ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2014

A High Performance Decimal Matrix Code Architecture for Improved Reliable Memory

M. Satya sri [1], K.Jyothi [2]

PG Student [VLSI&ES], Dept. of ECE, GIET, Rajahmundry, India [1]

Assistant professor, Dept. of ECE, GIET, Rajahmundry, India [2]

ABSTRACT: Protection codes are necessary to shield memory cells, to maintain good quality level of reliability. But, we don't find any optimized error detection and correction methods. Therefore, in this paper, we present a high performance Decimal Matrix Code to assure the reliability of memory. This protection code utilizes decimal procedure to detect errors, so that more errors were detected and corrected up to 32. Transient multiple cell upsets (MCUs) are major problems in the reliability of memories exposed to radiation environment. To prevent cell upsets from causing data corruption, more complex error correction codes (ECCs) are widely used to protect memory, but they would require higher delay so an efficient ERT (encoder-reuse technique) is proposed to reduce the area overhead of extra circuits, it utilize DMC encoder itself to be part of the decoder.

KEYWORDS: Error correction codes (ECCs), multiple cells upsets (MCUs), memory**.**

I.INTRODUCTION

The general idea for achieving error detection and correction is to add some redundant bits (i.e., some extra data) to a message, which receiver can use to check consistency of the delivered message, and to pick up data determined to be corrupt. Error-detection and correction scheme can be either systematic or non-systematic: In a systematic scheme, the transmitter sends the unique data, and attaches a fixed number of redundant bits, based on particular logic. If only the error detection is required, a receiver can check the same logic to the received data bits and compare its output with the receive check bits; if the values do not match, an error has occurred at some point throughout the transmission. Different types of codes used for Error detection and correction. In a system to uses a non-systematic code, the message is transformed into an encoded message that has at least as many bits as that message. Error detection and correction used to reduce the soft errors. Several techniques are used present to gate upsets in memories. For example, the Bose–Chaudhuri–Hocquenghem codes [8], Reed–Solomon codes [9], punctured difference set (PDS) codes [10], and matrix codes has been used to contact with MCUs in memories. Which requires more area, power, and delay overheads since the encoding and decoding circuits are more complex in these complicated codes.

Reed-Muller code [14] is another protection code that is able to detect and correct additional error than a Hamming code. The main drawback of this protection code is its more area and power requirements. Hamming Codes are more used to correct Single Error Upsets (SEU's) in memory due to their ability to correct single errors through reduced area and performance overhead [13]. Though brilliant for correction of single errors in a data, they cannot correct double bit errors. One more class of SEC-DED codes proposed to detect any number of errors disturbing a single byte. These codes are additional suitable than the conventional SEC-DED codes for protecting the byte-organized memories [15][16]. Though they operate through lesser overhead and are good for multiple error detection, they cannot correct more errors. There are additional codes such as the single-byte-error-correcting, double-byte-error-detecting (SBC-DBD) codes that can correct multiple errors as discussed in [10].

The Single-error-correcting, Double-error-detecting and Double-adjacent-error-correcting (SEC-DED-DAEC) code provides a low cost ECC methodology to correct adjacent errors as proposed in [12]. The only drawback through this code is the possibility of miss-correction for a small subset of many errors. AS CMOS technology scales down to nano-scale and memories are combined through an increasing number of electronic systems, the soft error rate in memory cells is increase, especially when memories operate in space environments due to ionizing effects of atmosphere.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2014

Interleaving technique has been used to restrain MCUs. However, interleaving technique may not be practically used in content-addressable memory (CAM), because of the tight coupling of hardware structures from both cells and comparison circuit structures.

Built-in current sensors (BICS) are proposed to assist with single-error correction and double-error detection codes to provide protection against cell upsets, which can only correct two errors in a word. More recently, in 2-D matrix codes (MCs) are proposed to efficiently correct MCUs per word with a low delay and in this word is divided into rows and columns. The bits per row are protected by Hamming code, while parity is added in each column. For the MC Implemented on Hamming, when two errors were detected by Hamming, the vertical syndrome bits were activated so that these two errors can be corrected. As a result, MC is capable of correcting only two errors. In decimal algorithm with Hamming code has been conceived to be applied at software level. It works to detect and correct soft errors by addition of integer values. The results obtained have shown that this approach have a lower delay overhead over other codes.

