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
Volume2,Issue 4,Oct 2014

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

Jangali Murnalini Yadav

Abstract

In order to safeguard memory against radiation, this study lays forth an effective method for correcting Multiple Bit Upsets (MBUs). There has been a significant increase in duplicate memory overhead due to the earlier employment of several complicated error correction codes (ECCs) to safeguard memories from MBUs.The study suggests using a 64-bit Matrix Code to guarantee memory stability. In order to find and fix additional mistakes, the suggested protection code used a process. The results shown that the suggested method offers some defense against memory-bound big MBUs. One of the most significant issues with the dependability of memory in a radiation environment is transient multiple bit upsets (MBUs). We use a 64-bit matrix for memory error correction in the suggested technique. 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. Memory protection matrix codes (MCs) derived from Hamming codes have recently been suggested. The fact that they are error correction codes twice is the biggest problem, and not every situation benefits from the expanded mistake correction capabilities. In addition, erasure codes are suggested as a means to lessen the space overhead of additional circuits without interfering with the overall encoding and decoding procedures. These days, protecting memory bits with codes is essential for maintaining a decent degree of dependability; to achieve this goal, several mistakes approaches for detection and correction are being used. The current MC has one major flaw: in order to keep memory reliability high, additional redundant bits are needed. Matrix coding increases dependability in the event of many bit upsets, reduces unnecessary bits, and corrects more errors than the current approach.

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

I. INTRODUCTION

The soft error rate in memory cells is swiftly rising 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. Some codes that have been used to handle MCUs in memory include the Bose, Chaudhuri, Hocquenghem, Reed-Solomon, and punctured difference set (PDS) codes. However, these codes incur additional costs in terms of space, energy, and latency. because these intricate codes need more sophisticated encoding and decoding systems.

DEPT OF ELECTRICAL & ELECTRONINCS ENGINEERING, GEETHANJALI COLLEGE OF ENGINEERING & TECHNOLOGY, KURNOOL, A.P, INDIA)

ISSN 2347–3657
Volume2,Issue 4,Oct 2014

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. Both systematic and non-systematic approaches to error identification and repair are possible: A systematic technique involves a deterministic algorithm deriving a defined number of check bits (or parity data) from the data bits and attaching them to the unique data that the transmitter provides. If the receiver only needs to identify errors, all it has to do is apply the same technique to the bits of data it has received and compare the results to the bits of check bits it has received; if the values don't match, then an error 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.

Fast time-to-market, adaptability, density, and cost-effectiveness are the primary reasons why static RAM based FPGAs are so popular in many different applications. The static memory of an FPGA that is based on SRAM is structured like an array of latches, and it holds the configuration data of the logic cells. Creating intricate digital circuits is a breeze using FPGA. Use of SRAM also results in a decrease in power consumption. When used at high frequencies, static random access memory (SRAM) may be just as power-hungry as dynamic random access memory (RAM), and some integrated circuits can consume several watts at maximum bandwidth. Conversely, static RAM used at a slightly reduced rate, as in programs using moderately clocked microprocessors, consumes very little power and may even have an almost insignificant power footprint. powered down to a few microwatts while not in use. Power management of SRAM-based memory architectures has been approached from several angles.

The functional programable gate array (FPGA) is an SRAM-customizable device that has an array of logic blocks and an interconnection network for routing and input/output (I/O). The re-programmability, low development cost, and excellent performance of SRAM-based FPGA devices are driving their rising popularity.Field-programmable gate arrays that use denser integration methods and are based on nanoscale technology. When it comes to electronic systems, memories are among the most utilized components. 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. These kinds of mistakes, known as soft faults, in FPGA devices impact the mapped design's functioning. A soft mistake can only harm the data being processed in memory and not the hardware of the system. Memory Buffer Unlock (MBU) protection has lately been enhanced with the use of Built-in Current Sensors (BICS) and Single Error Correction/Double Error Detection (SEC-DED) codes. However, in such case, only SEU could be fixed. This study introduces a general scrubbing system that uses the notion 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.

