iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 5 **Issue:** XI **Month of publication:** November 2017

DOI: http://doi.org/10.22214/ijraset.2017.11046

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor:6.887
Volume 5 Issue XI November 2017- Available at www.ijraset.com

Multi Fault Detection and Correction in Fault Tolerance Parallel Fats Using Bch Code

P.Pushpalatha[1], Sai Satish Inala[2]
[1,2]*Department of Electronics and Communication Engineering, University College of Engineering (A), JNTU Kakinada*

Abstract: ***A low complexity and error tolerant design has more demand in the signal processing systems. As technology increases as complexity of circuits also an increase that tends face reliability challenges and creates need for fault-tolerant implementations. Signal processing and communication circuits are affected by soft errors. The complexity increases as protection against soft errors increases in many applications. In communication systems data transfer protection coded with error correction codes (ECCs) has been proposed to efficiently protect data against soft errors. So far it has been evaluated only for single error correction codes. As technology increases multiple bit error are more frequent and thus more advanced protection is needed. In this case we have to implement the fault tolerance in order to preserve the data efficiently. ECCs is the one of the method to detect and correct errors in FFTs. This scheme evaluates protect data using double error correction (DEC) codes. This method allows most efficient protection when there is large number of parallel FFTs present. Results show that this DEC scheme reduces errors effectively and reduce the circuit overheads.***

Keywords: ***Data protection, soft errors, error correction codes (ECCs), single error correction, double error correction, parallel FFTs, fault-tolerant.***

I. INTRODUCTION

Soft errors induced by radiation particles have become one of the most challenging issues for modern digital circuits [1] [2]. For memories, which occupy a massive percent of the area of a chip, soft errors can lead to data corruption or even circuit malfunction [3]. Error correction codes (ECC) are extensively used to enhance the reliability and data integrity of the memories [4]-[6]. Single errors correction Hamming codes are commonly used. However, as the technology size scales down, multiple cell upsets have become an essential issue. In particular, double errors are the most common multiple errors pattern in memories [7]. Therefore, ECCs with multiple bit error correction functionality are now needed. One of the interesting Double error Correction (DEC) codes is the Bose-Chaudhuri-Hocquenghem (BCH) code [5], which requires a small number of parity bits and thus costs a low memory overhead. Its main disadvantage is that decoding is complex.

The electronics circuit is used in many applications. This application reliability is critical parameter like speed, power consumption and area. Components used in CMOS technology increases that tend to reliability problems. Faults are occurring in these systems to be reduced or eliminated by using fault-tolerance techniques. Some techniques are used to detect and correct errors to add redundancy at logic or system level to make sure that errors do not affect the functionality of the system [8]. Algorithmic-based fault tolerance (ABFT) is one of the techniques to detect and correct errors [9]. Over the years many ABFT techniques have been proposed to protect the blocks that are commonly used in the circuits. The absence of parallel filters or FFTs creates an opportunity to implement ABFT techniques for the whole group of parallel modules instead of for each one independently. Several works have considered the protection of FFTs [10][11]. In signal processing and communication circuits become more difficult, it is common to find several FFTs operating in parallel. More recently a general scheme based on the use of error correction codes has been proposed. This scheme is use Hamming code to detect and correct errors in circuit. By using Hamming code only correct single error but detect double errors. We propose another ECC code that is BCH code. it is possible to design binary BCH codes that can correct multiple bit errors. In this paper, the effectiveness of data protection against soft errors is evaluated in the presence of double errors using BCH codes in fault tolerance system. The results show that it can significantly reduce the circuit overhead.The rest of this letter is organized as follows. Section II reviews the previous fault tolerance parallel FFTs using Hamming code. The data protection scheme for DEC codes is explained in Section III, and evaluated in Section IV. Finally, the conclusions can be found in Section V.

II. FAULT TOLERANCE PARALLEL FFTS USING HAMMING CODE

The technique is well known to protect parallel FFTs developed by using the ECCs [10]. Here using simple single error correction Hamming code. This scheme is shown in single in fig. 1.

