ENHANCED FAULT TOLERANT PARALLEL FFTS USING ERROR CORRECTION CODES AND PARSEVAL CHECKS USING ANCIENT VEDIC SUTRA

K SREENIVASULU NAIK[1] M.MADHU BABU[2]
[1]MTech, Department of ECE, JNTUA College of Engineering , Anantapur, Andhra Pradesh,
[2]Department of ECE, JNTUA College of Engineering , Anantapur, Andhra Pradesh.

Abstract: Every year the complexity of communication systems and signal processing circuits will increase. This can be done by the CMOS technology scaling that allows the integration of too many transistors on a single device. This improved complexity will leads to failure of circuits. Identically, the transistors will run with low supply and are leads to errors due to noise and variations in the manufacturing. Hence, soft errors also increase as technology scaling increases. A soft error alters the actual value of the system which will disturb the electronic circuits operation. In order to overcome such situation, the best choice is to utilize Algorithmic-Based Fault Tolerance (ABFT) techniques for some applications. ABFT use algorithmic properties to locate and correct errors. ABFT technique is suitable for communication and signal processing applications. Fast Fourier Transforms (FFTs) is one of the examples which are major building blocks in many systems. To locate and correct errors in FFTs, many protection schemes have been proposed. Those protection schemes are Error Correction Codes and Parseval or Sum of Square (SOS) checks. These techniques are first applied to protect FFTs. Then by combining both error correction codes and Parseval or SOS checks a new improved protection scheme have been proposed and evaluated. The proposed scheme results in reduced computational time and implementation cost of protection.

Index Terms **–** Error correction codes (ECCs), SOS checks, FFTs, Vedic multiplier, soft errors.

I.INTRODUCTION

A filter is an electronic device that removes components or features of a signal. Filters are typically integrated in electronic circuits to pass all signals in bound frequency ranges and reject signals outside the bound frequency ranges. In circuit theory, a filter is connected with an electrical circuit that changes the amplitude characteristics of a signal with certain range of frequencies. Ideally, a filter won't add new frequencies to the input signal; instead it will modify or reject the unwanted frequencies of that signal. Nowadays filters used in wide range of applications that supported automotive, medical, and home-appliances. The techniques of filtering have been proposed because of the behavioral properties of signal changes. Filters are classified into two types based on the type of elements used in the circuits: Digital and analog filters. In signal process, a digital filter acts as a device to remove unwanted components or features from a signal. Digital filters are mainly used to separate the signals that are combined and also to restore the signals that are distorted. Most often, this suggests removing some frequencies and not others so as to suppress interfering signals and scale back background signal.

In present technology, we commonly found that several FFTs operating parallel in most of the systems. This parallel operation is exploited for fault tolerance. Specifically, soft errors are a very important issue and lots of schemes are intended over the years to mitigate them. In this brief, the protection of FFTs is studied. The schemes which are used to detect and correct errors are:

1) Evaluation of ECC technique using hamming code.
2) Evaluation of Parseval checks along with a redundant bit called as Parity-SOS technique.
3) The new scheme on which the use of ECC on the parseval checks instead of using parseval check on each FFT. This is called as Parity-SOS ECC technique.

When compared to other techniques the proposed technique is more area efficient.

Additionally, a Vedic multiplier is used in the implementation of FFT architecture in order to reduce the

computational time and to increase the accuracy when compared to other techniques.

II. LITERATURE SURVEY

The fault tolerant systems based on Error Correction Codes (ECCs) using Verilog is designed, implemented, and tested. It proposes that wherever the presence of parallel filters (FFTs) is seen have been protected using Error Correction Codes (ECCs). Here the Hamming Codes are used for error correction which consists of k bits and generates n bits by adding n−k parity check bits. These parity bits are generated by using r = 1+ $\log_2$ (k); where r is redundant (parity) bits; k is original data bits. In this scheme the redundant module is used in order to store the data and parity check bits and these data can be recovered afterword even if an error in any one of the bits is recognized. This is done by re -computing the parity check bits and comparing the results with the values stored. In this way using hamming codes errors within the circuit can be located and corrected in [1].

