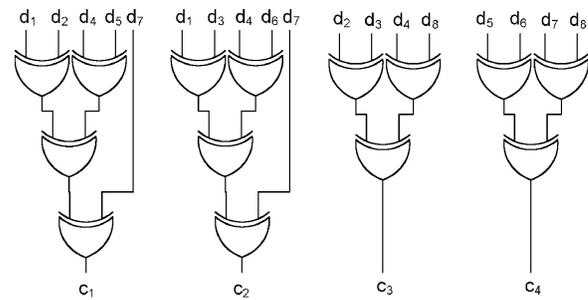


Fig. 1. Encoder for the SEC Hamming code with  $k = 8$ .

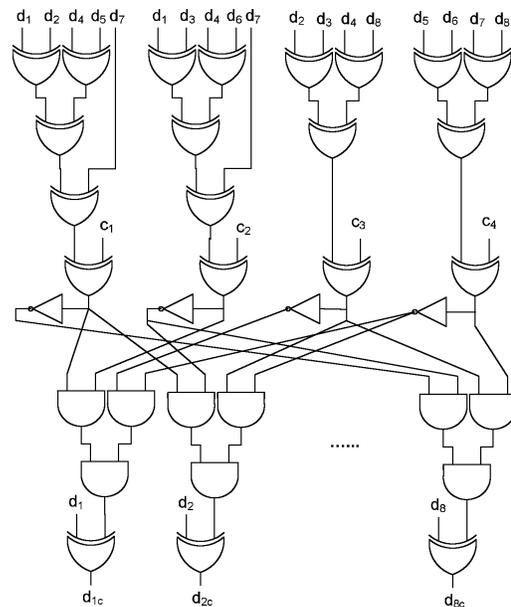


Fig. 2. Structure of the decoder for the SEC Hamming code with  $k = 8$ .

obtained syndrome must be checked against every column in  $H$  and if there is a match; that is the bit in error that is then corrected. Each of those checks requires an  $n - k$  input AND gate. The encoder for the Hamming code used as an example is illustrated in Fig. 1. The data bits ( $d_i$ ) are the inputs and the parity check bits ( $c_i$ ) the outputs.

The structure of the decoder is shown in Fig. 2. In this case, the data bits ( $d_i$ ) and the parity check bits ( $c_i$ ) are the inputs, and the outputs are the corrected data bits ( $d_{ic}$ ). It can be observed that the complexity and delay is larger in the case of the decoder as it is normally the case for most ECCs.

The number of parity bits required by a Hamming code grows logarithmically with the data block size, and the values for common block sizes are shown in Table I. These values are the same for other SEC codes, and as discussed before are an important parameter when the codes are used in memories.

SEC-DED codes are similar to SEC codes and can be obtained by using a parity check matrix  $H$  with an odd number of ones (odd weight) in all its columns [4]. This reduces the number of combinations that can be used in the columns, and therefore increases the number of additional bits required. This can be observed in Table I, where the parity check bits for traditional SEC-DED codes are illustrated. The encoding and decoding is similar to that of SEC codes with the addition of

TABLE I  
NUMBER OF PARITY CHECK BITS IN EXISTING SEC  
AND SEC-DED CODES

$k$	SEC (Hamming)	SEC-DED [4], [6]
8	4	5
16	5	6
32	6	7
64	7	8

TABLE II  
NUMBER OF ONE ELEMENTS IN THE H MATRIX FOR DIFFERENT CODES

$k$	Hamming SEC	Proposed SEC	Hsiao SEC-DED	[7] SEC-DED	Proposed SEC-DED
8	22	21	29	27	29
16	43	39	54	52	54
32	87	73	103	104	103
64	186	140	216	216	201

some logic to detect double errors. This logic simply performs the OR of the  $n - k$  syndrome bits and also the XOR of those bits. A double error is detected when the OR takes a value of one (at least one syndrome bit is different from zero, therefore there are errors) and the XOR a value of zero (an even number of syndrome bits are different from zero, i.e., more than one error has occurred).

