Asian Journal of Applied Science and Technology (AJAST)
Volume 1, Issue 2, Pages 1-5, March 2017

Design and Implementation of Fault Tolerant Parallel FFTs using Error Correction Codes

G.Divya[1] and P.Nagarajan[2]

[1]*PG Student, Department of ECE, Vivekanandha College of Engineering for Women, India. Email: divya30ece@gmail.com*
[2]*Assistant Professor, Department of ECE, Vivekanandha College of Engineering for Women, India. Email: greennagarajan@gmail.com*

Article Received: 17 February 2017 *Article Accepted: 27 February 2017* *Article Published: 01 March 2017*

ABSTRACT

In recent electronic circuits pose a reliability threat, a lot of applications are necessary to use protection against soft errors, there is no exceptions in the field of communications and signal processing systems. For several applications is to use Algorithmic Based Fault Tolerant techniques (ABFT) that try to expand the algorithmic properties to detect and correct the errors. The ABFT technique is well suitable for communications and signal processing applications. One of the examples is Fast Fourier Transforms that are a input building block in various systems. Some protection schemes are proposed. Among this, probably the use of the Parseval or sum of squares check is most broadly known. In modern communication systems are impressively use several blocks that blocks are operated in parallel. This type of technique is well suitable for FFTs protection. Consequently two enhanced protection schemes that merge the use of error correction codes and Parseval checks are proposed. The proposed technique to reduce the implementation cost of protection.

Keywords: Error correction codes (ECCs), Fast fourier transforms (FFTs) and Soft errors.

1. INTRODUCTION

The difficulty of communications and signal processing circuits increases every year. This is made possible by the CMOS technology scaling that enables the addition of more and more transistors on a single device. This increased complexity makes the circuits more susceptible to errors. At the same time, the scaling means that transistors operate with lower voltages and are more vulnerable to errors caused by noise and manufacturing variations. The meaning of radiation-induced soft errors also increases as technology scales. Soft errors can change the logical value of a circuit node creating a short term error that can affect the system operation. To ensure that soft errors do not affect the operation of a given circuit, a wide variety of techniques can be used. These contain the use of special manufacturing processes for the integrated circuits like, for example, the silicon on insulator. Another option is to design basic circuit blocks or complete design libraries to minimize the probability of soft errors. Finally, it is also probable to add redundancy at the system level to detect and correct errors. One typical example is the use of Triple Modular Redundancy (TMR) that triples a block and votes between the three outputs to detect and correct errors. For example TMR, the overhead is >200%. This is because the insecure module is virtual three times (which requires a 200% overhead versus the unprotected module), and additionally, voters are required to correct the errors making the overhead >200%. Another approach is to try to utilize the algorithmic properties of the circuit to detect/correct errors. This is typically referred to as Algorithm-Based Fault Tolerance (ABFT), this policy can decrease the overhead necessary to protect a circuit. Signal processing and communications circuits are well suited for ABFT as they contain normal structures and many algorithmic properties. Over the years, many ABFT techniques have been planned to protect the basic blocks that are commonly used in those circuits. Some works have considered the protection of digital filters. For example, the use of duplication using intense precision copies of the filter has been proposed as an option to TMR but with a lesser rate. In this brief, the security of parallel FFTs is considered. In particular, it is assumed that there can only be a single error on the method at any given point in time. This is a general statement while considering the protection against radiation-induced soft errors. There are three main contributions in this brief.

1) The estimation of the ECC method for the protection of equivalent FFTs showing its efficiency in terms of overhead and protection effectiveness.

2) The request of a new method based on the use of Parseval or sum of squares (SOSs) checks join with a parity FFT.

3) The proposal of a new method on which the ECC is used on the SOS checks as an alternative of on the FFTs.

2. EXISTING METHOD

The protection of filters has also been generally studied, occurrence of parallel filters to creates an opportunity to implement ABFT techniques for the whole collection of parallel modules as an alternative for each one independently. This has been studied for digital filters originally in where two filters were considered. In recent times, a broad scheme based on the use of error correction codes (ECCs) has been proposed. The design of filter implementation is complex. This will be denoted as $c1$ check. The same analysis applies to the other two redundant modules that will provide checks $c2$ and $c3$. Based on the differences observed on each of the checks, the module on which the error has occurred can be determined. The dissimilar patterns and the corresponding errors are summarized in Table 1. Once the module in error is known, the error can be corrected by reconstructing its output

using the residual modules. Related correction equations can be used to correct errors on the other modules. More advanced ECCs can be used to correct errors on many modules if that is required in a given application. The overhead of this technique, as discussed in, is lower than TMR as the number of redundant FFTs is related to the algorithm.