In this paper, we have proposed architecture for the implementation of fault -tolerant computation within a high throughput using SOSs checks combined with ECCs. The area overhead is minimized by reducing the redundant bits and usage of parseval checks per FFT. In this scheme, we have used SOSs checks on ECCs instead of SOSs check per FFT and one parity bit for overall circuit. Where SOSs are used to detect errors and the parity bit is used to correct the errors. Further to improve the performance of the system a Vedic multiplier is used in the implementation of FFT architecture. The Vedic multiplier minimizes the number of calculations and increases the accuracy. This paper has presented a detailed analysis of single fault correction using ECCs and SOSs checks in filters (FFTs).

III. ERROR TOLERANT TECHNIQUES FOR PARALLEL FFTS

Error Correction Using Hamming Codes

In a discrete time filter, $h[n]$ is an impulse response that performs the operation on the incoming signal $x[n]$ to produce the output signal y[n] as shown below:

$$y[n] = \sum_{l=0}^{\infty} x[n-l]h[l] \qquad (1)$$

This property can be exploited in the case of parallel filters that operate on different incoming signals, as shown in Figure 1. The protection scheme uses the Error Correction Hamming Code for single error correction. In this case, the system consists of four FFTs with $x_1[n]$, $x_2[n]$, $x_3[n]$, and $x_4[n]$ as input signals to produce four outputs $z_1[n]$, $z_2[n]$, $z_3[n]$, and $z_4[n]$ where each FFT can be equivalent to a bit. This is shown in Figure 1, where three redundant FFTs are used to form the parity check bits to detect and correct errors. The number of redundant (parity) bits used in this can be generated using r = 1+$\log_2$ (k); where r represents number of redundant (parity) bits; n represents the input bits. Here, each FFT is equivalent to 1 bit, then for 4 FFTs (4 bits) we require 3 redundant modules (3 bits).

The inputs to the 3 redundant FFTs are the linear combination of the original inputs i.e., $x_5[n]$, $x_6[n]$, $x_7[n]$ which corresponds to the outputs $z_5[n]$, $z_6[n]$, and $z_7[n]$. For example, the input to the primary redundant FFT is

$$X_5[n] = X_1[n] + X_2[n] + X_3[n] \qquad (2)$$

then its output z5 can be used to check that

$$Z_5[n] = Z_1[n] + Z_2[n] + Z_3[n] \qquad (3)$$

This can be represented as c1 check. Similarly, the remaining two redundant modules gives c2 and c3 checks. By comparing these checks with the original outputs, the module on which the error occurred can be determined. The error locations using hamming code are shown in Table I. Once error on which module has occurred is known, then it can be recovered with the help of remaining modules. For example, if an error occurred in z_1, then the reconstruction is carried as follows:

$$Z_{1c}[n] = Z_5[n] - Z_2[n] - Z_3[n] \qquad (4)$$

Similarly we can correct the errors on the other modules. The overhead of this scheme is the usage of redundant FFTs. Here 3 redundant FFTs are required to protect 4 FFTs, but for eleven FFTs only 4 redundant FFTs are needed. This ensures that the overhead decreases as the number of FFTs increases.

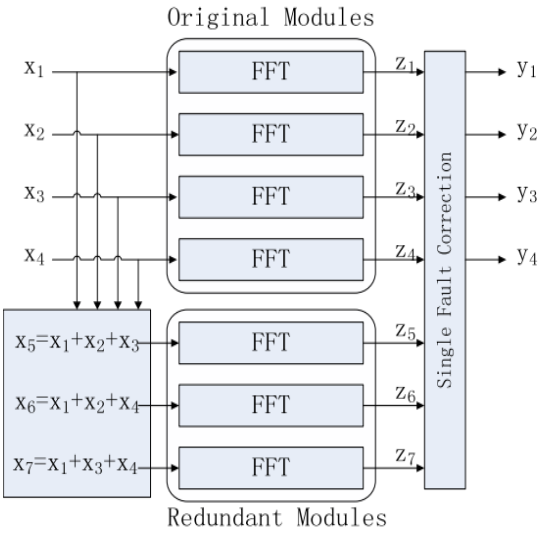


Fig. 1 ECC-based scheme for four filters using Hamming code.

TABLE I
THE HAMMING CODE ERROR LOCATION

$C_1C_2C_3$	Error Bit Position
0 0 0	No Error
1 1 1	Z_1
1 1 0	Z_2
1 0 1	Z_3
0 1 1	Z_4
1 0 0	Z_5
0 1 0	Z_6
0 0 1	Z_7

Fault tolerant FFT based on Parseval's check

Many techniques are proposed to protect the FFTs which are arranged in parallel. Parseval's technique is one amongst those techniques which is based on the Parseval theorem used to detect errors. This scheme states that Sum of Squares (SOSs) of inputs to the FFT is equal to the Sum of Squares (SOSs) of the outputs. This technique can be used to detect the errors with minimum overhead. For parallel FFTs, the SOS checks are often combined with the Error Correction Codes (ECCs) to minimize the overhead. Where Parseval's check is used only for detect errors and ECCs is used for error correction. This error correction can be achieved by using a single parity (redundant) bit for all the original FFTs. If any error is detected, then the output of parity FFT is used to correct the error. Hence, it is called as Parity-SOS which is illustrated in Figure2. By using this scheme multiple errors detection and correction is possible. If an error is detected using p_1, p_2, p_3, p_4 then it can be recovered by the parity FFT output by performing XOR operation with the outputs of the remaining FFTs.

The main advantage of this technique as compared to the previous schemes is that it can reduce the number of additional FFTs to one and also reduces the overhead to protect FFTs.

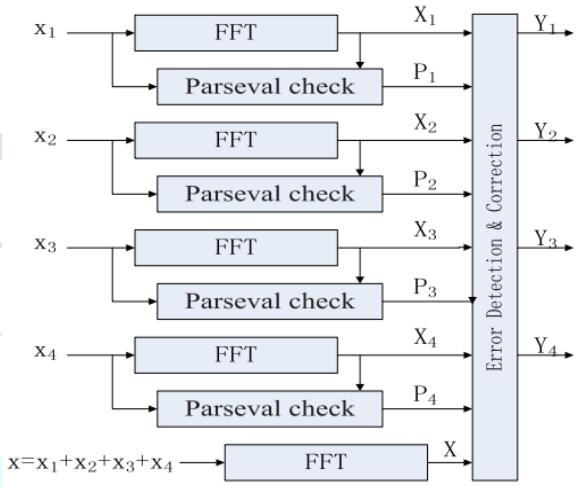


Fig. 2 Parity-SOS Scheme (first technique)

IV. PROPOSED PROTECTION SCHEMES FOR PARALLEL FFTS

Another alternative is to combine both the Parseval's (SOSs) check and ECC techniques i.e., simply the use of SOS checks on ECC instead of using an SOS check on each FFT which is illustrated in Figure 3. Further to correct the errors include an additional parity FFT which is similar to the Parity-SOS scheme. Where ECC scheme detects the errors and Parity-SOS scheme is able to correct those errors. Hence, it is called as Parity-SOS ECC technique. As compared with Parity-SOS scheme the main advantage of this technique is to minimize the required number of SOS checks.

In case if any module is affected by an error, then it can be detected by using P1, P2, P3, P4 checks. Later it can be corrected by re-computing the error caused FFT with the help of redundant (parity) FFT output and the remaining FFT outputs.

For an example, if an error occurred in the primary FFT can be detected by P1and the correction of error is done by using

$$X_{1c} = X - X_2 - X_3 - X_4 \quad (5)$$

The internal structure of Parseval (SOSs) checks is illustrated in Figure4. It consists of magnitude square, accumulator and magnitude comparator.

The overheads of these schemes can be calculated based on the number of SOS checks and redundant FFTs used in the implementation. This is illustrated in Table II for k FFTs, where k is a power of 2.

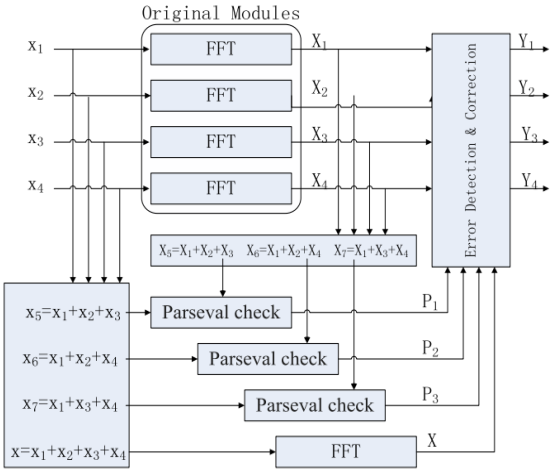


Fig. 3 Parity-SOS ECC Scheme (second-technique).

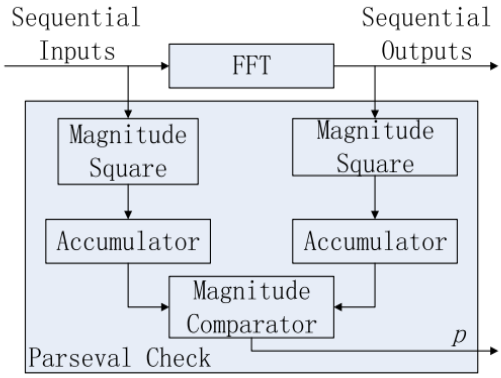


Fig. 4 Architecture of the SOS check.

TABLE II
OVERHEAD OF THE DIFFERENT SCHEMES TO PROTECT k FFTs

	FFTs	SOS checks
ECC	$1+\log_2(k)$	0
Parity-SOS	1	K
Parity-SOS-ECC	1	$1+\log_2(k)$

V. Vedic Sutra – Urdhva Tiryagbhyam

Urdhva Tiryagbhyam is used for multiplication of numbers. It is considered as the easiest multiplication method. So, in proposed system we replaced the multiplier by Vedic multiplier. By doing this we are able to get less power consumption, high accuracy and reduced delay.

Vedic mathematics consists of sixteen simple mathematical formulae's. These sixteen Vedic Sutras handle and cover nearly each part of arithmetic. They even applied to advanced issues which involves a huge variety of operations in mathematics. Among these sutras, Urdhva Tiryagbhyam sutra is a simple "Vertical and Crosswise" Multiplication. The main feature of it is minimized number of calculations, reduced computational time, reduced space and is applicable in all cases of multiplications. The implementation of Vedic multiplier consists of both half adders and full adders along with logical AND gate to perform multiplication.

Consider two 4-bit numbers, A (A3-A0) and B (B3-B0) and their Product is represented by P (P7-P0). In Figure5, the step by step multiplication of two 4-bit numbers using Urdhva Tiryagbhyam sutra is shown. Here, multiplication operation is simply performed by addition of partial products. In this both generation of partial products and their sum are performed simultaneously. Hence, it is also known as Parallelism architecture. In the typical style of Urdhva Tiryagbhyam, the logical AND gate is used for generating partial products and the addition of these partial products is done by using half-adders and full-adders.

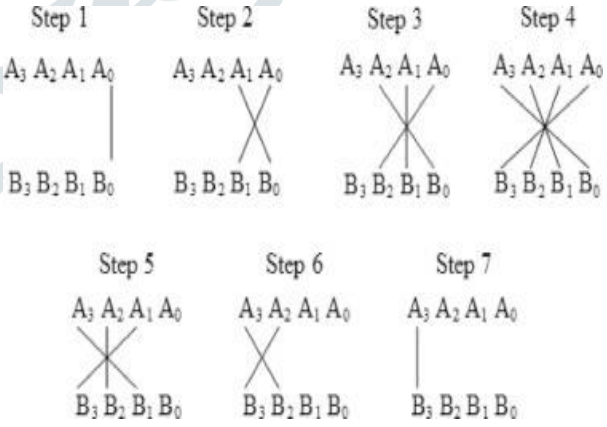


Fig. 5 s 4-bit binary multiplication using Urdhva Tiryagbhyam Sutra

VI.RESULTS

The written Verilog HDL Modules have successfully simulated and verified using Modelsim III 6.4b and synthesized using Xilinx ISE 13.2.

Simulation Result:

Inputs: X1 X2, X3, X4

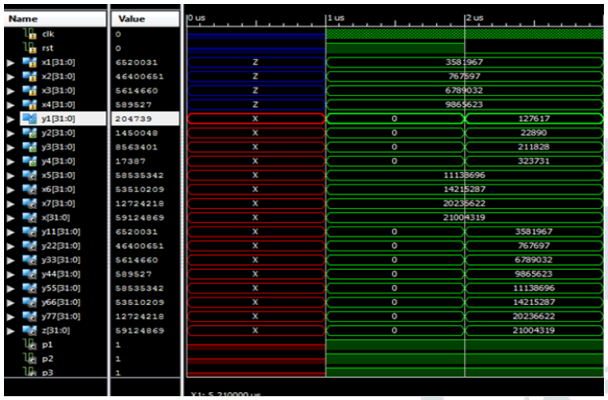
Outputs: Y11, Y22, Y33, Y44

Redundant/Parity input: X

Redundant Output: Z

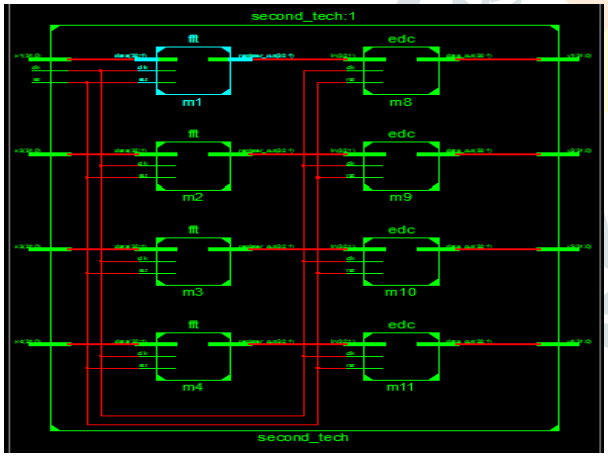
SOS Check inputs: X5, X6, X7, Y55, Y66, Y77

SOS Check outputs: P1, P2, P3

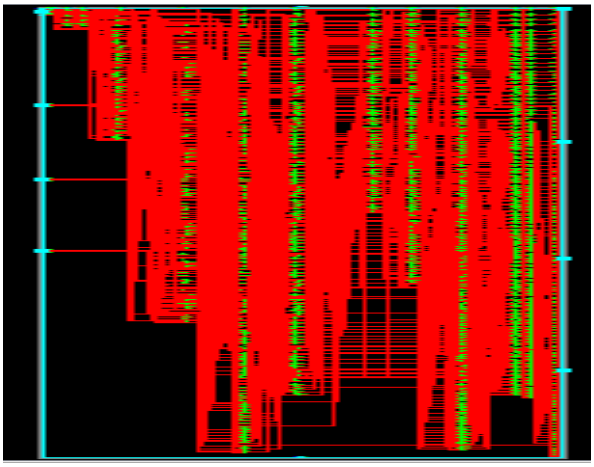


Synthesis Results:

RTL Schematic:



Technology Schematic:



Design Summary:

Device Utilization Summary (estimated Values)			
Logic Utilization	Used	Available	Utilization
No. of Slice Registers	228	4800	4%
No. of Slice LUTs	1180	2400	49%
No. of fully used LUT-FF pairs	104	1304	7%
No. of bonded IOBs	254	102	249%
No. of BUFG/ BUFGCTRLs	1	16	6%

Timing Report:

```
Timing Summary:
---------------
Speed Grade: -3

   Minimum period: 6.395ns (Maximum Frequency: 156.369MHz)
   Minimum input arrival time before clock: 3.712ns
   Maximum output required time after clock: 3.597ns
   Maximum combinational path delay: No path found
```

VII. CONCLUSIONS

Recognizing and correcting errors are difficult in communication and signal processing systems that will expand the use of enhanced fault tolerant implementation. Several filters which are used to implement the modern signal processing circuits are arranged in parallel manner. The proposed technique is used to protect these parallel FFTs by detecting and correcting the signal errors. The methodology uses SOS-ECC scheme to the outputs of

parallel FFTs for detecting and correcting errors. The Parseval or SOS checks can detect and locate errors whereas ECC is a simple parity (redundant) FFT to correct errors. Using the proposed technique the 8 point FFT with 32 bit input is protected. For the further improvement the Vedic multiplier which is based on Urdhva Tiryagbhyam Sutra is used in the architecture of FFT implementation. This Vedic multiplier increases the system performance by minimizing the number of calculations, reducing the computational time. Only single bit errors can be detected and corrected using this technique. And due to the usage of Vedic multiplier in the implementation of this technique it further reduces the area, computational time. As compared to existing techniques overall it results in high speed.

REFERENCES

[1] Z. Gao *et al.*, "Fault tolerant parallel filters based on error correction codes," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 24, no.2, pp. 769-773, Feb. 2016.

[2] E. P. Kim and N. R. Shanbhag, "Soft N-modular redundancy," *IEEE Trans. Comput.*, vol. 61, no. 3, pp. 323–336, Mar. 2012.

[3] P. Reviriego, S. Pontarelli, C. J. Bleakley, and J. A. Maestro, "Area efficient concurrent error detection and correction for parallel filters," *IET Electron. Lett.*, vol. 48, no. 20, pp. 1258–1260, Sep. 2012.

[4] P. Reviriego, C. J. Bleakley, and J. A. Maestro, "A novel concurrent error detection technique for the fast Fourier transform," in *Proc. ISSC*, Maynooth, Ireland, Jun. 2012, pp. 1– 5.

[5] S. Sesia, I. Toufik, and M. Baker, *LTE—The UMTS Long Term Evolution: From Theory to Practice*, 2nd ed. New York, NY, USA: Wiley, Jul. 2011.

[6] N. Kanekawa, E. H. Ibe, T. Suga, and Y. Uematsu, *Dependability inElectronic Systems: Mitigation of Hardware Failures, Soft Errors, andElectro-Magnetic Disturbances*. New York, NY, USA: Springer-Verlag, 2010.

[7] A. Sibille, C. Oestges, and A. Zanella, *MIMO: From Theory to Implementation*. San Francisco, CA, USA: Academic, 2010.

[8] M. Ergen, *Mobile Broadband—Including WiMAX and LTE*. New York, NY, USA: Springer-Verlag, 2009.

[9] S. Pontarelli, G. C. Cardarilli, M. Re, and A. Salsano, "Totally fault tolerant RNS based FIR filters," in *Proc. 14th IEEE Int. On-Line Test Symp. (IOLTS)*, Jul. 2008, pp. 192–194.

[10] B. Shim and N. R. Shanbhag, "Energy-efficient soft error-tolerant digital signal processing," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 14, no. 4, pp. 336–348, Apr. 2006.

[11] R. Baumann, "Soft errors in advanced computer systems," *IEEE Des. Test Comput.*, vol. 22, no. 3, pp. 258–266, May/Jun. 2005.

[12] M. Nicolaidis, "Design for soft error mitigation," *IEEE Trans. Device Mater. Rel.*, vol. 5, no. 3, pp. 405–418, Sep. 2005.

[13] G. L. Stüber, J. R. Barry, S. W. McLaughlin, Y. Li, M. A. Ingram, and T. G. Pratt, "Broadband MIMO-OFDM wireless communications," *Proc. IEEE*, vol. 92, no. 2, pp. 271–294, Feb. 2004.

[14] T. Hitana and A. K. Deb, "Bridging concurrent and non-concurrent error detection in FIR filters," in *Proc. Norchip Conf.*, Nov. 2004, pp. 75–78.

[15] J. Y. Jou and J. A. Abraham, "Fault-tolerant FFT networks," *IEEE Trans. Comput.*, vol. 37, no. 5, pp. 548–561, May 1988.