IJITCE

ISSN 2347- 3657

International Journal of
Information Technology & Computer Engineering

www.ijitce.com

Email : ijitce.editor@gmail.com or editor@ijitce.com

International Journal of Information Technology & Computer Engineering

ISSN 2347–3657
Volume 1,Issue 1,Feb 2013

AN EFFICIENT MULTIBIT UPSET DETECTION AND CORRECTION IN 64-BIT SRAM BASED FPGA MEMORY USING DECIMAL MATRIX CODE

Dr. MVS Prasad

Abstract:

In order to safeguard memory against radiation, this study lays forth an effective method for correcting Multiple Bit Upsets (MBUs). The main problem with the old method of protecting memories from MBUs—which included using several sophisticated error correction codes—was the increased redundant memory expense.To guarantee memory dependability, this article suggested a 64-bit Matrix Code. In order to find and fix additional mistakes, the suggested protection code used a process. The results demonstrated a degree of protection against huge MBUs in memory for the suggested method. One of the most significant issues with the dependability of memory in a radiation environment is transient multiple bit upsets (MBUs). To repair memory errors, the suggested approach makes use of a 64-bit matrix. More complicated error correction codes (ECCs) are often used to safeguard memory against MBUs corrupting data, but their primary drawback is the increased delay overhead they entail. There have been recent proposals for memory protection using matrix codes (MCs) based on Hamming codes. The fact that they are error correction codes twice is the biggest problem, and not every situation benefits from the expanded mistake correction capabilities. Also, without affecting the overall encoding and decoding procedures, erasure codes are suggested as a means to lessen the area overhead of additional circuits. These days, several error detection and repair techniques are used to secure memory bits with protection codes, which is essential for ensuring a high degree of dependability. More redundant bits are needed to ensure improved memory dependability, which is the only downside of the present MC. In order to guarantee stability in the face of multiple bit upset, remove unnecessary bits, and rectify more errors than the present system, the suggested approach employed matrix coding.

Keywords: FPGA, Multiple Bit Upsets, Reliability,Soft Errors

I. INTRODUCTION

The soft error rate in memory cells is growing at a rapid pace as CMOS technology shrinks to the nanoscale and memories are integrated with more and more electronic systems. This is particularly true when memories function in space environments, subjected to the ionizing effects of atmospheric neutrons, alpha particles, and cosmic rays. Memory reliability is greatly affected by single bit upsets, but in some memory applications, reliability is also greatly affected by multiple cell upsets (MCUs). For a long time, certain error correction codes (ECCs) have been extensively utilized to safeguard memories against soft faults, with the goal of making memory cells as fault-tolerant as feasible. For instance, memory-related MCUs have been addressed using codes such as Bose, Chaudhuri, Hocquenghem, Reed-Solomon, and punctured difference set (PDS). On the other hand, these codes incur additional space, power, and delay costs.

Department of ECE, ANU college of Engineering and Technology, Guntur, India

International Journal of
Information Technology & Computer Engineering
ISSN 2347–3657
Volume 1,Issue 1,Feb 2013