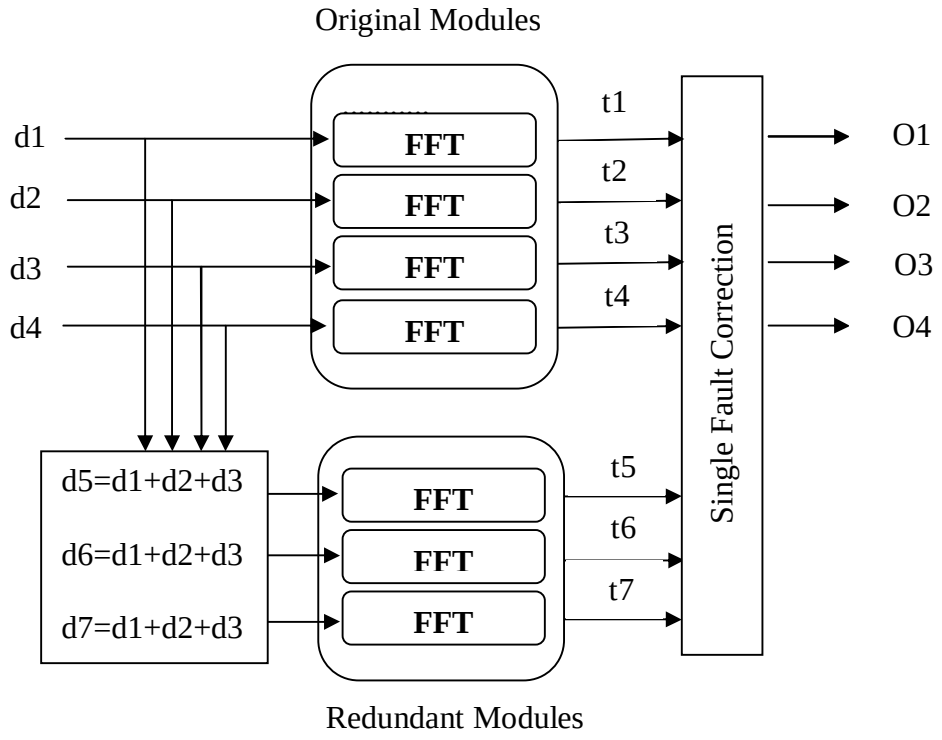


Fig. 1. Fault Tolerance Parallel FFTs using SEC

This system is consists of total seven FFT module. Here four FFTs is original module and below three modules is parity check or redundant modules. By using parity check modules to detect and correct errors. The inputs to the three redundant modules are linear combinations of the inputs and they are used to check linear combinations of the outputs. For example, the input to the first redundant module is

$$d5 = d1 + d2 + d3 \quad (1)$$

and since the DFT is a liner operation its output t5 can be used to check that

$$t5 = t1 + t2 + t3. \quad (2)$$

Error correction codes (ECC) takes some number of k bits to input and produces n bits output by adding some parity check bits. The number of parity check bits added by n-k bits. The parity check or redundancy bits are nothing but XOR combinations of the given k data bits.

The main focus is to properly design those combinations for exact error detection and correction. For example, let us consider a simple Hamming code [12]. Hamming rule is expressed by the following inequality.

$$2^p >= d + p + 1 \quad (3)$$

Here d = number of information data.

P = number of parity bits or redundancy bits.

Consider we want to transmit 7 bit information from transmitter section and encoding 11bits of information. To transmit 11 bit information data, 4 parity bits are needed. Calculation of these parity bits are given below.

$$P(1) = D_3 \oplus D_5 \oplus D_7 \oplus D_9 \oplus D_{11}$$
$$P(2) = D_3 \oplus D_6 \oplus D_7 \oplus D_{10} \oplus D_{11}$$
$$P(4) = D_5 \oplus D_6 \oplus D_7$$
$$P(8) = D_9 \oplus D_{10} \oplus D_{11} \quad (4)$$

The message information data taken as D=0001001. Resulting Output code word is C=00001001100.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor:6.887
Volume 5 Issue XI November 2017- Available at www.ijraset.com

With k = 4 and n = 7. In this case, the three parity check bits p1, p2, p3 are linear combinations of the data bits d_1, d_2, d_3, d_4 as follows:

$$p1 = d_1 \oplus d_2 \oplus d_3$$
$$p2 = d_1 \oplus d_2 \oplus d_4$$
$$p3 = d_1 \oplus d_3 \oplus d_4 \quad (5)$$