The number of ones in the parity check matrix is related to the number of XOR gates needed to generate and check the parity check bits. Therefore, it can be used as a first estimate of the complexity of the encoders and decoders [6]. The values for the different codes and block sizes are provided in Table II. The proposed SEC and proposed SEC-DED are the values for the codes that will be presented in the remainder of this paper. It can be observed that SEC codes have less ones than SEC-DED codes as they are simpler. For the Hamming codes, columns have been reordered to ensure that the ones with the lowest weights are used for the data bits. The SEC-DED codes with the lowest number of ones are those presented in [6], and therefore they will be used as the reference for comparison in the following. It is important to note that for the decoder, the number of ones gives only a rough idea of the complexity as the logic to identify the syndrome values that is independent of the number of ones is the most complex block.

### III. CONSTRUCTION OF LOW DELAY SINGLE-ERROR CORRECTION CODES

The proposed method to construct SEC and SEC-DED codes tries to minimize the number of ones in each row and in each column of the  $H$  matrix. Reducing the number of ones in the rows lowers the delay when computing the parity bits in the encoder and also when recomputing the parity checks in the decoder. Reducing the number of ones in the columns of the  $H$  matrix does not lower the delay by itself. To achieve a reduction in the delay, the final phase of decoding is modified. This is done by checking only for the bits that are one in each column to correct the corresponding bit.

For this modification to work, this checking must be sufficient to uniquely identify the column of  $H$  corresponding to

the bit affected by the error. This condition is satisfied if no column of  $H$  includes all the ones present in another column.

For data bits, this can be achieved for example, if all the data bits have the same number of ones  $w$  in their column of the  $H$  matrix. Then, as the columns are different, no column can include all the ones in another column as that would imply that the two columns are equal. To minimize the number of ones, the value  $w = 2$  can be used to obtain SEC codes. It is also interesting to analyze the case  $w = 3$  as in that case the code is SEC-DED.

Since for the parity bits the columns have only a one, the condition is not met as other columns have a one in that bit. Therefore, this modification cannot correct errors in the parity bits. This is not an issue for registers as the correction of parity bits is not normally needed as discussed before. The decoder modification combined with a low number of ones in the columns of the  $H$  matrix results in an additional reduction of the decoding delay.

The method to construct the code starts by finding the smallest value of  $n - k$  for which the following is true:

$$\binom{n-k}{w} \geq k. \quad (3)$$

For  $w = 2$ , this value can be found analytically by solving (3) that is a quadratic equation in  $n$ . As the value of  $n$  has to be larger than  $k$ , only one of the two possible solutions of the equation is valid in our case. The value of  $n - k$  obtained is

$$n - k \geq \left\lceil \frac{1 + \sqrt{8k + 1}}{2} \right\rceil \quad (4)$$

that shows a growth of the number of parity bits with the square root of  $k$  that is larger than the logarithmic growth of Hamming codes. This means that as  $k$  increases, the overhead of the proposed codes in terms of the number of additional parity bits compared to Hamming will also increase.

Similarly, for  $w = 3$ , the solution to (3) is given by

$$n - k \geq \left( \frac{\sqrt{243k^2 - 1}}{3^{3/2}} + 3k \right)^{1/3} + \frac{1}{3 \left( \frac{243k^2 - 1}{3^{3/2}} + 3k \right)^{1/3}} + 1 \quad (5)$$

that, as  $k$  is larger than one, can be approximated by

$$n - k \geq (6k)^{1/3} + 1. \quad (6)$$

The growth of the number of parity check bits with  $k$  is smaller than for  $w = 2$ , but is still larger than the logarithmic growth of traditional SEC-DED codes.