because these intricate codes need more sophisticated encoding and decoding systems.In order to accomplish error detection and repair, it is common practice to include redundant data in messages. This allows the receiver to verify the integrity of the message and recover corrupted data. It is possible to have a systematic or non-systematic approach to error identification and correction: In a methodical scheme, the sender transmits the one-of-a-kind data along with a predetermined amount of check bits, also called parity data, that are calculated from the data bits using a deterministic algorithm. If the receiver's main concern is with error detection, then all it needs to do is apply the same method to the data bits it has received and compare the results to the receive check bits; if the values don't match, then an error has happened during transmission. Lower-layer communication and dependable storage on media like CDs, DVDs, hard drives, and RAM both make frequent use of error-correcting codes.Due to its low cost, high density, flexibility, and rapid time-to-market, static RAM based FPGAs are most often employed in a range of applications. The configuration data of logic cells is stored in the static memory of an FPGA that is based on SRAM, which is structured as a latch array. Creating intricate digital circuits is a breeze using FPGA. Using SRAM also lowers power usage. At high frequencies, static random access memory (SRAM) may be just as power-hungry as dynamic random access memory (RAM), and certain integrated circuits can require several watts when operating at maximum bandwidth. In contrast, static RAM requires a tiny amount of power when not in use (a few micro watts) and uses even less power while running at a slightly slower rate, as is the case with applications that employ moderately clocked microprocessors. To control how much power SRAM-based memory architectures use, many methods have been suggested.Arrays of programmable logic blocks linked by a programmable routing network and I/O blocks make up an FPGA device that can be customized using SRAM. FPGA devices based on SRAM are gaining popularity because to their re-programmability, low development cost, and good performance.FPGAs using denser integration methods and nanoscale technology. One of the most common components of electronic systems is the memory. The efficiency of a circuit is significantly diminished when exposed to environmental radiation. A charged particle in the environment may cause a single-event upset (SEU) if it strikes a circuit's silicon and causes an error. Soft mistakes occur when an FPGA device makes a mistake that compromises the mapped design's functioning. The hardware of a system is unaffected by a soft mistake; the data being processed in memory is the only thing that gets damaged. One solution to this problem is the use of Built-in Current Sensors (BICS) in combination with Single Error Correction/Double Error Detection (SEC-DED) codes to safeguard memory against MBUs. However, only SEU could be fixed using those procedures. This study presents a general scrubbing system that uses the idea of the Erasure coding algorithm to recreate the erroneous configuration frame. It may be used for both mistake detection and rectification. This Erasure coding scheme sorts interleaving distance into horizontal and vertical parity, which is used to identify MBUs.

II. Patterns of MBUs

To control MBUs, which physically rearrange bits to create new words from the same logical word, the interleaving approach has been used. The interleaving method, however,may not be applicable to content-addressable memory (CAM) due to the close relationship between the hardware architectures of bits and comparison circuits.A more recent proposal suggests using 2-dimensional matrix codes (MCs), which split a word into many logical columns and rows, to effectively correct MBUs per word with a short decoding time. Each column has parity code applied to it, and the bits per row are secured by Hamming code. In a Hamming-based MC, the vertical syndrome bits are turned on to fix the two mistakes that are discovered by Hamming. Consequently, MC can only ever fix a maximum of two mistakes. It has been proposed to implement a solution at the software level that integrates algorithmic and Hamming code. To find and fix subtle mistakes, it adds integer values. The results demonstrate that this method outperforms other codes in terms of delay overhead. To help with single-error correction and double-error detection codes, built-in current sensors (BICS) are suggested to provide protection against MBUs. But this method can only fix two mistakes per word.To improve memory reliability, this work proposes a new matrix code based on the divide-symbol algorithm. An technique that uses integer addition and integer subtraction is used to find flaws in the proposed matrix coding. Using an algorithm has several benefits, one of which is improving memory dependability by maximizing error detection capabilities. Also, without interfering with the encoding and decoding procedures, erasure codes are suggested as a way

ISSN 2347–3657
Volume 1,Issue 1,Feb 2013

International Journal of
Information Technology & Computer Engineering

to reduce the area overhead of additional circuits.

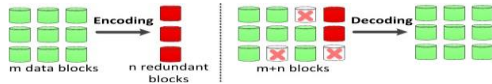


Fig. 1 Encoding and decoding of erasure codes.

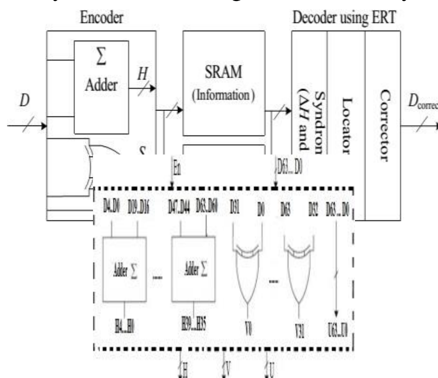
II. PROPOSED DMC

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

A. Proposed Schematic of Fault-Tolerant Memory:

The schematic of fault-tolerant memory is as shown in Fig. 2. The DMC encoder is fed with information bits D, during the encoding (write) process, and then the DMC encoder produces the vertical redundant bits V and horizontal redundant bits H. The obtained DMC code word is stored in the memory, once encoding process is completed. If the memory is affected by MCUs, in the decoding (read) process these errors can be corrected. The proposed DMC has higher fault- tolerant capability with higher performance because of decimal algorithm. The fault-tolerant memory uses ERT technique, to reduce extra circuit's area overhead and will be introduced in the following sections.Proposed DMC Encoder

