ELectronics EXpress

LETTER

IEICE Electronics Express, Vol.10, No.6, 1–5

Self-correcting check bit generator of error correction codes for memories

Sanguhn Cha and Hongil Yoon[a)]

School of Electrical and Electronic Engineering, Yonsei University, 134 Shichon-dong, Seodaemun-gu, Seoul 120–749, Korea

a) *hyoon@yonsei.ac.kr*

Abstract: A self-correcting check bit generator (SCO-CBG) is proposed that can correct errors in the check bits produced by faulty check bit generator (CBG). The SCO-CBG amends check bit errors using the concept of error correction code (ECC) techniques. In the SCO-CBG, the parity bits and predicted parity bits function as the check bits of the ECC techniques. The SCO-CBG and the conventional triple modular redundancy (TMR) CBG are implemented using 45 nm library. When compared with the TMR CBG, the SCO-CBG reduces the area overhead and power consumption for 128 data bit word by up to 48.4% and 47.0%, respectively.

Keywords: check bit generator, error correction code, memory

Classification: Integrated circuits

References

[1] N. Gaitanis, "TSC-error C/D circuits for SEC/DED product codes," *IEE Proceedings-E, Computers and Digital Techniques*, vol. 135, no. 5, pp. 253–258, 1988.

[2] F. Aymen, H. Belgacem, and K. Chiraz, "A new efficient self-checking Hsiao SEC-DED memory error correcting code," *Int. Conf. Microelectronics*, pp. 1–5, 2011.

[3] E. Fujiwara, *Code design for dependable systems: theory and practical application*, John Wiley & Sons, Hoboken, New Jersey, 2006.

[4] J. A. Maestro, P. Reviriego, C. Argyrides, and D. K. Pradhan, "Fault tolerant single error correction encoders," *J. Electronic Testing*, vol. 27, no. 2, pp. 215–218, 2011.

[5] M. Nicolaidis, "Design for soft error mitigation," *IEEE Trans. Device Mater. Rel.*, vol. 5, no. 3, pp. 405–418, 2005.

[6] S. Cha and H. Yoon, "High speed, minimal area, and low power SEC code for DRAMs with large I/O data widths," *Proc. IEEE Int. Symp. Circuits and Systems*, pp. 3026–3029, 2007.

1 Introduction

Transient errors due to cosmic neutrons, alpha particles, and radiations have emerged as a key reliability concern in memories. Error correction code (ECC) techniques have been widely used to remedy transient errors in

DOI: 10.1587/elex.10.20130103
Received February 06, 2013
Accepted March 05, 2013
Published March 25, 2013

memory cells; however, transient errors can occur not only in memory cells but also in logic circuits. Particularly, transient errors in the check bit generator (CBG) inhibit memory cell error correction and cause additional errors by mis-correcting words containing no error. As a result, assuring faultless CBG operation is very critical in bringing maximum ECC-protected memory reliability [1, 2, 3, 4, 5].

Self-checking CBGs [1, 2, 3] and triple modular redundancy (TMR) CBGs [4, 5] have previously been presented to provide an error protection in the CBG. The self-checking CBG can detect transient errors of the CBG during normal operations. However, beyond detection, the self-checking CBG cannot correct errors produced. TMR is the most commonly used solution for logic circuit error correction. However, TMR incurs excessive area overhead and power consumption, rendering it unacceptable in most designs [4, 5].

In this letter, a new self-correcting CBG (SCO-CBG) is proposed to self-correct check bit errors induced by faulty CBG using the concept of the ECC techniques. Also, a procedure for constructing three kinds of H-matrices is presented in order to efficiently implement the proposed SCO-CBG. Details regarding the proposed SCO-CBG and three H-matrices are discussed based on the odd-weight-column code [2, 3]. Comparative analyses between the proposed SCO-CBG and TMR CBG are also performed.

