IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 11, November 2017

Design of Area-Efficient Fault Tolerant Parallel FFTS using Multiple Error Correction

Janani M[1], Jaishankar B[2]

Final Year M.E (VLSI), Department of ECE, KPR Institute of Engineering & Technology, Coimbatore[1]

Associate Professor, Department of ECE, KPR Institute of Engineering & Technology, Coimbatore[2]

Abstract: As signal-processing circuits become more complex, it is common to find several filters or FFTs operating in parallel. Soft errors pose a reliability threat to modern electronic circuits. For some applications, an interesting option is to use algorithmic-based fault tolerance (ABFT) techniques that try to exploit the algorithmic properties to detect and correct errors. One example is fast Fourier transforms (FFTs) that are a key building block in many systems. Several protection schemes have been proposed to detect and correct errors in FFTs. Among those, probably the use of the Parsevals or sum of squares check is the most widely known. Recently, a technique that exploits this fact to implement fault tolerance on parallel filters has been proposed. In this brief, this technique is first applied to protect FFTs. Then, two improved protection schemes that combine the use of error correction codes and Parseval checks are proposed and evaluated. The results show that the proposed schemes can further reduce the implementation cost of protection.

Keywords: Algorithmic-based fault tolerance (ABFT), Error correction codes (ECCs),Fast Fourier transforms (FFTs),Soft errors,Parseval theorem.

I.INTRODUCTION

In recent years electronic circuits are increasingly present in automotive, medical and space applications. Digital filters are widely used in signal processing and communication system. In some cases, the reliability of those systems are critical and fault tolerant filter implementations are needed. Various types of applications, the circuits have to provide some degree of fault tolerance. Fault tolerance is the realization that always have faults (or the potential for faults) in our system. Need to design a system in such a way that it will be tolerant of those faults. That is, the system should compensate the faults and continue to function. This can be achieving by redundancy. This need is further increased by the intrinsic reliability challenges of advanced CMOS technologies. Soft errors it can change the logical value of a circuit node creating a temporary error that can affect the system operation. While using filters in parallel soft errors are vulnerable to the circuits. FFT,it plays an important role in digital signal processing. An interesting option is to use algorithmic based fault tolerance (ABFT) technique[3] that try to exploit the algorithmic properties to detect and correct errors. When the circuit to be protected has algorithmic or structural properties a better option can be to exploit those properties to implement fault tolerance.

II.TECHNIQUES USED

In the protection of parallel FFTs, it is assumed that there can only be a single error on the system at any given point in time. This is a common assumption when considering the protection against radiation-induced soft errors [4]. There are three main contributions in this brief.1) The evaluation of the ECC technique [1] for the protection of parallel FFTs showing its effectiveness in terms of overhead and protection effectiveness.2) Based on the use of Parseval or sum of squares (SOSs) checks [3] combined with a parity FFT. 3) The technique on which the ECC is used on the SOS checks instead of on the FFTs.

1. Parallel FFT Protection using ECCs

A General scheme based on the use of error correction codes (ECCs) has been used. An ECC are Forward Error Correction (FEC) is a process of adding redundant data or parity data, to a message, such that it can be recovered by a receiver even when a number of errors (upto the capability of the code being used).This technique can be used for operations, in which the output of the sum of several inputs is the sum of the individual outputs [1]. This is true for any linear operation as, for example, the discrete Fourier transform(DFT).

In a parallel FFT protection using ECCs it has two types of Modules namely original modules and redundant modules. Redundant modules used for detect and correct the errors. This scheme is shown in Fig.1.In this example simple single error correction Hamming code [2] is used. The error it can determined by using observed differences on each of the check the input.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 11, November 2017

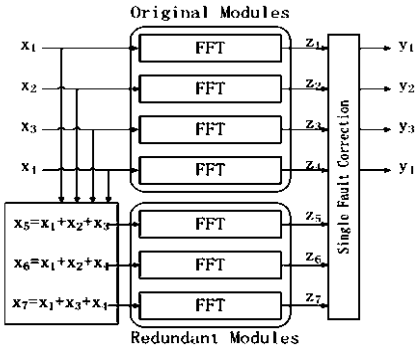


Fig 1: Parallel FFT protection using ECCs.