The contribution of this paper is a novel decimal matrix code (DMC) based on divide-symbol is implemented to provide enhanced memory reliability. The implemented DMC utilized decimal algorithm (decimal integer addition and decimal integer subtraction) to identify errors. By using decimal algorithm is that the error detection capability was maximized so that the reliability of memory was enhanced. Besides, the encoder-reuse technique (ERT) was implemented to minimize the area overhead of extra circuits (encoder and decoder) without disturbing the whole encoding and decoding processes, because ERT use DMC encoder itself to be part of the decoder.

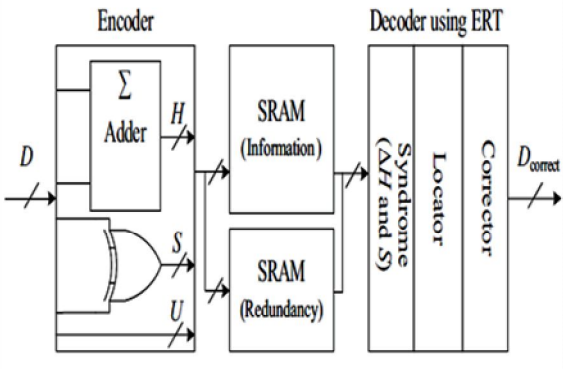


Fig.1 schematic of fault-tolerant memory protected with DMC.

II. HIGH PERFORMANCE IMPLEMENTATION OF DECIMAL MATRIX CODE

In this section, DMC is used to assure reliability in the presence of MCUs through reduced performance overheads, and a 64-bit word is encoded and decoded as an example based on the proposed techniques.

A. Schematic of Fault-Tolerant Memory

The schematic of fault-tolerant memory is depicted in Fig1. First, during the encoding (write) process, information bits D are giving to the DMC encoder, and the horizontal redundant bits H and vertical redundant bits V are obtained from the DMC encoder. After this encoding process DMC codeword is stored in the memory. If MCUs happen in the memory, these errors can be corrected in the decoding (read) method. Due to the advantage of decimal algorithm, DMC has high fault-tolerant capability. In the fault-tolerant memory, the Encoder Reuse technique was used to decrease the area overhead of extra circuits and will be introduced in the following sections.

B. DMC Encoder

In this DMC, first, the divide-symbol and place-matrix ideas were performed, i.e., the N-bit word was divided into k symbols of m bits (N = k × m), and these symbols were arranged in a k1 × k2 2-D matrix (k = k1 × k2, here the values

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2014

of k1 and k2 represents the number of rows and columns in the logical matrix respectively). Second, the horizontal redundant bits H were produced by performing decimal integer addition of selected symbols per row.
Here, each symbol is regard as a decimal integer

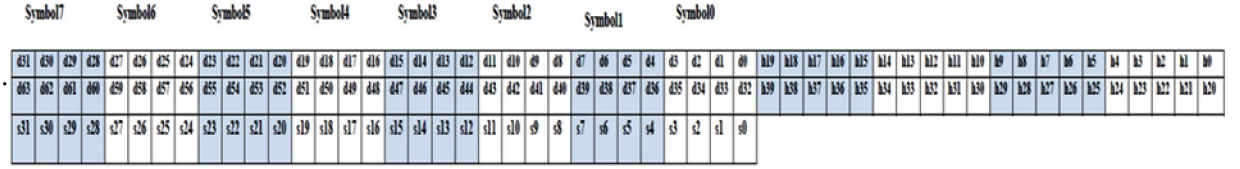


Fig.2 64-bits DMC logical organization (k=2× 8 and m = 4). Here, each symbol is regarded as a decimal integer.

Third, the vertical redundant bits V were obtained by binary operation among the bits per column. It should be noted that both divide-symbol and arrange-matrix are implemented in logical instead of in physical. Therefore, the DMC does not require changing the physical structure of the memory.