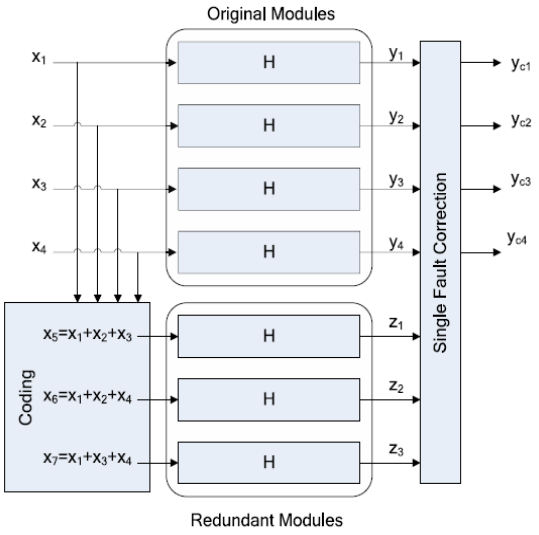

Fig.1. Parallel Filter Protection (Hamming Code)

To protect four FFTs, three redundant FFTs are required, but to protect eleven, the number of redundant FFTs in only four. This shows how the overhead reduces with the number of FFTs. In Section I, it has been mentioned that over the years, many techniques have been proposed to protect the FFT. One of them is the Sum of Squares (SOSs) check 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. This relationship can be used to detect errors with low overhead as one multiplication is needed for each input or output sample (two multiplications and adders for SOS per sample). For parallel FFTs, the SOS check can be combined with the ECC approach to reduce the protection overhead. Since the SOS check can only detect errors, the ECC part should be able to implement the correction. This can be done using the equivalent of a simple parity bit for all the FFTs. In addition, the SOS check is used on each FFT to detect errors.

When an error is detected, the output of the parity filter can be used to correct the error. This is better explained with an example. The first proposed scheme is illustrated for the case of four parallel filters. A redundant (the parity) filter is added that has the sum of the inputs to the original filter as input.

The two proposed techniques offer new alternatives to protect parallel filters that can be more capable than caring each of the filters independently. The proposed schemes have been evaluated using FPGA implementations to evaluate the protection overhead. The results explain that by combining the use of ECCs and Parseval checks, the safety overhead can be reduced compared with the use of just ECCs as proposed.

3. PROPOSED PROTECTION SCHEMES

The initial point for our work is the protection scheme based on the use of ECCs that was presented in for digital filters. This scheme is shown in Fig. 2. In this example, a simple single error correction Hamming code is used. The new scheme consists of four FFT modules and three redundant modules is added 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,

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

Given that the DFT is a linear operation, its output $z5$ can be used to verify that,

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

This shows how the overhead reduced with the number of FFTs. In this section, it has been mentioned that more than the years, a lot of techniques have been planned to protect the FFT. One of them is the Sum of Squares (SOSs) check that can be used to spot errors. The SOS check is based on the Parseval theorem that states that the SOSs of the inputs to the FFT are identical to the SOSs of the outputs of the FFT not including for a scaling aspect. This relationship can be used to detect errors through low overhead as one multiplication is needed for each input or output model (two multiplications and adders for SOS per sample).

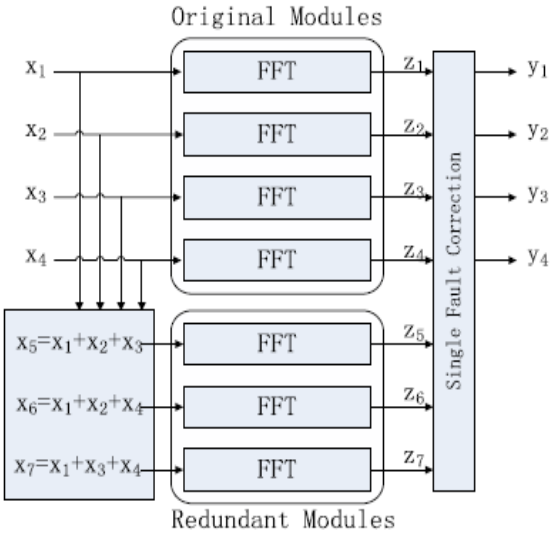

Fig.2. Parallel FFT protection using ECCs

The overhead of this technique, as discussed in, is poorer than TMR as the number of redundant FFTs is linked to the algorithm of the number of original FFTs. For example, to keep four FFTs, three redundant FFTs are needed, but to protect eleven, the number of unneeded FFTs in only four.