For the first redundant module is

$$x5 = x1 + x2 + x3 \quad (1)$$

and the DFT is a linear operation, its output z5 can be used to check that,

$$z5 = z1 + z2 + z3 \quad (2)$$

this will be denoted as c_1 check. Compare to the TMR based technique the overhead of the technique is lower [1].

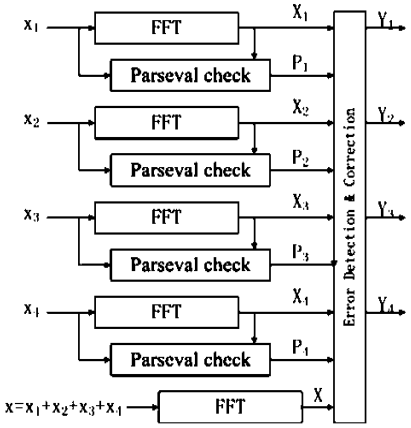
2. Parity-SOS fault tolerant parallel FFTs

Many techniques have been proposed to protect the FFT, One of them is the Sum of Squares (SOSs) check[3] that can be used to detect errors. The SOS check is based on the Parseval theorem that states that the SOSs of the inputs to the FFT are equal to the SOSs of the outputs of the FFT except for a scaling factor. For parallel FFTs, the SOS check can be combined with the ECC approach to reduce the protection overhead. The SOS check can only detect errors,

Fig 2: Parallel FFT protection using parity-SOS.

the ECC part implement the correction. This can be done using the equivalent of a simple parity bit for all the FFTs .A redundant (the parity) FFT is added that has the sum of the inputs to the original FFTs as input.An SOS check is also added to each original FFT. In case an error is detected (using P1, P2, P3, P4), the correction can be done by recomputing the FFT in error using the output of the parity FFT (X) and the rest of the FFT outputs.

$$X1c = X - X2 - X3 - X4 \quad (3)$$



3. Parity-SOS-ECC fault tolerant parallel FFTs

The parity-SOS-ECC based fault tolerant technique is the combination of the previous two techniques. Combine the SOS check and the ECC approach is instead of using an

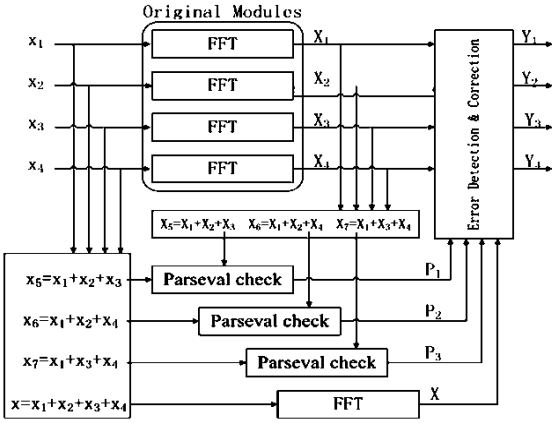
Fig 3.Parity-SOS-ECC fault tolerant parallel FFTs.

SOS check per FFT, use an ECC for the SOS checks. In this method, Parseval check is used to detect and correct the errors. The main advantage of this method is number of Parseval check is less compared with previous.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 11, November 2017



III.PROPOSED METHOD

In this proposed method multiple partial summation technique were useful for the multiple error detection and correction. In this previous existing methods it can able to correct single FFT at the time. For example if error occurs in A1 it will able to correct the error. If error occurs in A1 and A2 we can't able to correct the errors by the existing technique. The proposed work focuses on new technique for reducing the hardware overhead and increasing the error correction capability. The new technique which focuses on the existing systems limitations. The technique analyzed in the previous work has certain limitation due to the complexity of handling larger number of FFTs and Sum of Squares block. Instead of using Sum of Squares, Partial summation is used for calculating its parity at the input and the output side of the FFT.

