ENGINEERING IN ADVANCED RESEARCH SCIENCE AND TECHNOLOGY

ISSN 2278-2566
Vol.01, Issue.03
July -2019
Pages: -189-196

LATENCY OPTIMIZED FAULT TOLERANT ARCHITECTURE USING DMC AND MODIFIED CSA

1. **YARLAGADDA SUNITHA, 2. KAVULURU ANIL JAYAKIRAN**
1. M.Tech Student, Dept. of ECE, Prasiddha College of Engineering & Technology, Anathavaram,AP
2. Assistant Professor, Dept. of ECE, Prasiddha College of Engineering & Technology, Anathavaram,AP

ABSTRACT: Transient multiple cell upsets (MCUs) are becoming major issues in the reliability of memories exposed to radiation environment. To prevent 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. Recently, matrix codes (MCs) based on hamming codes have been proposed for memory protection. The main issue is that they are double error correction codes and the error correction capabilities are not improved in all cases. In this project, novel decimal matrix code (DMC) based on divide-symbol is proposed to enhance memory reliability with lower delay overhead. The proposed DMC utilizes decimal algorithm to obtain the maximum error detection capability. Moreover, the encoder-reuse technique (ERT) is proposed to minimize the area overhead of extra circuits without disturbing the whole encoding and decoding processes. ERT uses DMC encoder itself to be part of the decoder. Further, this project is enhanced using square root Carry select adder for improving propagation delay.

KEYWORDS: Decimal matrix code (DMC), Encoder-reuse technique (ERT), Error correction codes (ECCs), Matrix codes (MCs)

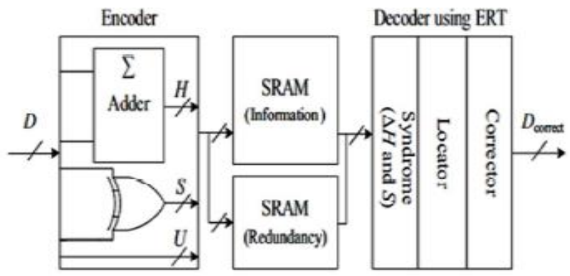
INTRODUCTION: The general idea for achieving error detection and correction is to add some redundant bits (i.e., some extra data) to a message, which receiver can use to check consistency of the delivered message, and to pick up data determinedto be corrupt. Error-detection and correction scheme can be either systematic or non-systematic: In a systematicscheme, 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, theBose– 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 overheadssince 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 aremore 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 (SBCDBD) 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 throughthis code is the possibility of miss-correction for a small subset of many errors. AS CMOS technology scales down to nanoscaleand memories are combined through an increasing number of electronic systems, the soft error rate in memory ase, especially when memories operate in space environments due to ionizing effects of atmosphere.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.

LITERATURE SURVEY:

Parallel Double Error Correcting Code Design to Mitigate Multi-Bit Upsets in SRAMs:A double-error correcting (DEC) ECC implementation technique suitable for SRAM applications is presented in this project. The upset distributions exhibit that MBU are the dominating contributor to overall soft error rate, and these MBU can range as large as 9-bits for LP and 13-bits for SF necessitating the usage of more powerful ECC schemes. Implementations of DEC and DEC-TED ECC using a parallel implementation approach in 90nm technology demonstrate that these codes can effectively be applied for SRAM applications.Protecting SRAM-based FPGAs Against Multiple Bit Upsets Using Erasure Codes:In this paper, we propose a generic scrub- bing scheme which reconstructs the erroneous con_guration frame based on the concept of erasure codes. Our proposed scheme does not require any changes to the FPGA architec- ture. Experimental results on a Xilinx Virtex-6 FPGA device show that the proposed scheme achieves error recovery cov- erage of 99.30% with only 3% resource occupation while the mean time to repair is comparable with previous schemes. Area Reliability Trade-Off in Improved Reed Muller Coding:In this paper we analyze the trade-off between the area and the reliability added in each chip employing the Reed Muller coding as the coding technique. We estimate the reliability and area increase of different orders of the Reed Muller decoding and observed that while the area increases, the reliability decreases. Our approach is to define a framework and help designers in order to decide on the configuration of the Reed Muller to be used. Finally, we provide aguideline to optimize the architecture making an optimal tradeoff between the area and the reliability. Multijunction Fault-Tolerance Architecture for Nanoscale Crossbar Memories:In this paper, we investigate a fault-tolerance scheme that utilizes redundancies in the rows and columns of a nanoscale crossbar molecular switch memory array. In particular, we explore the performance tradeoffs of time delay, power, and reliability for different amounts of redund-ancies. The results indicate an increase in fault-tolerance with small increases in delay and area utility.

DMC ENCODER:

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 wasmaximized so that the reliability of memory was enhanced. Besides, the encoder-reuse technique (ERT) wasimplemented 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.



