IMPACT: International Journal of Research in Engineering & Technology (IMPACT: IJRET)
ISSN(E): 2321-8843; ISSN(P): 2347-4599
Vol. 2, Issue 5, May 2014, 207-212

TRIPLE MODULAR REDUNDANCY LOW DELAY SINGLE ERROR CORRECTION CODE FOR PROTECTING DATA BITS

T. VENKATESH & A. SRIDEVI

Department of ECE, SNS College of Technology, Coimbatore, Tamil Nadu, India

ABSTRACT

Error correction codes are used for long years to protect memories from the soft errors. For a single bit error correction, the SEC single bit error correction code that correct one bit error per word are used. Double bit error detection code are used to detect the double bit errors. In the increasing of the technology the single bit error correction codes are used in the various places such as it is used to protect the registers. Suppose if we use the error correcting code means it affects the area delay added by the circuit. In case of the memory these are more important such that these extra bits are getting added to the each code word. For that the newly proposed codes are targeting to reduce the number of bits added by the code. In this paper a method to construct the low delay single error correction code is proposed. In order to increase the reliability of the circuit the Triple Modular redundancy is used. If the output of the decoder is taken three times at the input of the triple modular redundancy if any stuck at fault occurs on the data bits means it can be able to overcome the error and it will produce the correct output. If there is an error occur during one of the Triple modular redundancy method means also it can be able to recover the output correctly.

KEYWORDS: Error Correction Code (ECC), Single Error Correction Codes (SEC), Soft Errors

INTRODUCTION

Software Error Correction Code

When digital data is stored in non volatile memory, it is crucial to have a mechanism that can detect and correct a certain number of errors. Error correction code (ECC) encodes data in such a way that a decoder can identify and correct errors in the data. Typically, data strings are encoded by adding a number of redundant bits to them. When the original data is reconstructed, a decoder examines the encoded message to check for any errors. There are two basic types of ECC codes Block codes and convolutional. The block codes are referred to as “n” and “k” codes. A block of k data bits is encoded to become a block of n bits called a code word. In block codes, code words do not have any dependency on previously encoded messages. NAND Flash memory devices typically use block codes. The other type of code is Convolution codes. These codes produce code words that depend on both the data message and a given number of previously encoded messages. The encoder changes state with every message processed. Typically, the length of the code word is constant. Block codes are referred to as n and k codes.

A block of k data bits is encoded to become a block of n bits called a code word. A block code takes k data bits and computes (n - k) parity bits from the code generator matrix. The block code family can be divided in linear and non-linear codes. Either type can be systematic. Most block codes are systematic in that the data bits remain unchanged, with the parity bits attached either to the front or to the back of the data sequence. Linear Codes In linear block codes,

every linear combination of valid code words (such as a binary sum) produces another valid code word. In all linear codes, the code words are longer than the data words on which they are based. Micron NAND Flash memory devices use cyclic and Hamming linear codes

Error Detection Capability

For a code where dmin is the Hamming distance between code words, the maximum number of error bits that can be detected is t = (dmin – 1). This means that 1-bit and 2-bit errors can be detected for a code where dmin = 3 3.5

Error Correction Capability

For a code where dmin is the Hamming distance between code words, the maximum number of error bits that can be corrected is t = (dmin – 1) = 2. This means that 1-bit errors can be corrected for a code where dmin = 3. In coding theory, block codes comprise the large and important family of error-correcting codes that encode data in blocks. There is a vast number of examples for block codes, many of which have a wide range of practical applications. Block Codes are conceptually useful because they allow coding theorists, mathematicians, and computer scientists to study the limitations of all block codes in a unified way. Such limitations often take the form of bounds that relate different parameters of the block code to each other, such as its rate and its ability to detect and correct errors. Examples of block codes are Reed–Solomon codes, Hamming codes, Hadamard codes, Expander codes, Golay codes, and Reed–Muller codes. These examples also belong to the class of linear codes, and hence they are called linear block codes.

Error Detection Schemes

Error detection is most commonly realized using a suitable hash function (or checksum algorithm). A hash function adds a fixed-length tag to a message, which enables receivers to verify the delivered message by recomputing the tag and comparing it with the one provided. There exists a vast variety of different hash function designs. However, some are of particularly widespread use because of either their simplicity or their suitability for detecting certain kinds of errors (e.g., the cyclic redundancy check's performance in detecting burst errors).

Random-error-correcting codes based on minimum distance coding can provide a suitable alternative to hash functions when a strict guarantee on the minimum number of errors to be detected is desired. Repetition codes, described below, are special cases of error-correcting codes although rather inefficient, they find applications for both error correction and detection due to their simplicity.

SINGLE ERROR CORRECTION CODES

Existing Hamming Code Generation

Hamming code is a technique for detecting and correcting single bit errors in transmitted data. This technique requires that three parity bits (or check bits) be transmitted with every four data bits. The algorithm is called a (7, 4) code, because it requires seven bits to encoded four bits of data.

Background Concepts

In order to understand my implementation of Hamming codes, it helps to be comfortable with the concepts of matrix multiplication, modulo 2 arithmetic, and parity.

Modulo 2 Arithmetic

Modulo 2 arithmetic should be an easy concept to handle. Modulo 2 arithmetic is done in base 2 (binary), so the only numerals are 0 and 1. Since everything is modulo, there is no carrying either. Addition and multiplication are the only operators modulo 2 operators that we really care about. For maximum browser compatibility, It is represented these operations with an ordinary plus (+) and multiplication symbol (×).

Modulo 2 addition works as follows:

$0 + 0 = 0$

$0 + 1 = 1$

$1 + 0 = 1$

$1 + 1 = 0$

$A + B = B + A$

$A + B + C = (A + B) + C = A + (B + C)$

Modulo 2 multiplication works as follows:

$0 \times 0 = 0$; $0 \times 1 = 0$;$1 \times 0 = 0$;$1 \times 1 = 1$

$A \times B = B \times A$

$A \times B \times C = (A \times B) \times C = A \times (B \times C)$

The modulo 2 addition functions just like a logical Exclusive OR (XOR) and modulo 2 multiplication functions just like a logical AND.

Parity

In computer science terms, parity comes in two varieties even and odd. A string of bits (1's and 0's) has even parity if it contains an even number of 1's (the modulo 2 sum of the bits is 0), otherwise it has odd parity (the modulo 2 sum of the bits is 1). Many error detection schemes utilize parity bits. A parity bit is an extra bit that forces a binary string to have a specific parity. The parity bit for an even parity block may be computed by summing all the bits modulo 2. The parity bit for an odd parity block may be computed by summing all the bits modulo 2 and adding 1. In the hamming encoding the generator matrix is constructed by using the combination of the identity matrix and the and the parity check matrix. In the generator matrix the number of one is equal to the no of the exor gates used in the encoder section. The parity check matrix is constructed by taking the transpose of it. The data bits are getting multiplied with the generator Matrix we get the code word. The code word is getting multiplied with the parity check matrix if we are getting the zero vector means it is not having error. Suppose if we are having the error means it is compared with the every column of the parity check matrix the bit corresponding to the column is getting changed by using the NOT gate. The number of operations used here are more when compared with the proposed method.

CONSTRUCTION OF LOW DELAY SINGLE ERROR CORRECTION CODE

The proposed code tries to minimize the no of ones used in the parity check matrix. This is done by choosing the

no only two combination of ones in the parity check matrix. Thus this reduces the number of operations and the error can be easily detect it them the construction of them is shown in the simple structure as follows. The figure shows the reduced structure of the hamming code.