The DMC proposes, first, it performs the ideas of divide-symbol and arrange-matrix, i.e. symbols of m



bits (N = k × m) is obtained by dividing the N-bit word,and these symbols are arranged in a k1 × k2 2-D matrix (k = k1 × k2,

Fig. 2 Proposed schematic of fault-tolerant memoryprotected with DMC

where k1 and k2 values represent thenumbers of rows and columns in the logical matrix respectively). Second by decimal integer addition of selected symbols per row the horizontal redundant bits H are obtained. Here, each symbol is regarded as a decimal integer. Third, by binary operation among the bits per column the vertical redundant bits V are obtained. It should be noted that instead of in physical both divide-symbol and arrange-matrix are implemented in logical. Therefore, the physical structure of the memory is not required to be changed as according to the proposed DMC.

We considered a 64-bit word as an example, to explain the proposed DMC scheme, as shown in Fig. 2. From D0 to D63 cells are information bits. Eight symbols of 4-bit are obtained by dividing 64-bit word. By choosingk1 = 2 and k2 = 4 simultaneously. Horizontal check bits are H0–H39; vertical check bits are V0 through V31 are. However, it should be mentioned that the number of redundant bits and the maximum correction capability (i.e., the maximum size of MCUs can becorrected) are different when values for k and m are chosen different. Therefore, to maximize the correction capability and minimize the number of redundant bits k and m should be carefully adjusted to maximize the correction capability and minimize the number of redundant bits. For example, in this case, when k = 2×2and 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 upto 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

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 1,Issue 1,Feb 2013

considered, 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 follows

H4H3H2H1H0 = D3D2D1D0 + D19D18D17D16 (1) H9H8H7H6H5 = D7D6D5D4 + D23D22D21D20 (2)

and similarly for the horizontal redundant bits H14H13H12H11H10, H19H18H17H16H15 H16,H24H23H22H21H20, H29H28H27H26H25, H34H33H32H31H30 and H39H38H37H36H35 where "+" represents decimal integer addition. For the vertical redundant bits V, we have

V0 = D0 ^ D31 (3)

V1 = D1 ^ D32 (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 Fig. In this figure, H39 − H0 arehorizontal redundant bits, V31 − V0 are vertical redundant bits, and the remaining bits U63 − U0 are theinformation bits which are directly copied from D31 to D0. The enable signal En will be explained in the next section.

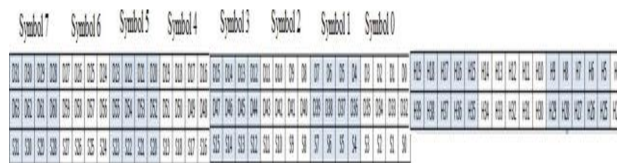


Fig 3: 64-bits DMC logical organization (k = 2 × 8 and m = 4). Here, each symbol is regarded as a decimal integer

Fig 4: 64-bit DMC encoder structure using multi bit adders and XOR gates

B. Proposed DMC Decoder

To obtain a word being corrected, the decoding process is required. For example, first, the received redundant bits H4H3H2H1H0' and V0'-V3' are generated by the received information bits D'.Second, the horizontal syndrome bits ΔH4H3H2H1H0 and the vertical syndrome bits S3 − S0 can be calculated as follows:

ΔH4H3H2H1H0 = H4H3H2H1H0' − H4H3H2H1H0 (5)information bits D' and compared 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, inthe error corrector, these errors can be corrected by inverting the values of error bits.

$$S0 = V0'^\wedge V0 \quad (6)$$

and similarly for the rest vertical syndrome bits, where "−" represents decimal integer subtraction. When ΔH4H3H2H1H0 and S3 − S0 are equal tozero, the stored code word has original information

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

bits in symbol 0 where no errors occur. When ΔH4H3H2H1H0 and S3 − S0 are nonzero,theinduced errors (the number of errors is 4 in this case) are detected and located in symbol 0, and then these errors can be corrected by

$$D0correct = D0 \wedge S0 \qquad (7)$$

The proposed DMC decoder is depicted in Fig, which is made up of the following sub modules, and each executes a specific task in the decoding process: syndrome calculator, error locator, and errorcorrector. It can be observed from this figure that the redundant