To explain the implemented DMC scheme, we have to consider a 64-bit word as an example, as shown in Fig. 2. The cells from D_0 to D_{63} are information bits. This 64-bit word was divided into sixteen symbols of 4-bit. $k_1 = 2$ and $k_2 = 4$ have to select simultaneously. H_0–H_{39} are horizontal check bits; V_0 through V_{31} are vertical check bits. The maximum correction capability (i.e., the maximum size of MCUs can be corrected) and the number of redundant bits are different when the different values for k and m are select. Therefore, k and m must be carefully adjusted to decrease the correction capability and minimize the number of redundant bits. For example, in this case, when k = 2×2 and m = 8, only 1-bit error can be corrected and the number of redundant bits is 80. When k = 4 × 4 and m = 2, 3-bit errors can be corrected and the number of redundant bits is reduced to 32. However, when k = 2 × 4 and m = 4, the maximum correction capability is up to 5 bits and the number of redundant bits is 72. In this paper, in order to enhance the reliability of memory, the error correction capability is first measured,so k = 2 × 8 and m = 4 are utilized to construct DMC.

The horizontal redundant bits H can be obtained by decimal integer addition as follow:

$$H_4H_3H_2H_1H_0 = D_3D_2D_1D_0 + D_{19}D_{18}D_{17}D_{16} \quad (1)$$

$$H_9H_8H_7H_6H_5 = D_7D_6D_5D_4 + D_{23}D_{22}D_{21}D_{20} \quad (2)$$

And similarly for the horizontal redundant bits $H_{14}H_{13}H_{12}H_{11}H_{10}$, $H_{19}H_{18}H_{17}H_{16}H_{15}H_{16}$, $H_{24}H_{23}H_{22}H_{21}H_{20}$, $H_{29}H_{28}H_{27}H_{26}H_{25}$, $H_{34}H_{33}H_{32}H_{31}H_{30}$ and $H_{39}H_{38}H_{37}H_{36}H_{35}$ were calculated. Where "+" represents decimal integer addition.

For the vertical redundant bits V, we have

$$V0 = D_0 \wedge D_{31} \quad (3)$$

$$V1 = D_1 \wedge D_{32} \quad (4)$$

and similarly for the rest vertical redundant bits.

The encoding can be performed by decimal and binary addition operations from (1) to (4). The encoder that computes the redundant bits using multi-bit adders and XOR gates is shown in Figure. In this figure, $H_{39} - H_0$ are horizontal redundant bits,

$V_{31} - V_0$ are vertical redundant bits, and the remaining bits $U_{63} - U_0$ are the information bits which are directly copied from D_{31} to D_0.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2014

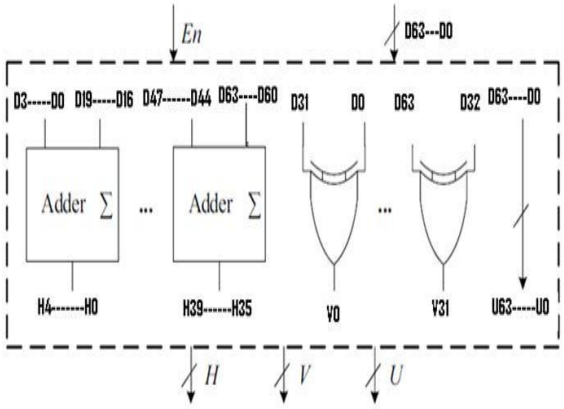


Fig.3 64-bit DMC encoder structure using multi bit adders and XOR gates

C. DMC Decoder

To obtain a word being corrected, the decoding process is required. For example, first, the received redundant bits $H_4H_3H_2H_1H_0$' and V_0'-V_3' are generated by the received information bits D'. Second, the horizontal syndrome bits $\Delta H_4H_3H_2H_1H_0$ and the vertical syndrome bits S_3−S_0 can be calculated as follows:

$$\Delta H_4H_3H_2H_1H_0=H_4H_3H_2H_1H_0'-H_4H_3H_2H_1H_0 \quad (5)$$

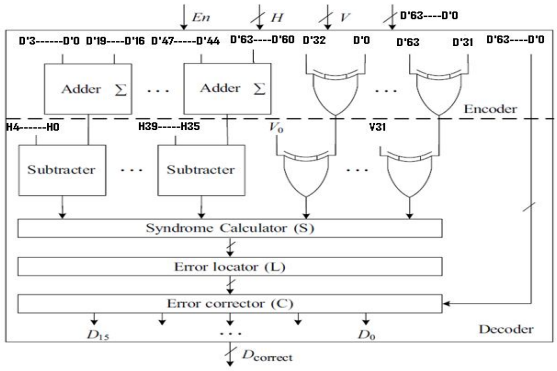
$$S_0=V_0'\wedge V_0 \quad (6)$$

and alike for the rest vertical syndrome bits, where "–" represents decimal integer subtraction.