In the example, an error on d_1 will lead to cause errors on the three parity checks. Similarly an error on d_2 will effects only in p1 and p2. Error on d_3 will cause error in p1 and p3; and finally an error on d_4 will results an error in p2 and p3. Therefore, the data bit which is in error can be located and that error can be corrected. This is usually done in terms of the generating G and parity check H matrixes. In the example, those are given by

$$G = \begin{bmatrix} 1000111 \\ 0100110 \\ 0010101 \\ 0001011 \end{bmatrix} \quad H = \begin{bmatrix} 1110100 \\ 1101010 \\ 1011001 \end{bmatrix} \quad (6)$$

A. Location of error in hamming code

TABLE I

s1s2 s3	Error Bit Location
0 0 0	No error
1 1 1	t1
1 1 0	t2
1 0 1	t3
0 1 1	t4
1 0 0	t5
0 1 0	t6
0 0 1	t7

From the equation (2) it can be observed that is denoted as s1 check. Same way applies to remaining to redundant filters that will provide checks s2 and s3. Based on three checks, the module on which the error has occurred can be determined. The different errors are summarised in above Table I.
Checking is done by testing if

$$Z_1[n] = t_1[n] + t_2[n] + t_3[n]$$
$$Z_2[n] = t_1[n] + t_2[n] + t_4[n]$$
$$Z_3[n] = t_1[n] + t_3[n] + t_4[n]. \quad (7)$$

the error is detected, the error can be corrected by reconstructing its output by using remaining modules. For example, when an error on y_3 is detected, it can be corrected by making

$$O_{c3}[n] = z_3[n] - t_1[n] - t_4[n] \quad (8)$$

Similarly remaining correction equations can be used correct errors on the other modules. Here using hamming code only correct single error on module. Extended Hamming and Hsiao codes are most common in fast single bit error detection. These codes are not capable of triple error detection. More advanced ECCs can be used to correct errors on the multiple modules is the needed in given application.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor:6.887
Volume 5 Issue XI November 2017- Available at www.ijraset.com

III.FAULT TOLERANCE PARALLEL FFTS USING BCH CODE

BCH codes is an abbreviation for Bose, Ray- Chaudhuri, Hocquenghem, invented in 1960s and today they are used as a baseline for many recent Error Correcting codes. They are the powerful class of multiple error correction codes with well-defined mathematical properties. In order to deal with double errors that are the primary multiple bit error pattern in modern-day technology nodes, memories should be protected by ECCs with multiple bit error correction functionality. For example, DEC BCH codes just require a small number of parity bits [13]. This means that the data protected by DEC codes costs low area overhead. However, the decoder circuitry of these codes is complex. In above scheme is implemented with one of the error correction code that is hamming code. This scheme is correct single error only and detects double error. In this case study we implemented with another error correction code that is BCH code, it can correct multiple errors.