In the second step to constructing the codes, a different combination of  $w$  of the  $n - k$  added bits is used for each of the first  $k$  columns of the  $H$  matrix. Equation (3) guarantees that there are sufficient different combinations. The remaining  $n - k$  columns form an identity matrix of size  $n - k$ . An  $H$  matrix constructed using this procedure for  $w = 2$  and  $k = 8$  is shown in (7). Compared with the matrix in (2), it can be observed that the number of parity bits ( $n - k$ ) is five instead of four. However, the maximum number of ones in any row is five compared with six in (2). The number of ones is two

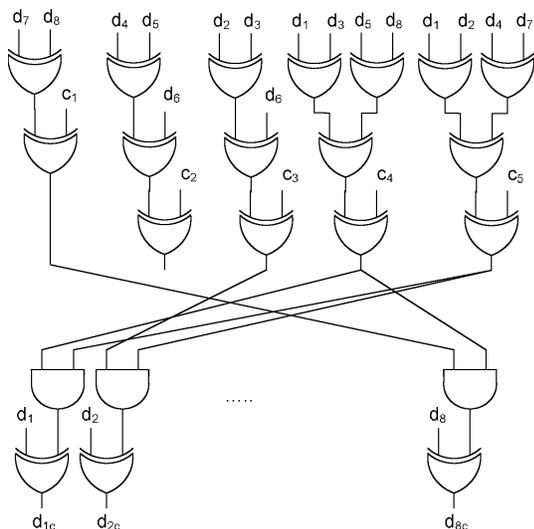


Fig. 3. Structure of the decoder for the proposed SEC code with  $k = 8$ .

TABLE III

NUMBER OF PARITY BITS IN THE PROPOSED SEC AND SEC-DED CODES

$k$	SEC (Weight Two)	SEC-DED (Weight Three)
8	5	5
16	7	6
32	9	7
64	12	9

in every column compared to some columns with three ones in the Hamming matrix. This reduction in the number of ones enables a lower encoding and decoding delay. The proposed decoder is illustrated in Fig. 3. Compared with the one in Fig. 2, it can be observed that the logic depth is significantly smaller. The reduction in the upper part comes from having less ones in the rows of  $H$ . The reduction in the lower part comes from having only two ones in the columns of  $H$  and using the modified decoding to correct errors.

$$H = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

In a general case, a Hamming code will have rows with a number of ones that is roughly  $k/2$ . This compares with the proposed SEC codes ( $w = 2$ ) for which the number of ones in a row is by design at most  $n - k - 1$ . Similarly, to locate an error a traditional SEC code requires an  $n - k$  input AND gate compared with a simple two input AND gate in the proposed code. In practical implementations, this results in a significant reduction of the encoding and decoding delays, as discussed in the next section. The number of parity bits is, however, larger than for traditional SEC codes. Table III illustrates for different values of  $k$  the number of parity check bits required in the proposed scheme. Those can be directly compared with the values for the Hamming codes in Table I.

For the proposed SEC-DED codes ( $w = 3$ ), a three-input AND gate is needed to locate an error compared with the  $n - k$

TABLE IV  
DELAY ESTIMATES FOR THE SEC ENCODER AND DECODER

$k$	Hamming		Proposed SEC		Reduction	
	Enc	Dec	Enc	Dec	Enc	Dec
8	0.27	0.42	0.23	0.37	14.8%	11.9%
16	0.33	0.49	0.28	0.40	15.2%	18.4%
32	0.39	0.56	0.30	0.45	23.1%	19.6%
64	0.45	0.65	0.35	0.48	22.2%	26.1%

input AND gate of traditional codes. The number of parity check bits required is the same as existing SEC-DED codes for small values of  $k$ , as can be seen comparing Tables I and III. The number of ones in the parity check matrix is also the same as in Hsiao SEC-DED codes [4] for small values of  $k$  (see Table II). For larger values of  $k$ , reductions in the number of ones in the parity check matrix are obtained at the expense of additional parity check bits. One interesting observation is that Hsiao codes for small values of  $k$  have a weight of three in all the data bits. Therefore, the proposed optimized error location scheme can be used to reduce the delay of the decoders when we are only interested in correcting errors on the data bits.

One distinct feature of the proposed codes is that they correct errors on the data bits only. This is similar to other codes such as orthogonal Latin square (OLS) codes [10]. However, in OLS codes, each pair of data bits participates in at most one shared parity check bit to ensure that majority logic decoding can be used. This is different from the proposed scheme in which the goal is to ensure that no data bit participates in all the parity check bits, in which another data bit participates. This is then used to simplify the location and correction of an error, as described before. Another difference is that OLS codes are commonly used when multiple error correction capabilities are needed although SEC can also be implemented. The main issues with SEC OLS codes are that they are only implemented for a few block sizes and require a large number of parity check bits.

Finally, it is worth mentioning that the parity check matrixes of the proposed codes are similar to that of low density parity check (LDPC) codes commonly used in communication systems [11]. Nevertheless, since LDPC codes usually have large block size, and must provide multiple error correction, the encoding and decoding procedures are very different from our proposed codes and require complex logic circuitry [11].

#### IV. EVALUATION

To evaluate the benefits of the proposed codes in practical implementations, the method has been used to design SEC and SEC-DED codes for the values of  $k$  in Table II. Then, the encoders and decoders have been implemented in HDL. The designs have then been synthesized for the 45 nm OSU FreePDK Standard Cell Library [12] using Synopsys Design Compiler. The results of the proposed SEC codes are compared with those of an SEC Hamming code. For SEC-DED, the proposed codes are compared with the optimized SEC-DED codes recently proposed in [6]. In all cases, the decoders only correct errors in the data bits.

TABLE V  
AREA ESTIMATES FOR THE SEC ENCODER AND DECODER

$k$	Hamming		Proposed SEC		Reduction	
	Enc	Dec	Enc	Dec	Enc	Dec
8	84.5	416.3	71.3	306.9	15.6%	26.3%
16	177.4	630.7	168.9	504.6	4.8%	20.0%
32	399.4	918.4	321.0	762.1	19.6%	17.0%
64	855.1	1982.8	685.6	1559.5	19.8%	21.4%

TABLE VI  
DELAY ESTIMATES FOR THE SEC-DED ENCODER AND DECODER

$k$	SEC-DED [6]		Proposed SEC-DED		Reduction	
	Enc	Dec	Enc	Dec	Enc	Dec
8	0.29	0.45	0.29	0.41	0.0%	8.9%
16	0.36	0.54	0.32	0.50	11.1%	7.4%
32	0.46	0.61	0.37	0.58	19.6%	4.9%
64	0.47	0.65	0.42	0.61	10.6%	6.2%

TABLE VII  
AREA ESTIMATES FOR THE SEC-DED ENCODER AND DECODER

$k$	SEC-DED [6]		Proposed SEC-DED		Reduction	
	Enc	Dec	Enc	Dec	Enc	Dec
8	143.1	415.7	113.6	355.8	20.6%	14.4%
16	247.3	694.6	227.1	563.7	8.2%	19.8%
32	464.4	1060.2	455.2	927.3	2.0%	12.5%
64	814.2	1904.8	905.7	1881.4	-11.2%	1.2%

The delay estimates (in ns) and area estimates (in  $\mu\text{m}^2$ ) for both the encoder (Enc) and the decoder (Dec) SEC codes are shown in Tables IV and V. For delay, significant reductions over a Hamming code are achieved that in some cases exceed 25%, confirming the low delay of the proposed SEC codes. For area, significant savings are also obtained in most cases. This means that the use of the proposed codes may also be more cost effective than Hamming codes, since the area savings in the encoder and decoder can outweigh the cost of the additional flip-flops needed by our code.