When $\Delta H_4H_3H_2H_1H_0$ and $S_3 - S_0$ are equal to zero, the stored codeword have original information bits in symbol 0 where no errors happen. When $\Delta H_4H_3H_2H_1H_0$ and $S_3 - S_0$ are nonzero, the induced errors (the quantity of errors is 4 in this case) are detected and located in symbol 0, and then the errors can be corrected by

$$D_{0correct} = D_0 \wedge S_0 \quad (7)$$

The DMC decoder is depicted in Fig 4, which is prepared up of the following sub modules, and each executes a particular task in the decoding process: syndrome calculator, error locator, and error corrector. It can be observed from the fig.4 that the redundant bits must be recomputed from the received information bits D' and compare to the original set of redundant bits in order to obtain the syndrome bits ΔH and S. Then error locator uses ΔH and S to detect and locate which bits some errors occur in. Finally, in the error corrector, these errors can be corrected by inverting the values of error bits.



Extra circuit	*En* signal		Function
	Read signal	Write signal	
Encoder	0	1	Encoding
	1	0	Compute syndrome bits

Fig.4 64-bit DMC decoder structure using ERT

10.15662/ijareeie.2014.0310043

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2014

In the proposed scheme, the circuit area of DMC is minimized with reusing its encoder. This is calling the ERT. The ERT can decrease the area overhead of DMC without disturbing the entire encoding and decoding processes. From Fig 4, it can be practical that the DMC encoder is also reused for obtaining the syndrome bits in DMC decoder. Therefore, the entire circuit area of DMC can be minimized as a result of using the existent circuits of encoder. Besides, the figure shows the proposed decoder with an allow signal En for deciding whether the encoder needs to be a part of the decoder. In other words, the En signal is used for distinguishing the encoder from the decoder, and it is under manage of the write and read signals in memory. Therefore, in the encoding (write) mode, the DMC encoder is only an encoder to execute the encoding operations. However, in the decoding (read) mode, this encoder is employed for computing the syndrome bits in the decoder. These obviously show how the area overhead of extra circuits can be substantially decreased.