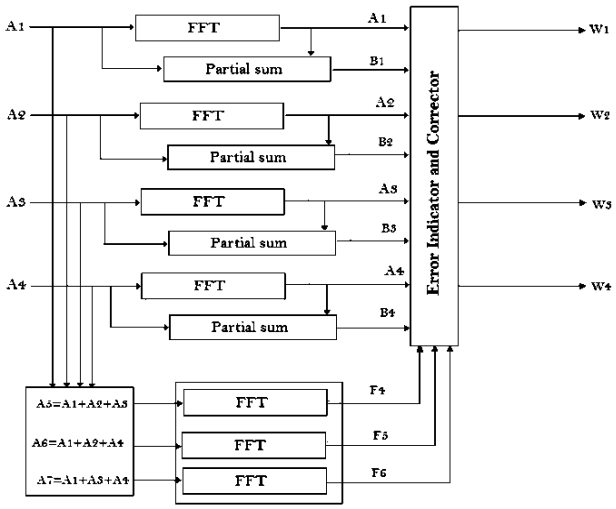


Fig 4.Parallel FFT protection using partial summation.

It sums all possible node values of 4-point FFT along with the twiddle factors. Multiplication operation which leads to the complexity in Sum of Squares is thus eliminated using only the adder blocks. Technique 1 also uses 3 parallel redundant FFT in the case of 4-parallel FFT design for correcting multiple errors in fault tolerant for soft error. Partial Summation block for less error prone applications. For example when the error occurs in A1 and A2 then it can be detected by the partial summation used individually for FFT blocks. The first check equation is selected in such a way that the both error block signals are not present.

$$A5=A1+A2+A3 \quad (4)$$
$$A6=A1+A2+A4 \quad (5)$$
$$A7=A1+A3+A4 \quad (6)$$

The error in first FFT is corrected by using equation (6) as

$$B1=B7-B3-B4 \quad (7)$$

Once the error is detected using in the first FFT is corrected then equation (4) can be used for correcting error in the second one which is given as,

$$B2=B5-B1-B3 \quad (8)$$

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 11, November 2017

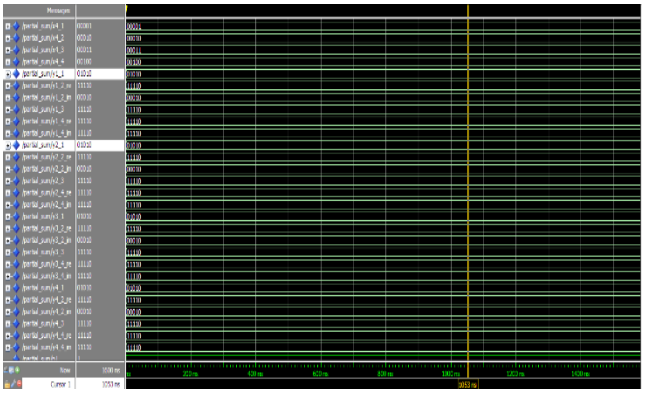


Fig 6. partial summation method -error part

The Partial Summation reduces the number of additional squaring blocks and also reduces the shelter overhead. In our implementation those blocks are protected with adders used to compute the inputs and the outputs to the redundant FFTs in the existing technique with a small impact on circuit complexity as they are much simpler than the FFT computation.

IV.RESULTS

The existing techniques and proposed method results obtained by using tools like ModelSim and Xilinx. In the proposed method the errors present in the original module FFTs it can be detected by using partial summation. The detected error in can be corrected by using ECC approach as a redundant module. In Fig 5 it shows the error part, Fig 6 it shows the corrupted part detection after that final corrected output using partial summation shown in Fig 7.