International Journal of
Information Technology & Computer Engineering
ISSN 2347–3657
Volume 1,Issue 1,Feb 2013

bits must be recomputed from the received

Fig 5: 64-bit DMC decoder structure using ERT

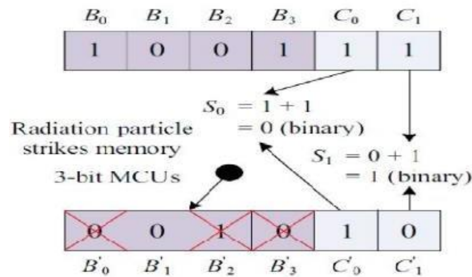
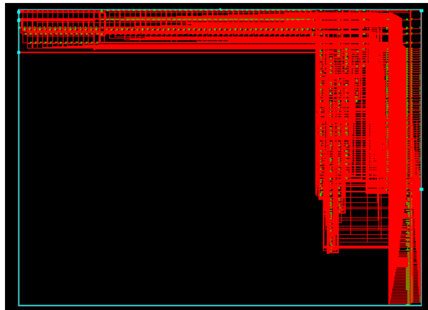


Fig 6: limits of binary error detection in simple binary operations

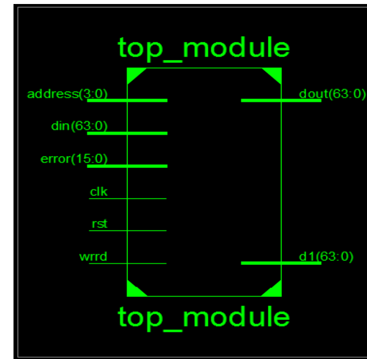


By recycling its encoder, the suggested method reduces the DMC's circuit size. This system is

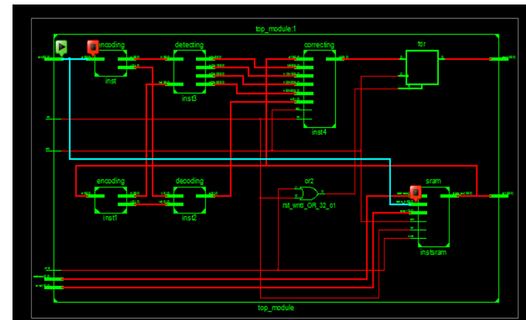
known as the ERT. Without interfering with the whole encoding and decoding process, the ERT may lower the DMC area overhead. It is also possible to acquire the syndrome bits in the DMC decoder by reusing the DMC encoder, as shown in Figure. Consequently, by using the existing encoder circuits, the total circuit size of the DMC may be decreased. In addition, the suggested decoder with an enable signal En for determining whether the encoder is required to be part of the decoder is also shown in this picture. To rephrase, the En signal, which is controlled by the write and read signals in memory, is used to differentiate the encoder from the decoder. As a result, the DMC encoder is only an encoder during the write phase of encoding. Decoding (reading) requires this encoder, however, since it calculates the syndrome bits. These examples demonstrate how the area overhead of additional circuits may be significantly reduced.

III. RESULT

SBlock diagram



RTL Schematic



Technology Schematic

Simulation Results

IV. CONCLUSION

This work introduced DMC as a means to guarantee memory reliability. More mistakes were found and fixed since the suggested protection code used a decimal approach to identify them. The findings collected suggest that the proposed technique provides a greater degree of protection against big MCUs in memory. To maximize memory reliability while minimizing the number of redundant bits, a reasonable combination of k and m should be chosen based on radiation experiments in actual implementation. The only drawback of the proposed DMC is that more

redundant bits are required to maintain higher reliability of memory. So, in order to keep the suggested approach reliable and reduce the number of unnecessary bits, further work will be done in the future

.**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]

[7] 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.

[8] 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," Micro electron. J., vol. 42, no. 3, pp. 553–561, Mar. 2011.

[9] R. Naseer and J. Draper, "Parallel double error correcting code design to mitigate multi-bit upsets inSRAMs," in Proc. 34th Eur. Solid-State Circuits, Sep. 2008, pp. 222–225.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

A. Sanchez-Macian, P. Reviriego, and J. A.

Maestro, “Hamming SEC-DAED and extended hamming SEC-DED-TAED codes through selective