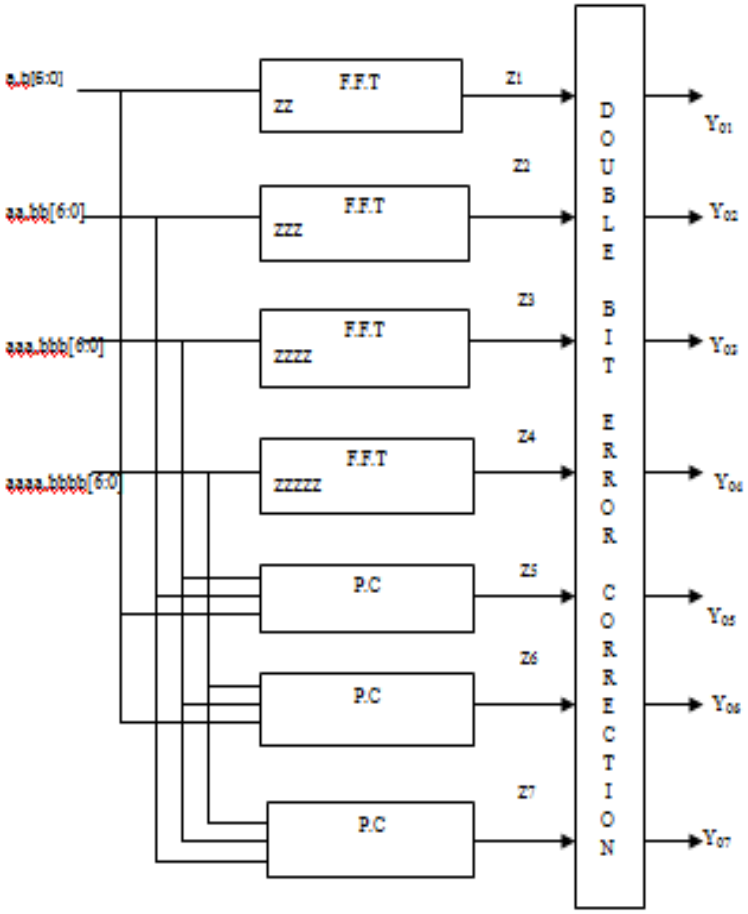


Fig 2. Parallel FFT protection using DEC.

It is assumed that there is only a double error on the system at any given point in time. There are three main contributions. They are

A. Error Correction Code is assessed to protect the parallel FFTs which show its effectiveness in terms of overhead and protection effectiveness.

B. A new technique is proposed based on the use of Parseval or sum of squares (SOSs) checks combined with parity FFT.

C. A new technique is proposed on which the ECC (BCH Code) is used on the SOS checks instead of the FFTs.

This scheme is evaluated by using FPGA implementations to assess the protection overhead. The protection overhead can be reduced by combining the use of ECCs and parseval checks.

IV.RESULTS

Simulation is carried out using Xilinx ISE 13.2 design suit tool with Spartan 3E as the target device. Simulation results of (15,5) BCH decoder are discussed below.

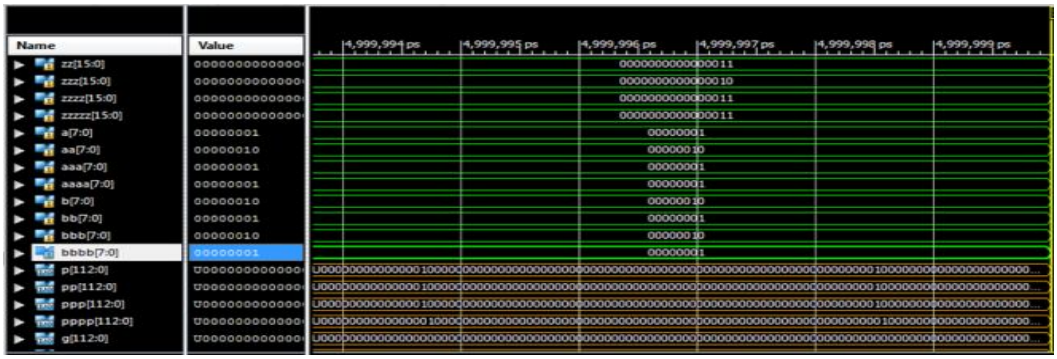


Fig 3. Input waveform

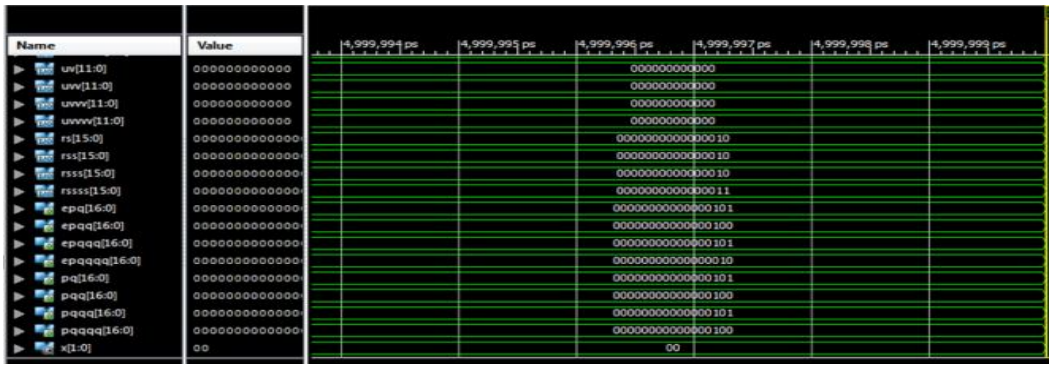


Fig 4. Output waveform

Table I shows the results of hamming and BCH code in Fault tolerance system. Here compare the area and time delay of these two codes in system. From the table it can be seen that the BCH code with fault tolerance parallel FFTs shows significant reductions in circuit area and time delay.