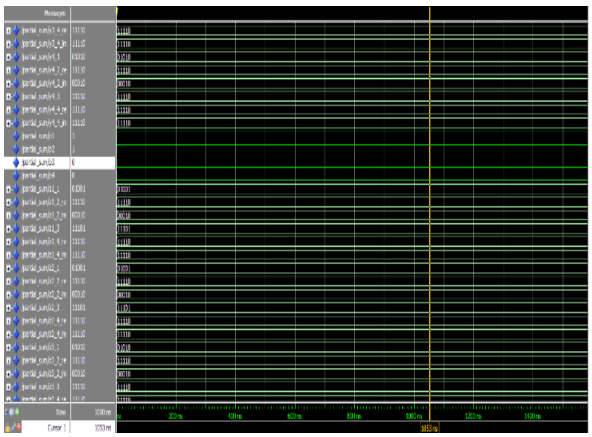


Fig 7 .corrupted part detection

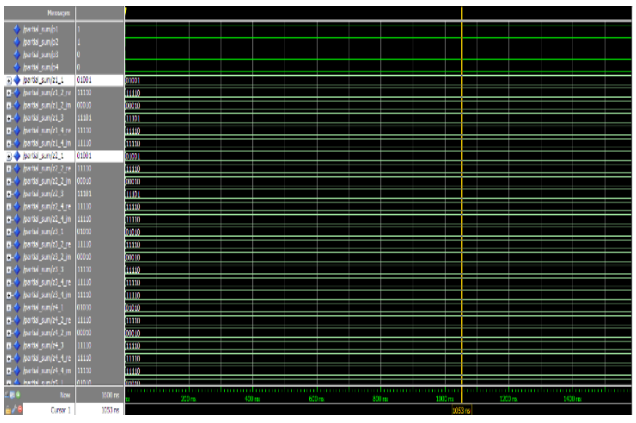


Fig 8 .Final error free output waveform.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 11, November 2017

Table 1:The comparison table made based on area, delay and power.

TECHNIQUES	AREA (Gates used)	Delay (ns)	Power (mW)
ECC	16,735	3.346	263
Parity-SOS	10,561	3.969	253
Parity-SOS-ECC	9,584	3.249	252
Partial Summation	6,948	5.051	246

V. CONCLUSION

Final observation from existing techniques that the ECC scheme can detect all errors on the other hand the SOS check detects most errors but does not guarantee the correction of all errors. The combination of ECC and Parseval check technique it can able to achieve the lowest overhead among those three existing techniques. Additionally it has advantage of the combination of (ECC & Parseval check) this method efficient area usage. Those three existing techniques its useful for single FFT error correction only. For handling multiple FFTs at the time partial summation method were useful.it also achieves the efficient area usage. In future work will be implemented in signal processing application. The partial summation it is useful for multiple FFTs error handling it also has some limitation. Overcome the limitations by modifying the structure of it. That modified structure it will applied in the OFDM applications.

REFERENCES

[1] Gao . Z et al, (Feb 2015) "Fault tolerant parallel filters based on error correction codes." IEEE Trans Very Large Scale Integer.(VLSI) Syst, Vol 23, No. 2, pp. 384-387.
[2] Hamming R W(April 1950),*Error detecting and error correcting codes*.Bell Syst.Tech J,Vol.29,No.2,pp.147-160.
[3] Reddy A.L.N and Banerjee.P,(Oct 1990),*Algorithm-based fault Detection for signal processing applications*, IEEE Trans. Comput,Vol 39,No.10,pp.z 1304-1308.
[4] Nicolaidis .M .V ,(Sep 2005),"*Design for soft error mitigation*," IEEE Trans.Device Mater.Rel,Vol.5,No.3,pp.405-418.
[5] Jou J .Y,Abraham J.A,(May 1988),*Fault-tolerant FFT networks* IEEE Trans.Comput,Vol.37,No.5,pp.548-561.
[6] Wang S.J,Jha N.K,(July 1994),*Alogorithm-based fault tolerance for FFT networks*.IEEE Trans.Comput,Vol.43,No.7,pp.849-854.