Parallel FFTs, the SOS check can be combined with the ECC approach to decrease the protection overhead. Since the SOS check can just detect errors, the ECC part should be able to apply the correction. This can be completed using the

equivalent of a simple parity bit for all the FFTs. In addition, the SOS check is used on every FFT to notice errors. When an error is detected, the output of the parity FFT can be used to exact the error. This is better explained with an example. In Fig. 2, the first proposed scheme is illustrated for the instance of four parallel FFTs.

This combination of a parity FFT and the SOS verify reduces the number of additional FFTs to just one and may, therefore, reduce the protection overhead. In the following, this scheme will be referred to as corresponding-SOS. Another possibility to combine the SOS check and the ECC approach is instead of using an SOS check per FFT, make use of an ECC for the SOS checks.

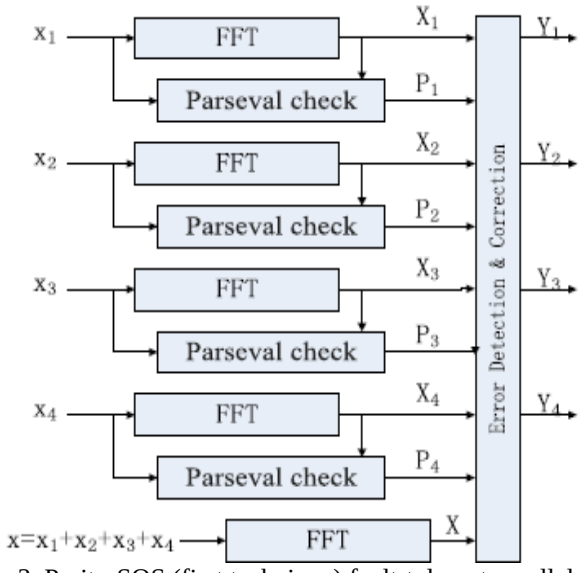

Fig.3. Parity-SOS (first technique) fault-tolerant parallel FFTs

A redundant FFT is added to facilitate has the sum of the inputs to the original FFTs as input. The SOS check is also added to each original FFT. In case an error is detected (using *P*1, *P*2, *P*3, *P4)*, the correction can be completed by re-computing the FFT in error using the output of the parity FFT (*X*) and the rest of the FFT outputs.

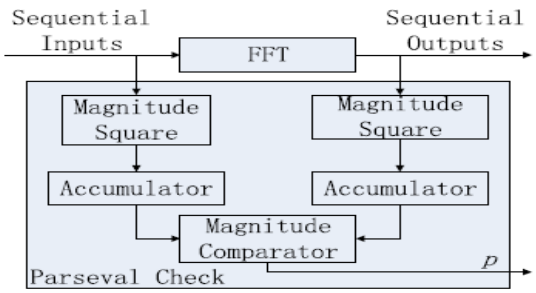

Fig.4. Implementation of the SOS check

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

The parity-SOS scheme, an additional parity FFT is used to correct the errors. This second technique is shown in Fig. 3. The main benefit over the first parity- SOS scheme is to reduce the number of SOS checks needed.

The overheads of the two proposed schemes can be initially estimated using the number of additional FFTs and SOS check blocks needed. This information is summarized for a set of *k* original FFT modules assuming *k* is a power of two. It can be observed that the two proposed schemes reduce the number of additional FFTs to just one.