TABLE I
RESOURCE COMPARISON ANALYSIS OF HAMMING AND BCH CODE IN FAULT TOLERANCE PARALLEL FFTS

Logic Utilization		Fault Tolerance Parallel FFTS using	
		Hamming Code	BCH Code
AREA	Number of Slices	49%	40%
	Number of 4 input LUTs	41%	34%
	Number of Slice flip flops	8%	6%
Delay		58.04ns	42.03ns

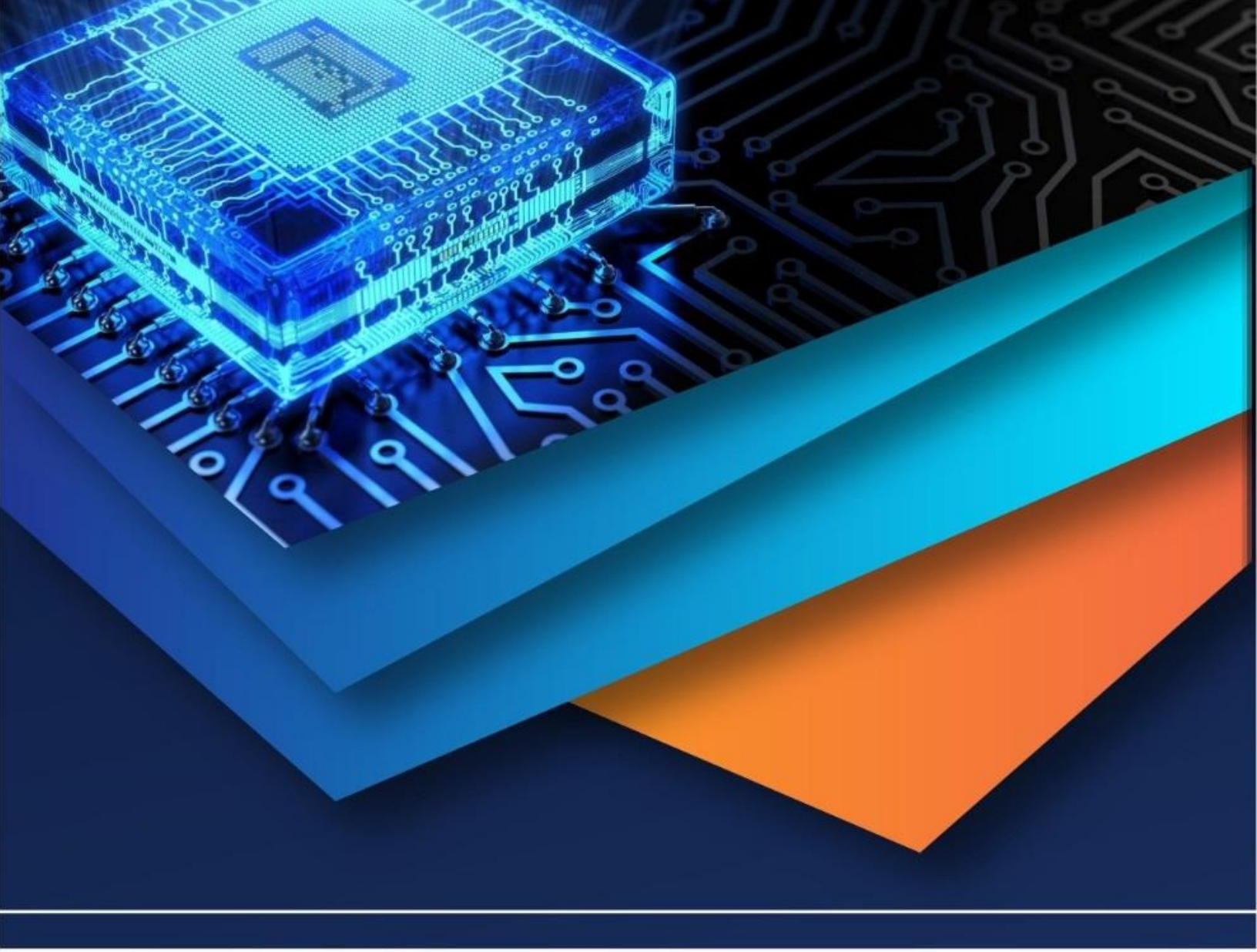
V. CONCLUSIONS

In the all above sections concisely studied regarding the protection against soft errors. This letter has presented a method to detect and correct multi-error using BCH codes fault tolerance system. The multi error correction in fault tolerance parallel FFTs using BCH code shows that BCH code is more efficient for data communication and protection because of its low coding complexity and high coding rate. In point of area and delay fault tolerance parallel FFTs using BCH code is best for low number of bits. Hamming code is computationally simple and understanding and has the capacity to detect up to three errors but can correct only one error per message. Hamming codes are relatively efficient when sending small amount of data but they get increasingly inaccurate as the

number of bits increases. So BCH code is better than hamming code. Therefore, fault tolerance parallel FFTs using BCH code is an interesting option to protect against double errors.

REFERENCES

[1] R. C. Baumann, "Radiation-induced Soft Errors in Advanced Semiconductor Technologies," IEEE Trans, Device Mater. Reliab., vol. 5, no. 3, pp. 301-316, Sept. 2005.
[2] P. Hazucha and C. Svensson, "Impact of CMOS Technology Scaling on the Atmospheric Neutron Soft Error Rate," IEEE Trans. Nucl. Sci., vol. 47, no. 6, pp. 2586-2594, Dec. 2000.
[3] R. D. Schrimpf and D. M. Fleetwood, Radiation Effects and Soft Errors in Integrated Circuits and Electronic Devices. Singapore: World Scientific, 2004.