III.SIMULATION RESULTS

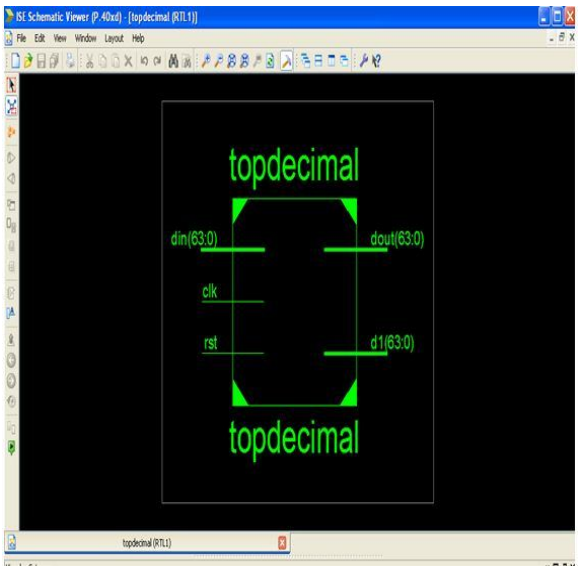


Fig.5 Block diagram

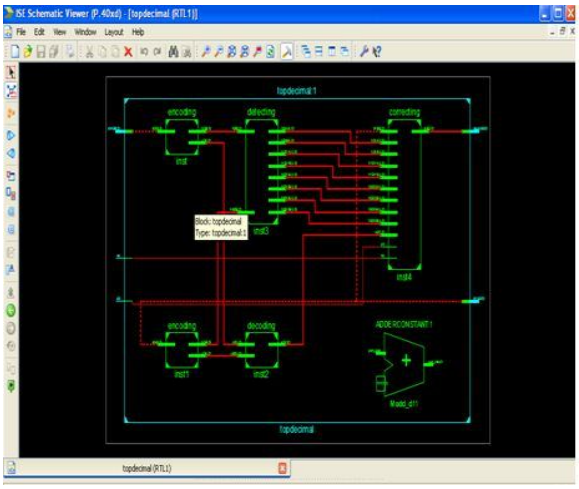


Fig.6 RTL Schematic diagram

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2014

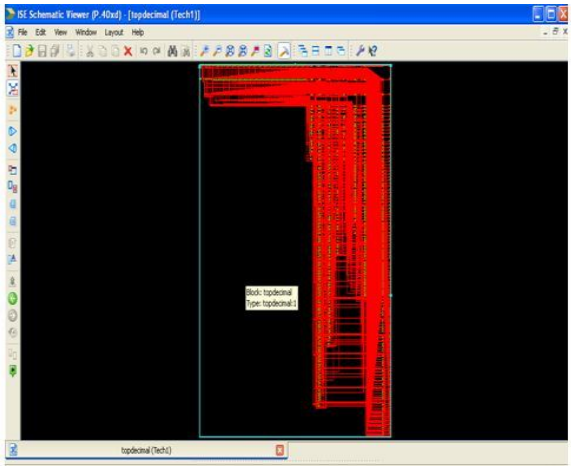


Fig.7 Technology schematic

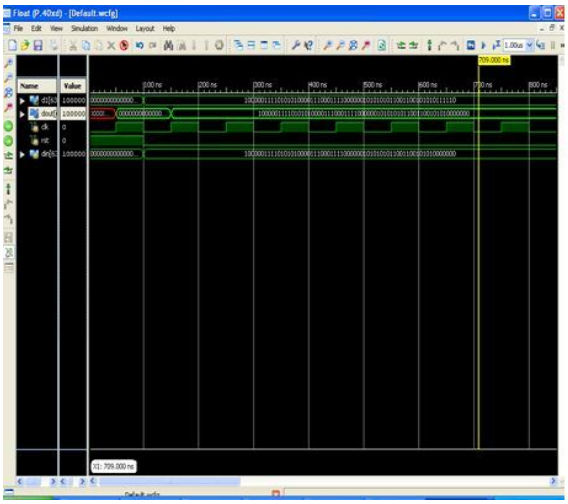


Fig.8 Simulation output waveform

IV.PERFORMANCE ANALYSIS

Performance simulation of DMC is carried out successfully with model Sim Altera 6.5e simulator.
Simulation and synthesis has been carried out with XILINX ISE.The frequency is 132.258MHZ, the power supply voltage is 1.1Vand delay of 7.038ns, no.of slice flip-flops are 184, 4 input LUTs 326 and no. of IOBS 194 and no. of GCLKS 1.

V. CONCLUSION

In this method we have implemented the 64-bit decimal matrix code for detection and correction of errors in memories. To avoid MCUs from causing data corruption, more complex error correction codes (ECCs) are widely used to protect memory, but the main problem is that they would require higher delay overhead. Newly, matrix codes (MCs) based on hamming codes have been proposed for memory protection. In this implemented system error detection and correction rate was increased compared to 32-bit decimal matrix code.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2014