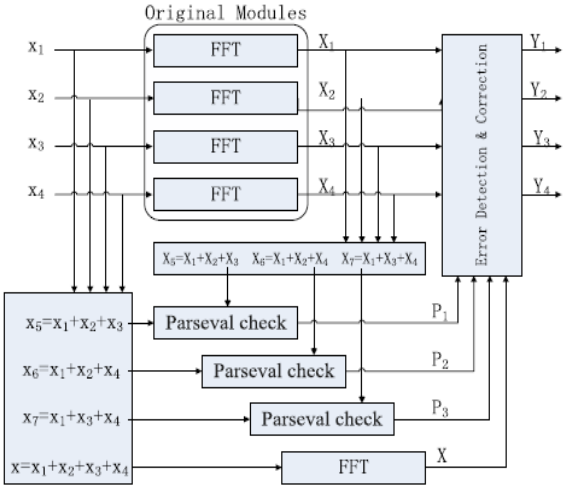

Fig.5. Parity-SOS-ECC fault-tolerant parallel FFTs

Table.1. Resources Usage for a Single FFT and SOS Check

	FFT	SOS Check
Slices	1367	494
Flip-Flop	1037	141
LUT-4	2530	974

Tables 1 explain the results when different number of parallel FFTs is protected. The point is to illustrate how the relative overheads of the different techniques vary with the number of parallel FFTs. In parentheses, the cost virtual to an unprotected implementation is also provided. The results show that all techniques have a rate factor of <2. This demonstrates that the ECC-based technique is also competitive to keep FFTs and requires a much lower cost than TMR. The parity-SOS-ECC technique has the lowest resource use in all cases and, therefore, is the best option to minimize the implementation cost.

The FFT and the various protection techniques have been implemented using Verilog. The results obtained are first table provides the resources needed to implement a single FFT and an SOS check. The results show that the FFT is more complex than the SOS check as expected. The difference will be much larger when a totally parallel FFT implementation is used. The parity-SOS-ECC scheme, the number of SOS checks also grows logarithmically and they are simpler to implement than FFTs. Therefore, it remains more competitive than the ECC scheme regardless of the number of FFTs protected.

To better illustrate this phenomenon, the number of slices required for the different schemes and number of FFTs is plotted. It can be observed that eight is the value for which parity-SOS and ECC have almost the same cost. In each simulation run, one error is inserted to mimic the behavior of

soft errors that occur in isolation. For ECC protected parallel FFTs, a tolerance level of 1 is used for the equation checks. For the parity-SOS and parity-SOS-ECC schemes, the fault coverage is determined by the tolerance level τ used in the Parseval check.

As a summary, the results show that the parity-SOS scheme outperforms only the ECC scheme for small number of parallel FFTs and the parity-SOS-ECC scheme always provides the best results.

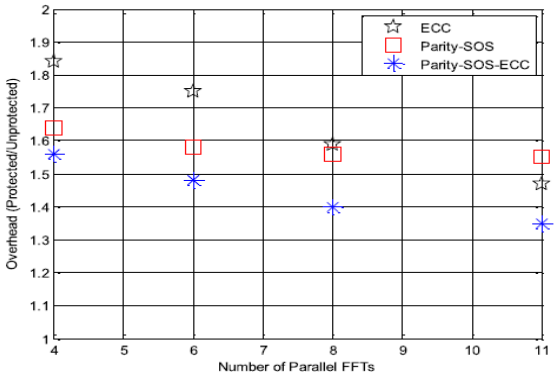

Fig.6. Overhead Comparison for the Number of Slices

Simulation Result for Parallel FFTS (Input)

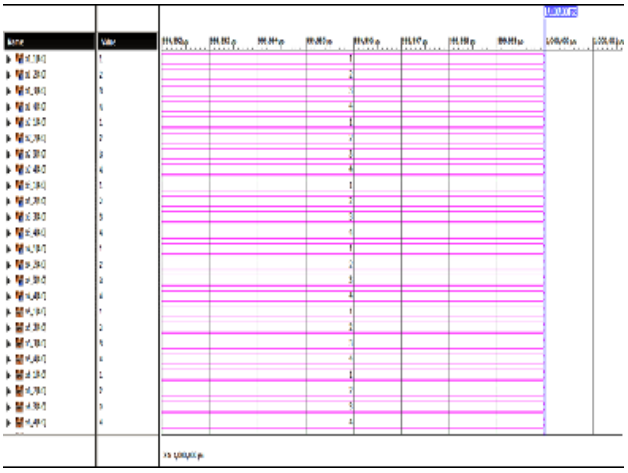

Fig.7. Simulation Result for FFTs

Simulation Result for Parallel FFTs (Output)