[4] M. Blaum, R. Goodman and R. McEliece, "The Reliability of Single Error Protected Computer Memories", IEEE Trans. on Computers, vol. 37, no. 1, pp. 114–119, Jan. 1988.
[5] 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.
[6] S.S. Liu, P. Reviriego, L.Y. Xiao, J.A. Maestro "Reducing the Cost of Triple Adjacent Error Correction in Double Error Correction Orthogonal Latin Square Codes", IEEE Trans, Device Mater. Reliab., vol. 16, no. 2, pp. 269-271, Mar. 2016.
[7] S. Baeg, S. Wen, and R. Wong, "SRAM Interleaving Distance Selection with a Soft Error Failure Model," IEEE Trans. Nucl. Sci., vol. 56, no. 4, pp. 2111–2118, Aug. 2009.
[8] M. Nicolaidis, "Design for soft error mitigation," IEEE Trans. Device Mater. Rel., vol. 5, no. 3, pp. 405–418, Sep. 2005.
[9] A. L. N. Reddy and P. Banerjee, "Algorithm-based fault detection for signal processing applications," IEEE Trans. Comput., vol. 39, no. 10, pp. 1304–1308, Oct. 1990.
[10] Zhen Gao, Pedro Reviriego, Zhan Xu, Xin Su, Ming Zhao, Jing Wang, and Juan Antonio Maestro, "Fault Tolerant Parallel FFTs Using Error Correction Codes and Parseval Checks", IEEE Trans. Vlsi, vol. 24, no. 2, pp. 1063–8210, Feb. 2016.
[11] J. Y. Jou and J. A. Abraham, "Fault-tolerant FFT networks," IEEE Trans.Comput., vol. 37, no. 5, pp. 548–561, May 1988.
[12] R. W. Hamming, "Error detecting and error correcting codes," Bell Syst. Tech. J., vol. 29, no. 2, pp. 147–160, Apr. 1950.
[13] Shanshan Liu, Pedro Reviriego and Liyi Xiao "A Evaluating Direct Compare for Double Error Correction Codes," IEEE Transactions on Device and Materials Reliability., vol. 20, pp. 1530-4388 , Aug. 2017.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY