Low-cost single error correction multiple adjacent error correction codes

P. Reviriego, S. Pontarelli, J.A. Maestro and M. Ottavi

Soft errors that affect flip-flops are a major issue in advanced electronic circuits. As technology scales, multiple bit errors become more likely. This limits the applicability of traditional protection techniques like triplication with voting or single error correction codes that can correct only one error. Multiple errors tend to affect adjacent bits, and therefore it is interesting to use error correction codes that can correct adjacent errors. The issue with these codes is that they require a large area and delay that limits their use to protect flip-flops in circuits. Presented are codes that can be implemented with low area and delay and can correct multiple adjacent errors.

Introduction: Soft errors are a major issue in electronic circuits. A soft error changes the logical value of one or more flip-flops and can be caused for example by radiation particles. A number of techniques can be used to mitigate the effects of soft errors [1]. Among these, triplication with voting is commonly used to protect flip-flops. However, as technology scales, soft errors that affect more than one flip-flop become common [2]. The flip-flops affected by the errors are those placed physically close in the circuit [3]. These multiple errors compromise the effectiveness of triplication as they are likely to affect two of the replicated flip-flops. One option is to modify the placing of the flip-flops to ensure that the three replicas of a given flip-flop are placed physically apart. This implies modifications to the standard circuit design flow and will increment the interconnection densities, thus increasing the area of the circuit. Another option is to use error correction codes that can correct multiple adjacent errors. One example is single error correction double adjacent error correction (SEC-DAEC) codes [4]. The main issues when these codes are used to protect flip-flops are their area and delay overheads. These overheads are due to the encoding and decoding logic and also to the flip-flops needed to store parity check bits. The added delay limits the use of such codes in high-speed circuits where delay is critical.

In this Letter, low-cost single error correction multiple adjacent error correction (SEC-MAEC) codes are presented. The proposed codes are also evaluated and compared with triplication with voting and with existing SEC-DAEC codes [4]. The results show that the proposed codes have lower area and delay than existing SEC-DAEC codes. The area is similar to that of triplication with voting and the delay is larger. However, triplication with voting cannot correct multiple adjacent errors and therefore is not suitable when multiple bit errors are present.

Proposed codes: The codes are better explained by describing the encoding and decoding processes. For a block of k flip-flops (d_i) the code computes k parity check bits (p_i) as follows:

$$p_i = d_i \oplus d_{mod(i-s,k)} \tag{1}$$

where s is a parameter that will determine the ability of the code to correct multiple adjacent errors. The decoding starts by computing the syndrome:

$$s_i = p_i \oplus d_i \oplus d_{mod(i-s,k)} \tag{2}$$

Then each of the data bits is decoded by computing:

$$d_i' = (s_i \wedge s_{mod(i+s,k)}) \oplus d_i \tag{3}$$

where d_i is the bit stored in the flip-flop and d_i' the decoded bit.

From the above equations it can be observed that the encoding and decoding logic is very simple. More precisely, for each data bit four *xor* gates and one *and* gate are needed. The placement of the bits in the block is assumed to be $d_0, d_1, \ldots, d_{k-1}, p_0, p_1, \ldots, p_{k-1}$. As an example, the encoder, decoder and bit placement for $k = 8$ and $s = 2$ are illustrated in Fig. 1, while Fig. 2 shows the corresponding G and H matrixes.

The codes can correct single errors as an error on data bit d_i and will cause bits s_i and $s_{mod(i+s,k)}$ to be one such that according to (3) the error is corrected. On the other hand, an error on the parity check bits (p_i) will affect only one syndrome bit (s_i) and therefore according to (3) no miscorrection will take place.

In addition to single errors, the codes can correct bursts of b adjacent errors where b is a function of k and s. This can be proved as follows: considering an error that affects bit d_i, the error will be corrected unless any of the two parity bits that check that bit, p_i and $p_{mod(i+s,k)}$ are affected by an error. That can occur if any of those parity bits are affected or the other bits checked by those checks are affected. In the first case, since parity bits are at a distance of k, s or $k+s$, of bit d_i, the worst case is s. For example, in Fig. 1, if there is an error on bit d_0, parity check p_0 is at a distance $k = 8$ and p_2 is at a distance $k + s = 10$ of bit d_0. For bit d_6, parity check p_6 is at a distance $k = 8$ and p_0 is at a distance $s = 2$ of bit d_6.