PART II: MBU DESIGNS

To control MBUs, which physically rearrange bits to create new words from the same logical word, the interleaving approach has been used. Nevertheless, the interleaving method mightwon't work in content-addressable memory (CAM) due to the close relationship between the hardware architectures of bits and comparison circuits.2-D matrix codes (MCs) have now been suggested as a way to effectively correct MBUs per word while keeping the decoding latency low. These codes partition a word into numerous rows and columns by logic. Hamming code is used to safeguard the bits per row, and parity code is applied to each column. 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. A software-level strategy has been developed 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. As a means of bolstering defense against MBUs, built-in current sensors (BICS) are suggested to work in tandem with single-error correction and double-error detection codes. Nevertheless, this method is limited to fixing two mistakes each word.An innovative matrix code that improves memory reliability is suggested in this study, which is based

International Journal of
Information Technology & Computer Engineering
ISSN 2347–3657
Volume2,Issue 4,Oct 2014

on the divide-symbol algorithm. A combination of integer addition and integer subtraction is used by the suggested matrix code as a technique to detect faults. 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 to reduce the area overhead of additional circuits.

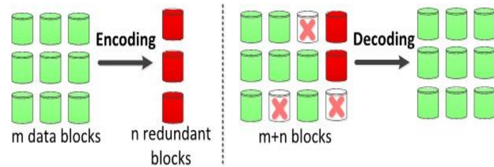
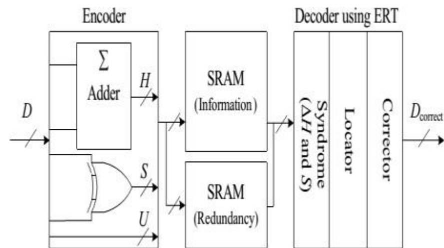


Fig. 1 Encoding and decoding of erasure codes.

II. PROPOSED DMC



In this section, DMC is proposed to assure reliability in the presence of MCUs with reduced performance

B. 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, 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 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.

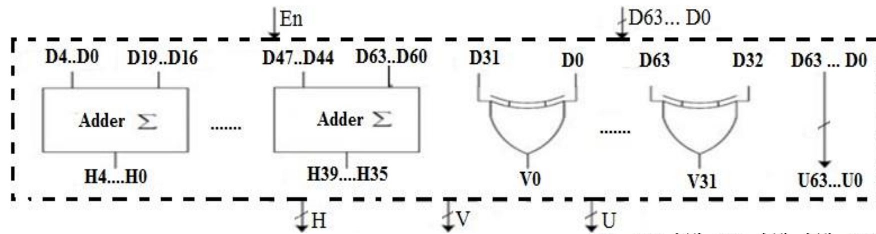
Fig. 2 Proposed schematic of fault-tolerant memoryprotected with DMC

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 bitsare 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

International Journal of
Information Technology & Computer Engineering
ISSN 2347–3657
Volume2,Issue 4,Oct 2014

= 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 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

C. 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

International Journal of Information Technology & Computer Engineering

ISSN 2347–3657
Volume2,Issue 4,Oct 2014

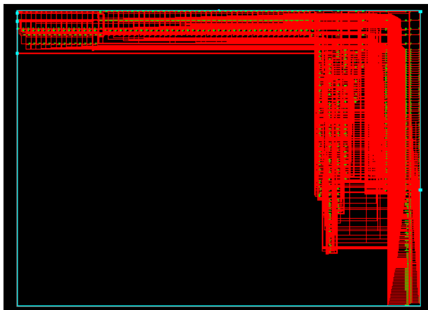
ΔH and S. Then error locator uses ΔH and S to detect and locate which bits some errors occur in. Finally, in the error corrector, these errors can be

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

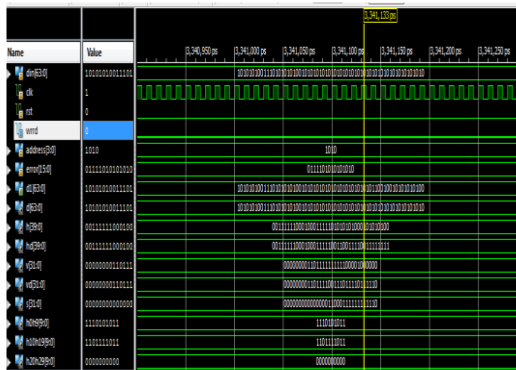
s.