For SEC-DED codes, the area and delay estimates are presented in Tables VI and VII. It is important to note that the delay results for the decoders are for the correction of data bits. The delay for double error detection is larger in most cases. However, double error detection will only be used to signal an unrecoverable error, and therefore it only has to be smaller than the clock cycle. The correction of data bits is followed by the actual circuit logic, and therefore adds directly to the circuit delay. Therefore, the impact on circuit delay is due to the correction of the data bits, and it makes sense to report it. The results also show delay reductions compared with the SEC-DED codes proposed in [6] although smaller than in the case of the SEC codes. The area is also reduced in most cases. The impact on the delay of the proposed ECCs when used in a circuit would be that of the encoder plus the decoder. For SEC codes and  $k = 16$  bits that is 0.75 ns while

for SEC-DED codes the value goes up to 0.82 ns. This means that for a 250 MHz circuit approximately 20% of the clock will be devoted to the ECC protection. For circuits that have a higher clock rate, there will be parts of the circuit that have significant timing margin, and therefore the ECCs can still be used to protect those parts of the circuits while the faster triplication with voting can be used on the critical paths.

## V. CONCLUSION

In this paper, a method to construct low delay SEC and SEC-DED codes was presented. The proposed method used some additional parity bits to reduce the number of ones in the rows and columns of the parity check matrix. This reduction was then used to simplify the decoding logic to achieve lower delay and area. The proposed method was evaluated and compared with traditional SEC Hamming and SEC-DED codes, showing significant reductions in both area and delay. The proposed codes can be useful to protect registers in circuits where the area and delay of the encoder and decoder can be a more important issue than the number of parity bits. The codes can also be useful to protect high-speed memories or caches as they can minimize delay at the expense of a few additional parity check bits.

## ACKNOWLEDGMENT

The authors would like to thank V. Gherman, S. Evain, and Y. Bonhomme for providing the  $H$  matrixes of the codes presented in [6].

## REFERENCES

- [1] M. Nicolaidis, "Design for soft error mitigation," *IEEE Trans. Device Mater. Reliab.*, vol. 5, no. 3, pp. 405–418, 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, 1984.
- [3] G. C. Cardarilli, M. Ottavi, S. Pontarelli, M. Re, and A. Salsano, "Fault tolerant solid state mass memory for space applications," *IEEE Trans. Aerospace Electron. Syst.*, vol. 41, no. 4, pp. 1353–1372, Oct. 2005.
- [4] M. Y. Hsiao, "A class of optimal minimum odd-weight column SEC-DED codes," *IBM J. Res. Develop.*, vol. 14, pp. 395–301, Jul. 1970.
- [5] R. W. Hamming, "Error detecting and error correcting codes," *Bell Syst. Tech. J.*, vol. 29, pp. 147–160, Apr. 1950.
- [6] V. Gherman, S. Evain, N. Seymour, and Y. Bonhomme, "Generalized parity-check matrices for SEC-DED codes with fixed parity," in *Proc. IEEE On-Line Testing Symp.*, Jul. 2011, pp. 198–20.
- [7] M. Richter, K. Oberlaender, and M. Goessel, "New linear SEC-DED codes with reduced triple bit error mis-correction probability," in *Proc. IEEE On-Line Testing Symp.*, Jul. 2008, pp. 37–42.
- [8] A. M. Saleh, J. J. Serrano, and J. H. Patel, "Reliability of scrubbing recovery-techniques for memory systems," *IEEE Trans. Reliab.*, vol. 39, no. 1, pp. 114–122, Apr. 1990.
- [9] S. Lin and D. J. Costello, *Error Control Coding*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 2004.
- [10] 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.
- [11] G. Li, I. J. Fair, and W. A. Krzymien, "Low-density parity-check codes for space-time wireless transmission," *IEEE Trans. Wirel. Commun.*, vol. 5, no. 2, pp. 312–322, Feb. 2006.
- [12] J. E. Stine *et al.*, "FreePDK: An open-source variation-aware design kit," in *Proc. IEEE Int. Conf. MSE*, Jun. 2007, pp. 173–174.