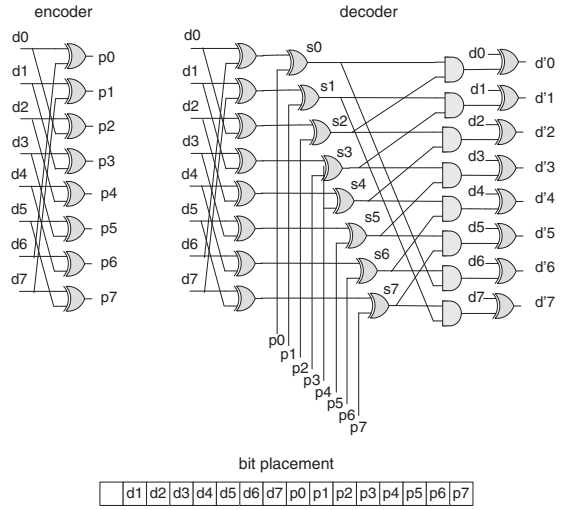

Fig. 1 *Encoder and decoder for proposed code with $k = 8$ and $s = 2$*

$$G = \begin{bmatrix} 1000000010100000 \\ 0100000001010000 \\ 0010000000101000 \\ 0001000000010100 \\ 0000100000001010 \\ 0000010000000101 \\ 0000001010000010 \\ 0000000101000001 \end{bmatrix} \qquad H = \begin{bmatrix} 1000001010000000 \\ 0100000101000000 \\ 1010000000100000 \\ 0101000000010000 \\ 0010100000001000 \\ 0001010000000100 \\ 0000101000000010 \\ 0000010100000001 \end{bmatrix}$$

Fig. 2 *Generating matrix G and parity matrix H for proposed code with $k = 8$ and $s = 2$*

To ensure that these errors do not occur for a burst of maximum size b we need $s \geq b$. In the second case, the other data bits that are checked by p_i and $p_{mod(i+s,k)}$ are at a distance of either s or $k - s$ from bit d_i. Therefore we need to ensure that $k - s \geq b$ and $s \geq b$. Following the previous example of an error on d_0, the other bits checked by p_0 and p_2 are d_6 and d_4 which are at a distance $k - s = 6$ and $s = 2$ of bit d_0, respectively. Finally, an error on bit d_i can occur when there are errors in parity check bits p_i and $p_{mod(i+s,k)}$ as therefore d_i is miscorrected. That can occur when there are errors on the parity check bits themselves (which are at a distance s) or when there are errors on the other bits checked by p_i and $p_{mod(i+s,k)}$. These bits are at a distance of either s or $k - s$ from the bit d_i but the distance may be in the same direction so that the bits can be at a distance $(k - s) - s$. For example, this occurs for bit d_6 that is checked by p_6 and p_0. The other bits that are checked by those parity check bits are d_0 and d_4 that are at a distance $k - s = 6$ and $s = 2$, respectively, from bit d_6. Therefore, to ensure that these errors do not occur for a burst of maximum size b we need $(k - s) - s \geq b$ or $k \geq 2s + b$, which is the most restrictive condition. In a practical design it is desirable to minimise the value of s to reduce the interconnection cost. Therefore, as $s \geq b$ this is achieved when $s = b$ and the condition becomes $k \geq 3b$. The maximum value for which this condition is true is given by $b = floor(k/3)$.

Table 1: Maximum adjacent burst size that can be corrected

k	$b = s$
8	2
12	4
16	5
24	8
32	10

The maximum values of b for different values of k are summarised in Table 1. It can be observed that the values grow with the number of flip-flops protected k.