2 Self-correcting check bit generator

In the proposed SCO-CBG, check bit errors produced by permanent faults or transient errors in the CBG are corrected using the concept of the ECC techniques. In the ECC techniques, two types of check bits are required to correct the errors. The first type of check bits is generated in the encoding operation and is stored in a memory array; the second type of check bits is generated during the decoding operation. The syndromes are generated by comparing the two types of check bits. Fig. 1 shows a block diagram of the SCO-CBG, which consists of six units: 1) the CBG, 2) the parity generator, 3) the parity predictor, 4) the parity-syndrome generator, 5) the error locator, and 6) the corrector. In the SCO-CBG, the parity bits and predicted parity bits function as the two types of check bits in the ECC techniques. The parity-syndromes are generated by comparing the parity bits and predicted parity bits. If no error exists in the check bits, the value of the parity-syndrome becomes a zero vector; otherwise, the value of the parity-syndrome is a non-zero vector. The parity-syndromes are decoded to determine the locations of the erroneous check bits using the error locator. Finally, the corrector amends the erroneous check bits.

In order to implement the SCO-CBG, three H-matrices are required. The first H-matrix is the $r \times k$ binary matrix $\boldsymbol{H}_c$ which is the H-matrix of the conventional ECCs such as the Hamming code and odd-weight-column code and used by the CBG. The second H-matrix is the $m \times r$ binary matrix $\boldsymbol{H}_p$ used by the parity generator and error locator. The third H-matrix is the $m \times k$ binary matrix $\boldsymbol{H}_{pp}$ used by the parity predictor. The numbers of the check bits, data bits, and parity bits are denoted as r, k, and m, respectively. And the vectors with r check bits, k data bits, m parity bits, m predicted parity bits are denoted as $\boldsymbol{c}=(c_0\ c_1\ \cdots\ c_{r-1})$, $\boldsymbol{d}=(d_0\ d_1\ \cdots\ d_{k-1})$, $\boldsymbol{p}=(p_0\ p_1\ \cdots\ p_{m-1})$, and $\boldsymbol{pp}=(pp_0\ pp_1\ \cdots\ pp_{m-1})$, respectively. The check bits,

DOI: 10.1587/elex.10.20130103
Received February 06, 2013
Accepted March 05, 2013
Published March 25, 2013

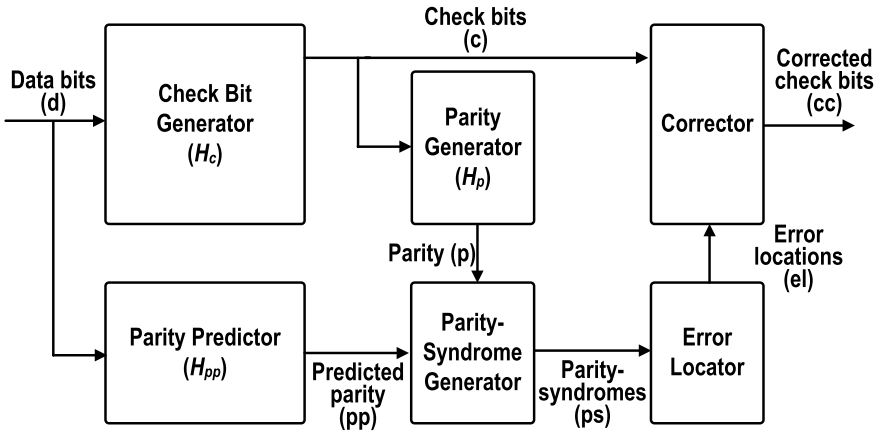

Fig. 1. Block diagram of the self-correcting check bit generator

parity bits, and predicted parity bits are computed by the relations $\boldsymbol{c}=\boldsymbol{d}\cdot\boldsymbol{H}_c^{\boldsymbol{T}}$, $\boldsymbol{p}=\boldsymbol{c}\cdot\boldsymbol{H}_p^{\boldsymbol{T}}$, and $\boldsymbol{pp}=\boldsymbol{d}\cdot\boldsymbol{H}_{pp}^{\boldsymbol{T}}$, respectively, where the superscript $\boldsymbol{T}$ denotes the transpose of the matrix. The vectors with m parity-syndromes, r error locations, and r corrected check bits are indicated by $\boldsymbol{ps}=(ps_0\ ps_1\ \cdots\ ps_{m-1})$, $\boldsymbol{el}=(el_0\ el_1\ \cdots\ el_{r-1})$, and $\boldsymbol{cc}=(cc_0\ cc_1\ \cdots\ cc_{r-1})$, respectively. The parity-syndromes are generated by the relation $\boldsymbol{ps}=\boldsymbol{p}\text{\^{}}\boldsymbol{pp}$, the error locations are decoded by the relation $\boldsymbol{el}=\boldsymbol{ps}\cdot\boldsymbol{H}_p$ and the check bit errors are corrected by the relation $\boldsymbol{cc}=\boldsymbol{c}\text{\^{}}\boldsymbol{el}$, where ^ is the exclusive-or operator. In the SCO-CBG, errors in the CBG can be corrected by a combination of these operations.

3 Construction of H-matrices

In the ECC techniques, fewer 1's in the H-matrix means fewer modulo-2 additions and thus smaller check bit generation delay. Also, a reduction in the number of 1's means fewer gates, leading to reduced hardware overhead and increased hardware reliability [3, 6]. As a consequence, the three H-matrices for the SCO-CBG must be constructed carefully to enable efficient implementation of the SCO-CBG.

In this section, the construction procedure for the H-matrices is explained based on the odd-weight-column code, a well-known single error correction and double error detection (SEC-DED) code. The H-matrix of the odd-weight-column code is selected to be the $\boldsymbol{H}_c$ for the CBG. To implement the SCO-CBG with the most flexibility, the matrix $\boldsymbol{H}_p$ for the parity generator and error locator is constructed from the H-matrices of the shortened Hamming single error correction (SEC) codes. If there is no error on the CBG, parity generator, and parity predictor, the predicted parity bits should be identical to the parity bits. As a result, $\boldsymbol{H}_{pp}$ for the parity predictor is calculated from $\boldsymbol{H}_c$ and $\boldsymbol{H}_p$ as $\boldsymbol{H}_{pp}=\boldsymbol{H}_p\Delta\boldsymbol{H}_c$ with the operator Δ describing the relation of $\boldsymbol{A}\Delta\boldsymbol{B}=(a_{11}\times b_{11})\text{\^{}}(a_{12}\times b_{21})$ where $\boldsymbol{A}$ is 1×2 matrix having a_{11} and a_{12} elements and $\boldsymbol{B}$ is 2×1 matrix having b_{11} and b_{21} elements. $\boldsymbol{H}_p$ is constructed from the $m\times r$ H-matrices of $(r+m,\ r)$ shortened Hamming SEC codes using the following constraints:

1) There should be no zero-weight columns.
2) Each column should be different from the other columns.

DOI: 10.1587/elex.10.20130103
Received February 06, 2013
Accepted March 05, 2013
Published March 25, 2013

3) The number of 1's in each row should be an integer as close to the ratio of the number of 1's in the H-matrix to the number of rows.
4) Columns should be selected to minimize the number of 1's in $\boldsymbol{H}_p$ and $\boldsymbol{H}_{pp}$.

The third constraint enforces every row to have a similar number of 1's in $\boldsymbol{H}_p$, thereby reducing the latency in the parity generator. The fourth constraint minimizes the number of 1's in $\boldsymbol{H}_p$ and $\boldsymbol{H}_{pp}$, leading to a reduction in both the area overhead and the power consumption of the parity generator and parity predictor [3, 6].

Fig. 2 (a) shows a 6 × 16 $\boldsymbol{H}_c$, the H-matrix of the (22, 16) odd-weight-column code for 16 data bits and 6 check bits. Fig. 2 (b) illustrates 4 × 6 $\boldsymbol{H}_p$ constructed from the H-matrices of the (10, 6) shortened Hamming SEC codes under the four constraints. Fig. 2 (c) illustrates a 4 × 16 $\boldsymbol{H}_{pp}$ resulting from the relation $\boldsymbol{H}_{pp}=\boldsymbol{H}_p\Delta\boldsymbol{H}_c$. The SCO-CBG for the SEC-DED code can be easily expanded to accommodate a large number of data bits using $\boldsymbol{H}_c$ as the H-matrix of the conventional odd-weight-column code for large data bits with $\boldsymbol{H}_p$ generated under the four constraints and $\boldsymbol{H}_{pp}$ obtained by the relation $\boldsymbol{H}_{pp}=\boldsymbol{H}_p\Delta\boldsymbol{H}_c$.