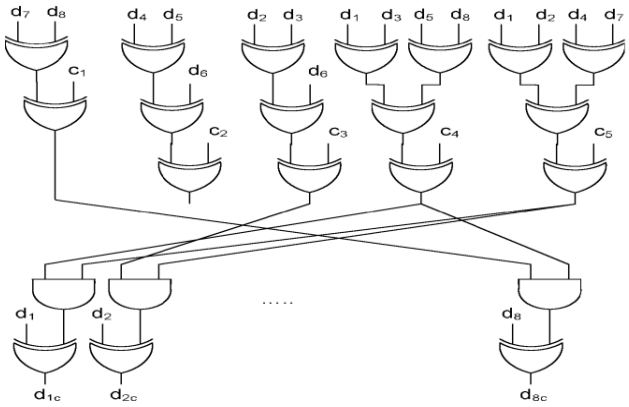

Figure 1: Structure of the Proposed Decoder

Thus the number of gates are reduced then to increase the reliability of the circuit. The soft error occurs on anyone of the data received bit is recovered. If the output of the decoder is taken three times at the input of the triple modular redundancy if any stuck at fault occurs on the data bits means it can be able to overcome the error and it will produce the correct output. If there is an error occur during one of the Triple modular redundancy method means also it can be able to recover the output correctly.

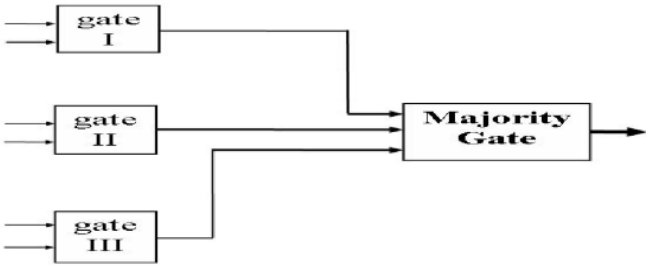

Figure 2: Triple Modular Redundancy

The Triple modular redundancy is an added advantage thus it increase the reliability of the circuit. The output of the decoder is taken into the three stages if we induce any stuck at fault at the following output the original data is recovered at the output section. The number of gates which are used here gets reduced Thus this is having the lot of advantage over the other methods, Thus this method performance is much accurate than any of the other single error correcting code methods.

EVALUATION

The proposed method is compared with that of the existing method the number of gates are more reduced. Then the reliability of them is also increased if any error occurred during the soft error means it is recovered. Thus this is having the lot of advantage over the other methods, Thus this method performance is much accurate than any of the other single error correcting code methods This significantly improves the chip areas but the size of the operation is getting reduced. Thus the process is performed on the Modelsim and the output is given as follows If there is an error occurred during one of the Triple modular redundancy method means also it can be able to recover the output correctly.

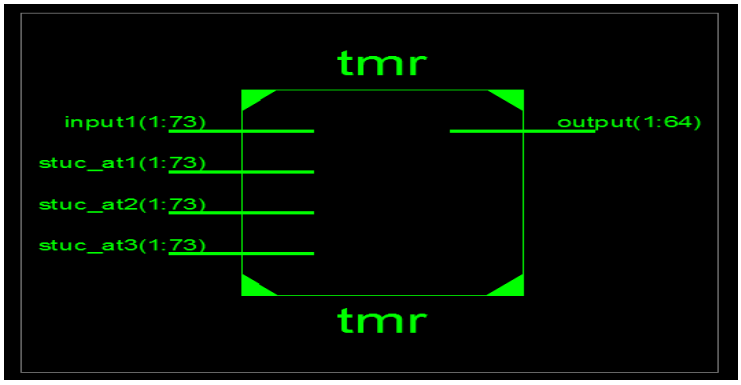

Figure 3: RTL Schemating Diagram TMR Low Delay Single Error Correction Code

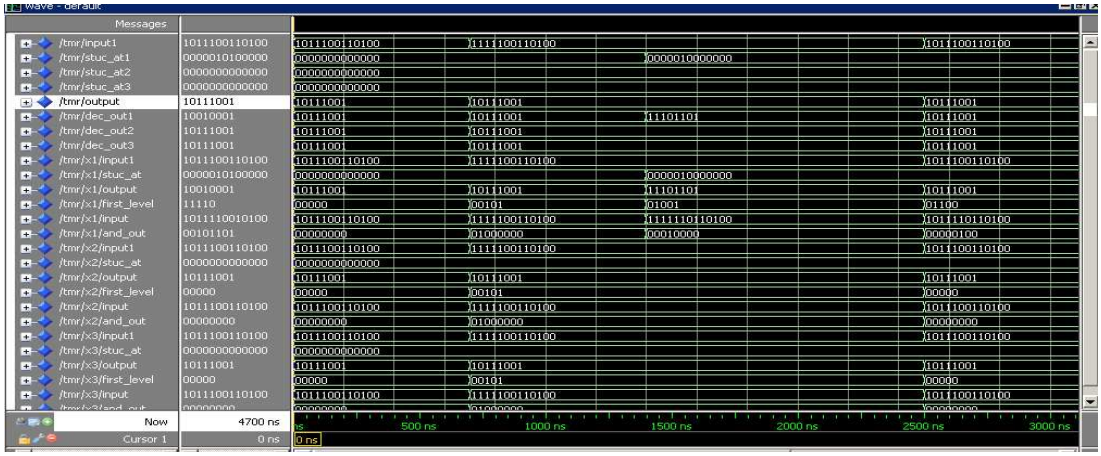

Figure 4: Output Waveform of the TMR Low Delay Single Error Correction Code

CONCLUSIONS

Thus in this paper a method to construct the low delay single error correction code with less operation is constructed further to improve the reliability of the circuit the triple modular redundancy is constructed. In this paper a single bit error detection and the correction is done this can be further extended to provide the double bit error detection. Thus this method sufficiently reduces the delay and the operation size of it. Thus this i\s the advantage of the proposed method which is called as the triple modular redundancy low delay single Error correction code. Thus this is the advantage of it If there is an error occur during one of the Triple modular redundancy method means also it can be able to recover the output correctly

REFERENCES

1. A Method to Construct Low Delay Single Error Correction Codes for Protecting Data Bits Only Pedro Reviriego, Salvatore Pontarelli, Juan Antonio Maestro, and Marco Ottavi, ieee transactions on computer-aided design of integrated circuits and systems, vol. 32, no. 3, march 2013.
2. C. L. Chen and M. Y. Hsiao, “Error-correcting codes for semiconductor memory applications: A state-of-the-art review,” IBM J. Res. Develop., vol. 28, no. 2, pp. 124–134, 1984.
3. G. C. Cardarilli, M. Ottavi, S. Pontarelli, M. Re, and A. Salsano, “Fault tolerant solid state mass memory for space applications,” IEEE Trans. Aerospace Electron. Syst., vol. 41, no. 4, pp. 1353–1372, Oct. 2005.

4. M. Y. Hsiao “A class of optimal minimum odd-weight column SECDED codes,” IBM J. Res. Develop., vol. 14, pp. 395–301, Jul. 1970.
5. R. W. Hamming, “Error detecting and error correcting codes,” Bell Syst. Tech. J., vol. 29, pp. 147–160, Apr. 1950.
6. V. Gherman, S. Evain, N. Seymour, and Y. Bonhomme, “Generalized parity-check matrices for SEC-DED codes with fixed parity,” in Proc. IEEE On-Line Testing Symp., Jul. 2011, pp. 198–20.
7. M. Richter, K. Oberlaender, and M. Goessel, “New linear SEC-DED codes with reduced triple bit error miscorrection probability,” in Proc. IEEE On-Line Testing Symp., Jul. 2008, pp. 37–42.
8. A. M. Saleh, J. J. Serrano, and J. H. Patel, “Reliability of scrubbing recovery-techniques for memory systems,” IEEE Trans. Reliab., vol. 39, no. 1, pp. 114–122, Apr. 1990.
9. S. Lin and D. J. Costello, Error Control Coding, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 2004.