Evaluation: To check the effectiveness of the proposed codes, they were implemented in Matlab and tested for the values of k and s shown in Table 1. Random adjacent errors were inserted varying the number of errors from 1 to b. In all cases, for the values of b in Table 1, the errors were corrected. The total number of error combinations tested in each case was one hundred thousand.

To evaluate the area and delay of the proposed codes, they were implemented in HDL and synthesised for a 45nm library [5]. The implementation comprises the flip-flops, the encoder and the decoder. Triplication with voting and existing SEC-DAEC codes [4] were also implemented. The area and delay results are summarised in Tables 2 and 3 in terms of μm^2 and nanoseconds. In the case of SEC-DAEC no code was proposed for $k = 8$ and therefore results are only given for the other values of k. It can be observed that the proposed codes have much lower area and delay than existing SEC-DAEC codes. When the comparison is made with triplication, the cost is only slightly lower and the delay is significantly larger. However, since triplication cannot correct adjacent errors, its use is limited when multiple bit errors are present.

Table 2: Area estimates for different values of k

k	Triplication with voting	SEC-DAEC [4]	SEC-MAEC proposed
8 bits	326.5	n.a.	315.2
16 bits	652.91	1603.1	630.4
32 bits	1305.8	3450.6	1260.8

Table 3: Delay estimates for different values of k

k	Triplication with voting	SEC-DAEC [4]	SEC-MAEC proposed
8 bits	0.10	n.a.	0.18
16 bits	0.10	0.54	0.18
32 bits	0.10	0.64	0.18

Conclusion: In this Letter, low-cost single error correction multiple adjacent error correction (SEC-MAEC) codes have been proposed and evaluated. These codes are of interest when multiple soft errors affect a circuit causing errors in adjacent flip-flops. The results show that when used to protect flip-flops in a circuit their area is lower than that of existing codes and similar to triplication. The delay is also significantly reduced and the ability to correct multiple adjacent errors is increased compared to previous single error correction codes that can also correct adjacent errors.

Acknowledgments: This work was supported by the Spanish Ministry of Science and Education under Grant AYA2009-13300-C03. M. Ottavi is funded by the Italian Ministry for University and Research; Program 'Incentivazione alla mobilità di studiosi stranieri e italiani residenti all'estero', D.M. n.96, 23.04.2001. This Letter is part of a collaboration in the framework of COST ICT Action 1103 'Manufacturable and Dependable Multicore Architectures at Nanoscale'.

© The Institution of Engineering and Technology 2012
13 September 2012
doi: 10.1049/el.2012.3270

P. Reviriego and J.A. Maestro (*Departamento de Ingeniería Informática, Universidad Antonio de Nebrija, C. Pirineos 55, Madrid, Spain*)

E-mail: previrie@nebrija.es

S. Pontarelli and M. Ottavi (*Department of Electronic Engineering, University of Rome, 'Tor Vergata'*)

References

1 Nicolaidis, M.: 'Design for soft error mitigation', *IEEE Trans. Device Mater. Reliab.*, 2005, **5**, (3), pp. 405–418
2 Ibe, E., Taniguchi, H., Yahagi, Y., Shimbo, K., and Toba, T.: 'Impact of scaling on neutron-induced soft error rate in SRAMs from a 250 nm to a 22 nm design rule', *IEEE Trans. Electron Devices*, 2010, **57**, (7), pp. 1527–1538
3 Satoh, S., Tosaka, Y., and Wender, S.A.: 'Geometric effect of multiple-bit soft errors induced by cosmic ray neutrons on DRAMs', *IEEE Electron Device Lett.*, 2000, **21**, (6), pp. 310–312
4 Dutta, A., and Touba, N.A.: 'Multiple bit upset tolerant memory using a selective cycle avoidance based SEC-DED-DAEC code'. 25th IEEE VLSI Test Symp., Berkeley, CA, USA, 2007, pp. 349–354
5 Stine, J.E., *et al.*: 'FreePDK: an open-source variation-aware design kit'. Proc. IEEE Int. Conf. Microelectronic Systems Education, (MSE'07), San Diego, CA, USA, June 2007, pp. 173–174