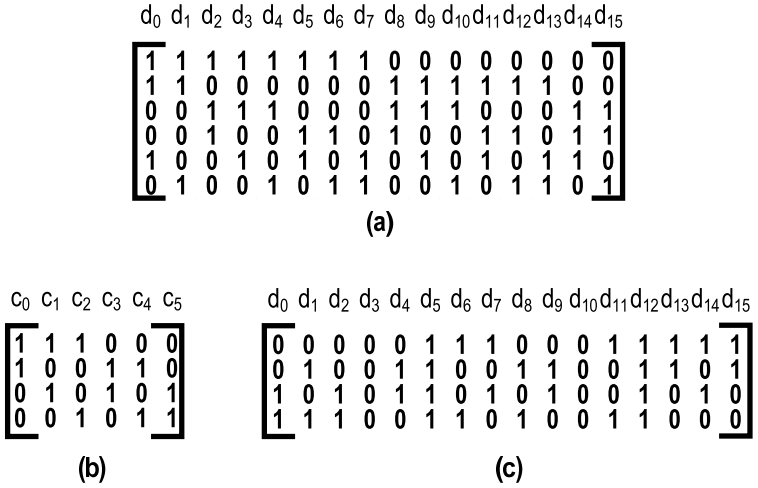

Fig. 2. (a) 6 × 16 H_c for the check bit generator, (b) 4 × 6 H_p for parity generator and error locator, and (c) 4 × 16 H_{pp} for parity predictor

4 Comparison of check bit generators

In order to evaluate the performance of the SCO-CBG, the SCO-CBG and TMR CBG are implemented based on the odd-weight-column codes for the cases with 32, 64, and 128 data bits per word using the Synopsis Design Vision synthesis tool and NanGate 45 nm open-cell library at a 1.25 V supply voltage. Table I tabulates the area, delay, and power consumption of the SCO-CBG and TMR CBG implementations for 32, 64, and 128 data bits per word.

Compared with the TMR CBG, the SCO-CBG brings about a 16.6% increase, a 33.7% decrease, and a 53.2% decrease in the power-delay-area product (PDAP) for 32, 64, and 128 data bits per word, respectively. Although the area and power consumption are sufficiently reduced compared with the TMR CBG, the large delay is a shortcoming for the SCO-CBG. Fortunately, some scheme is able to hide the speed penalty of

DOI: 10.1587/elex.10.20130103
Received February 06, 2013
Accepted March 05, 2013
Published March 25, 2013

Table I. Comparison of performance characteristics for the SCO-CBG and TMR CBG as a function of the number of data bits per word

		32 data bits	64 data bits	128 data bits
Area (μm^2)	TMR	443	977	2,238
	SCO	273	537	1,154
Delay (ns)	TMR	0.29	0.57	1.12
	SCO	0.69	1.14	1.92
Power consumption (μW)	TMR	395	1,084	2,726
	SCO	314	654	1,444

the CBG in the path of write operation in memory using separate and compensated clock signal for check bit arrays [5]. The SCO-CBG with this scheme can ideally substitute the TMR CBG with the maximum benefit in the PDAP. Furthermore, the SCO-CBG is useful for memories employing the scrubbing scheme requiring frequent check bit generation with minimal power consumption.

5 Conclusions

A novel CBG is proposed to self-correct faults in the CBG with less area overhead and power consumption. In the SCO-CBG, check bit errors induced by faults in the CBG are corrected using the ECC techniques. In the SCO-CBG, the parity bits and predicted parity bits function as the check bits being generated in the encoding and decoding operations of the ECC techniques. Also, the construction procedure for the three H-matrices is presented to implement the proposed self-correcting CBG efficiently. When compared to the TMR CBG, the SCO-CBG reduces the area overhead and power consumption. By implementing the proposed SCO-CBG with a scheme that can make up for the speed penalty in the SCO-CBG, a highly reliable CBG can be realized with great cost effectiveness.

Acknowledgments

This work was supported by the National Research Foundation (NRF) grant funded by the Korean government (No. 2010-0026822).

DOI: 10.1587/elex.10.20130103
Received February 06, 2013
Accepted March 05, 2013
Published March 25, 2013