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. It can be observed that the DMC encoder is also reused for obtaining the syndrome bits in DMC decoder. Therefore, the whole circuit area of DMC can be minimized as a result of using the existent circuits of encoder. Besides, this figure also shows the proposed decoder with an enable 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 the control of the write and read signals in memory. Therefore, in the encoding (write) process, the DMC encoder is only an encoder to execute the encoding operations. However, in the decoding (read) process, this encoder is employed for computing the syndrome bits in the decoder. These clearly show how the area overhead of extra circuits can be substantially reduced.In the proposed DMC, first, the divide-symbol and arrange-matrix ideas are performed, i.e., the N-bit word is divided into k symbols of m bits (N = k × m), and these symbols are arranged in a k1 × k2 2-D matrix (k = k1 × k2, where the values of k1 and k2 represent the numbers of rows and columns in the logical matrix respectively). Second, the horizontal redundant bits H are produced by performing decimal integer addition of selected symbols per row. Here, each symbol is regarded as a decimal integer. Third, the vertical redundant bits V are 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 proposed DMC does not require changing the physical structure of the memory. To explain the proposed DMC scheme, we take a 32-bit word as an example, as shown in Fig.4.2. The cells from D0 to D31 are information bits. This 32-bit word has been divided into eight symbols of 4-bit. k1 = 2 and k2 = 4 have been chosen simultaneously. H0–H19 are horizontal check bits; V0 through V15 are vertical check bits. However, it should be mentioned that 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 chosen. Therefore, 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 = 2×2 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 considered, so k = 2 × 8 and m = 4 are utilized to construct DMC.

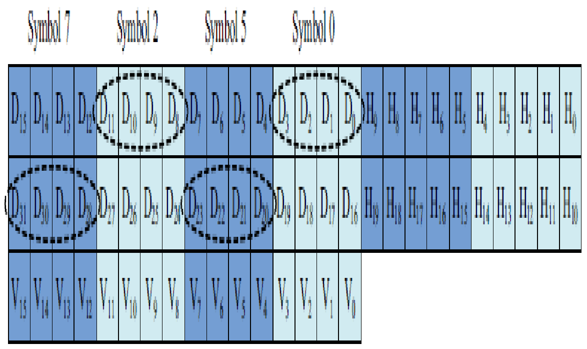

Symbol 7
Symbol 2
Symbol 5
Symbol 0
D15 D14 D13 D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 D1 D0 H9 H8 H7 H6 H5 H4 H3 H2 H1 H0
D31 D30 D29 D28 D27 D26 D25 D24 D23 D22 D21 D20 D19 D18 D17 D16 H19 H18 H17 H16 H15 H14 H13 H12 H11 H10
V15 V14 V13 V12 V11 V10 V9 V8 V7 V6 V5 V4 V3 V2 V1 V0

Fig: 3.2 32 bit word can be divided as 8 symbols with k=2x4 and m=4.

Horizontal redundant bits are calculated as follows

H9 H8 H7 H6 H5 = D7 D6 D5 D4 + D15 D14 D13 D12 (1)

H4 H3 H2 H1 H0 = D3 D2 D1 D0 + D11 D10 D9 D8 (2)

Similarly for the horizontal redundant bits H14 H13H12 H11 H10 and H19 H18H17H16 H15, where "+" represents decimal integer addition.

Vertical redundant bits are calculated as follows

$V0 = D0 \oplus D16$ (3)

$V1 = D1 \oplus D17$ (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, H19 − H0 are horizontal redundant bits, V15 − V0 are vertical redundant bits, and the remaining bits U31 − U0 are the information bits which are directly copied from D31 to D0

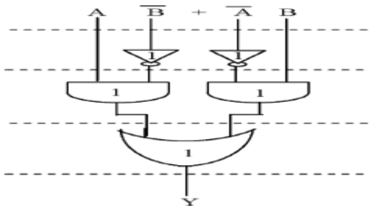
CARRY SELECT ADDER:

Design of area- and power-efficient high-speed data path logic systems are one of the most substantial areas of research in VLSI system design. In digital adders, the speed of addition is limited by the time required to propagate a carry through the adder. The sum for each bit position in an elementary adder is generated sequentially only after the previous bit position has been summed and a carry propagated into the next position.

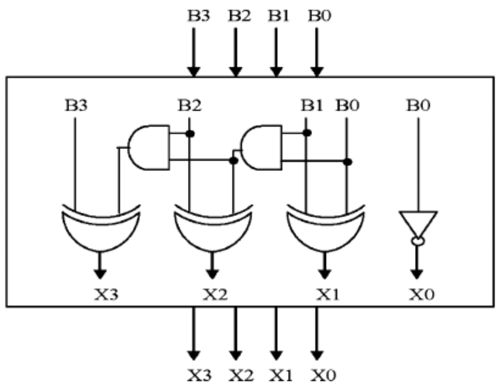
The CSLA is used in many computational systems to alleviate the problem of carry propagation delay by independently generating multiple carries and then select a carry to generate the sum [1]. However, the CSLA is not area efficient because it uses multiple pairs of Ripple Carry Adders (RCA) to generate partial sum and carry by considering carry input Cin=0 and Cin=1, then the final sum and carry are selected by the multiplexers (mux).

The basic idea of this work is to use Binary to Excess-1 Converter (BEC) instead of RCA with

Cin=0 in the regular CSLA to achieve lower area and power consumption [2]–[4].