REFERENCES

[1] Jing Guo, Liyi Xiao, Member, IEEE, Zhigang Mao, Member, IEEE, and QiangZhao,"Enhanced memory reliability against multiple cell upsets using Decimal Matrix Code" IEEE Trans. Very Large ScaleIntegr. (VLSI) Syst., vol. 22, no. 1, pp.127-135, Mar 2013.
[2] 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
[3] E. Ibe, H. Taniguchi, Y. Yahagi, K. Shimbo, and T. Toba, "Impact of scaling on neutron induced soft error in SRAMs from an 250 nm to a 22 nm design rule," IEEE Trans. Electron Devices, vol. 57, no. 7, pp. 1527–1538, Jul. 2010.
[4] C. Argyrides and D. K. Pradhan, "Improved decoding algorithm for high reliable reed muller coding," in Proc. IEEE Int. Syst. On Chip Conf., Sep. 2007, pp. 95–98.
[5] A. Sanchez-Macian, P. Reviriego, and J. A. Maestro, "Hamming SEC-DAED and extended hamming SEC-DED-TAED codes through selective shortening and bit placement," IEEE Trans. Device Mater. Rel., to be published.
[6] S. Liu, P. Reviriego, and J. A. Maestro, "Efficient majority logic fault detection with difference-set codes for memory applications," IEEETrans. Very Large Scale Integr. (VLSI) Syst., vol. 20, no. 1, pp. 148–156, Jan. 2012.
[7] M. Zhu, L. Y. Xiao, L. L. Song, Y. J. Zhang, and H. W. Luo, "New mix codes for multiple bit upsets mitigation in fault-secure memories," Microelectron. J., vol. 42, no. 3, pp. 553–561, Mar. 2011.
[8] R. Naseer and J. Draper, "Parallel double error correcting code design to mitigate multi-bit upsets in SRAMs," in Proc. 34th Eur. Solid-StateCircuits, Sep. 2008, pp. 222–225.
[9] G. Neuberger, D. L. Kastensmidt, and R. Reis, "An automatic technique for optimizing Reed-Solomon codes to improve fault tolerance in memories," IEEE Design Test Comput., vol. 22, no. 1, pp. 50–58, Jan.–Feb. 2005.
[10] P. Reviriego, M. Flanagan, and J. A. Maestro, "A (64,45) triple error correction code for memory applications," IEEE Trans. Device Mater. Rel., vol. 12, no. 1, pp. 101–106, Mar. 2012.
[11] S. Baeg, S. Wen, and R. Wong, "Interleaving distance selection with a soft error failure model," IEEE Trans. Nucl. Sci., vol. 56, no. 4, pp. 2111–2118, Aug. 2009.
[12] K. Pagiamtzis and A. Sheikholeslami, "Content addressable memory (CAM) circuits and architectures: A tutorial and survey," IEEE J.Solid-State Circuits, vol. 41, no. 3, pp. 712–727, Mar. 2003.
[13] 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.
[14] E. Ibe, H. Taniguchi, Y. Yahagi, K. Shimbo, and T. Toba, "Impact of scaling on neutron induced soft error in SRAMs from an 250 nm to a 22 nm design rule," IEEE Trans. Electron Devices, vol. 57, no. 7, pp. 1527–1538, Jul. 2010.
[15] C. Argyrides and D. K. Pradhan, "Improved decoding algorithm for high reliable reed muller coding," in Proc. IEEE Int. Syst. On Chip Conf., Sep. 2007, pp. 95–98.
[16] A. Sanchez-Macian, P. Reviriego, and J. A. Maestro, "Hamming SEC-DAED and extended hamming SEC-DED-TAED codes through selective shortening and bit placement," IEEE Trans. Device Mater. Rel., to be published.
[17] S. 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.
[18] M. Zhu, L. Y. Xiao, L. L. Song, Y. J. Zhang, and H. W. Luo, "New mix codes for multiple bit upsets mitigation in fault-secure memories," Microelectron. J., vol. 42, no. 3, pp. 553–561, Mar. 2011.
[19] R. Naseer and J. Draper, "Parallel double error correcting code design to mitigate multi-bit upsets in SRAMs," in Proc. 34th Eur. Solid-State Circuits, Sep. 2008, pp. 222–225.
[20] G. Neuberger, D. L. Kastensmidt, and R. Reis, "An automatic technique for optimizing Reed-Solomon codes to improve fault tolerance in memories," IEEE Design Test Comput., vol. 22, no. 1, pp. 50–58, Jan.–Feb. 2005.
[21] P. Reviriego, M. Flanagan, and J. A. Maestro, "A (64,45) triple error correction code for memory applications," IEEE Trans. Device Mater. Rel., vol. 12, no. 1, pp. 101–106, Mar. 2012.
[22] S. Baeg, S. Wen, and R. Wong, "Interleaving distance selection with a soft error failure model," IEEE Trans. Nucl. Sci., vol. 56, no. 4, pp. 2111–2118, Aug. 2009.