$$S0 = V0'\wedge V0 \qquad (6)$$

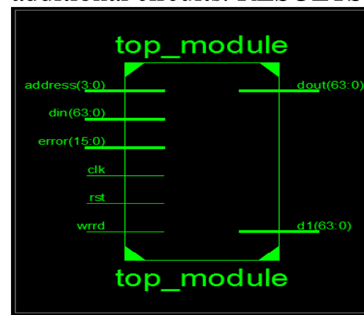
and similarly for the rest vertical syndrome bits, where "–" represents decimal integer subtraction. When ΔH4H3H2H1H0 and S3 − S0 are equal to zero, the stored code word has original information bits in symbol 0 where no errors occur. When ΔH4H3H2H1H0 and S3 − S0 are nonzero, theinduced errors (the number of errors is 4 in this case) are detected and located in symbol 0, and then these errors can be corrected by



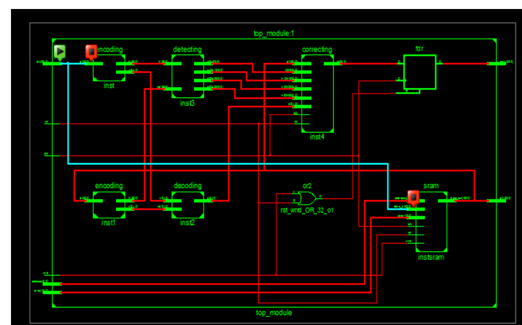
$$D0correct = D0 \wedge S0 \qquad (7)$$



The proposed DMC decoder is depicted in Fig, whichis 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 bits must be recomputed from the receivedDMC decoder additionally makes use of the encoder to get the syndrome bits. 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. They make it quite evident how to drastically cut down on the area overhead of additional circuits. RESULTSBlock diagram



RTL Schematic



Technology Schematic

Simulation Results

ISSN 2347–3657
Volume2,Issue 4,Oct 2014

Fig 5: 64-bit DMC decoder structure using ERT

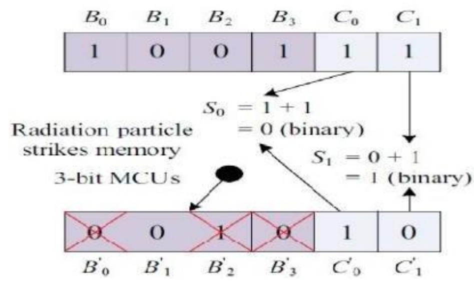


Fig 6: limits of binary error detection in simple binary operations

In the proposed scheme, the circuit area of DMC is minimized by reusing its encoder. This is called the ERT. The ERT can reduce the area overhead of DMC without disturbing the whole encoding and decoding processes. From Fig, it can be observed that the DMC

V. FINAL THOUGHTS

This work introduced DMC as a means to guarantee memory reliability. The suggested protection code made use of a decimal method to identify flaws, leading to a higher rate of error detection and correction. According to the findings, the suggested approach offers better protection against memory-resident big MCUs. 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] S. Liu, P. Reviriego, and J. A. Maestro, "Efficient majority logic fault detection with difference-set codes for memory applications," IEEETrans. VeryLarge 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," Micro electron.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

ISSN 2347–3657
Volume2,Issue 4,Oct 2014

SRAMs," in Proc. 34th Eur. Solid-State Circuits, Sep. 2008, pp. 222–225.

[9] G. Neuberger, D. L. Kastensmidt, and R. Reis, "An automatic technique for optimizing Reed-

[11] 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

Solomon codes to improve fault tolerance in memories," IEEE Design Test Comput., vol. 22, no.1, pp. 50–58, Jan.–Feb. 2005.

[10] P.

nmtechnology 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.

[17] A. Sanchez-Macian, P. Reviriego, and J. A. Maestro, "Hamming SEC-DAED and extended hamming SEC-DED-TAED codes through selective