.Delay and area evaluation of an xor gate



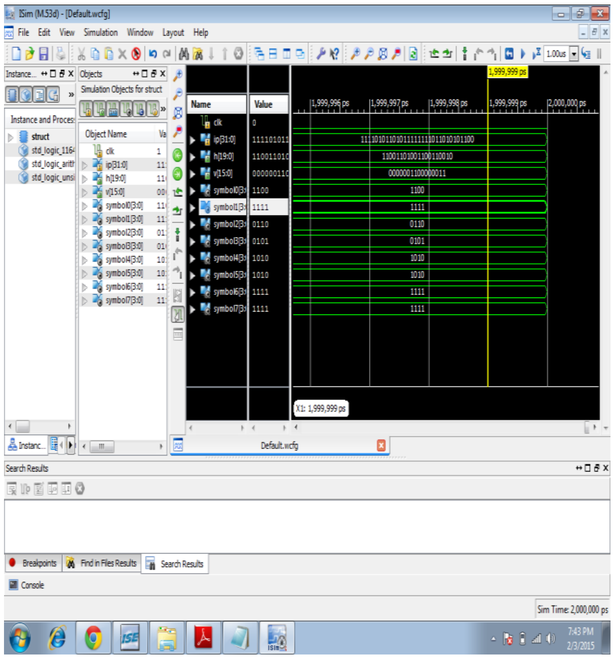
. The carry-ripple adder is composed of many cascaded single-bit full-adders. The circuit architecture is simple and area-efficient. However, the computation speed is slow because each full-adder can only start operation till the previous carry-out signal is ready. In the carry select adder, N bits adder is divided into M parts. Each part of adder is composed two carry ripple adders with cin_0 and cin_1, respectively. Through the multiplexer, we can select the correct output result according to the logic state of carry-in signal. The carry-select adder can compute faster because the current adder stage does not need to wait the previous stage's carry-out signal. The summation result is ready before the carry-in signal arrives; therefore, we can get the correct

computation result by only waiting for one multiplexer delay in each single bit adder. In the carry select adder, the carry propagation delay can be reduced by M times as compared with the carry ripple adder. However, the duplicated adder in the carry select adder results in larger area and power consumption.

AREA-EFFICIENT CARRY SELECT ADDER

The carry ripple adder is constructed by cascading each single-bit full-adder [1]. In the carryripple adder, each full-adder starts its computation till previous carry-out signal is ready. Therefore, the critical path delay in a carry ripple adder is determined by its carry-out propagation path. For an N-bit full-adder as illustrated in Fig. 1, the critical path is N-bit carry propagation path in the full-adders. As the bit number N increases, the delay time of carry ripple adder will increase accordingly in a linear way. In order to improve the shortcoming of carry ripple adder to remove the linear dependency between computation delay time and input word length, carry select adder is presented [2]. The carry select adder divides the carry ripple adder into M parts, while each part consists of a duplicated (N/M)-bit carry ripple adder pair, as illustrated in Fig. 2 as M=16 and N=4. This duplicated carry ripple adder pair is to anticipate both possible carry input values, where one carry ripple adder is calculated as carry input value is logic "0" and another carry ripple adder is calculated as carry input value is logic "1".When the actual carry input is ready, either the result of carry "0" path or the result of carry "1" path is selected by the multiplexer according to its carry input value. An example of 5-bit carry select adder is illustrated in Fig. 3. To anticipate both possible carry input values in advance, the start of each M part carry ripple adder pair no longer need to wait for the coming of previous carry input. As a result, each M part carry ripple adder pair in the carry select adder can comput paralel

RESULT:



```
Name          Value
clk           0
ip[31:0]      111101011   11110101101011111111011010101100
h[19:0]       110011010   11001101001100110010
v[15:0]       000000110   0000001100000011
symbol0[3:    1100        1100
symbol1[3:    1111        1111
symbol2[3:    0110        0110
symbol3[3:    0101        0101
symbol4[3:    1010        1010
symbol5[3:    1010        1010
symbol6[3:    1111        1111
symbol7[3:    1111        1111

1,999,996 ps  1,999,997 ps  1,999,998 ps  1,999,999 ps  2,000,000 ps
X1: 1,999,999 ps
```

CONCLUSION In this paper, novel per-word DMC was proposed to assure the reliability of memory. The proposed protection code utilized decimal algorithm to detect errors, so that more errors were detected and corrected. The obtained results showed that the proposed scheme has a superior protection level against large MCUs in memory. Besides, the proposed decimal error detection technique is an attractive opinion to detect MCUs in CAM because it can be combined with BICS to provide an adequate level of immunity.

CONCLUSION:

In this implemented project, novel per-word DMC was proposed to assure the reliability of memory. The protection code utilized decimal algorithm to detect errors, so that more errors were detected and corrected. The obtained results showed that the implemented scheme has a superior protection level against large MCUs in memory. Besides, the implemented decimal error detection technique is an attractive opinion to detect MCUs in CAM because it can be combined with CSA to provide an adequate level of immunity.

REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.