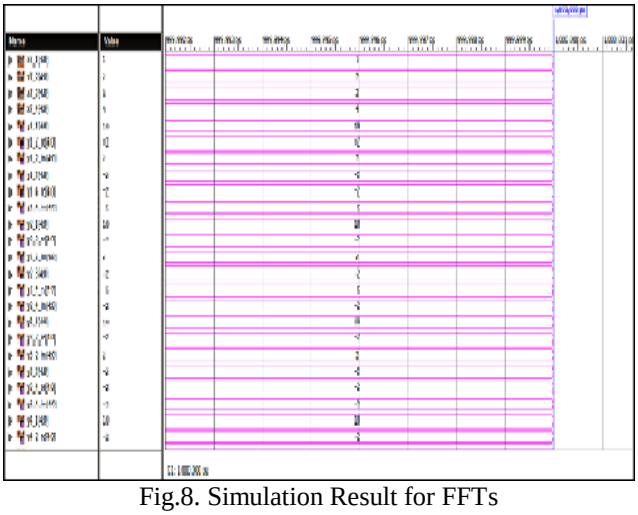

Fig.8. Simulation Result for FFTs

Simulation Result for Parallel FFTs

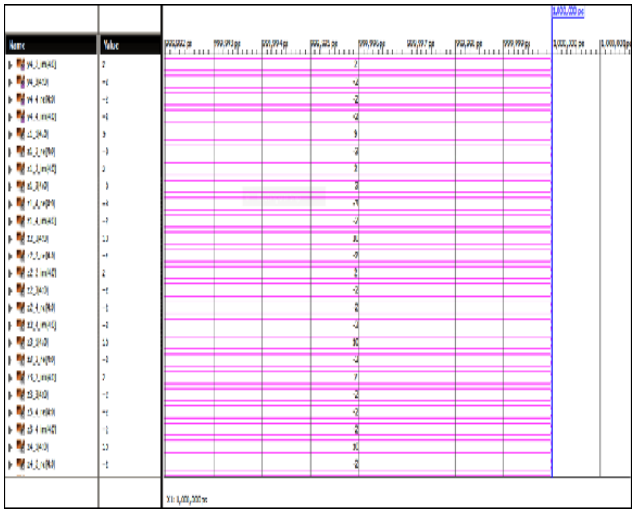

Fig.9. Simulation Result for FFTs

Simulation Result for Parallel FFTs

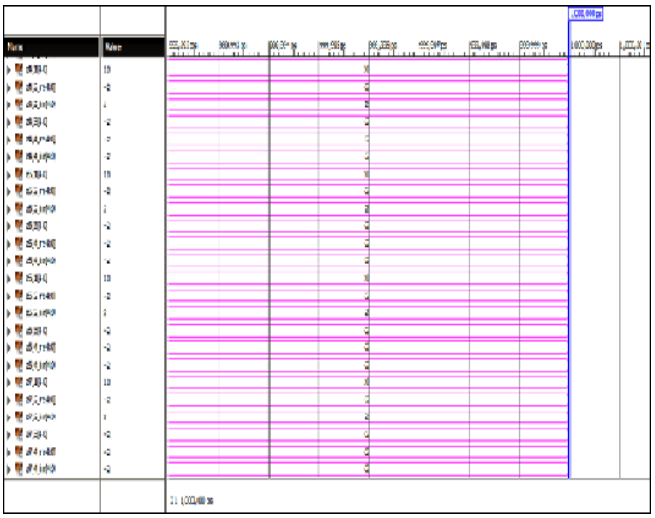

Fig.10. Simulation Result for FFTs

4. CONCLUSION

In this paper briefly studied about the protection against soft errors. Two techniques are proposed and evaluated the error correction rate. The first technique is based on merge an ECCs and with traditional SOS check. The SOS check is used to identify the errors and finds the location of errors, finally parity FFT is used for error correction.

The detection and location of errors can be completed using SOS check per FFT otherwise using a set of SOS checks that form an ECC. The proposed techniques calculate both in terms of implementation density and error detection capabilities. The second technique, which consisting of a parity FFT and set of SOS checks that form the ECC.

Result of the second technique provides the better implementation complexity. In terms of error protection, the fault injection experiments verify that the ECC method can recover all the errors that are out of the tolerance range. The fault exposure for the parity-SOS scheme and the

parity-SOS-ECC scheme is ~99.9% and then tolerance level for SOS check is 1.

REFERENCES

[1] Gao Z., et al., "Fault tolerant parallel filters based on error correction codes" (2015), *IEEE Trans. on Very Large-Scale Integr. (VLSI) Syst.*, Vol.23, No.2, pp.384–387.

[2] Baumann R., "Soft errors in advanced computer systems" (2005), *IEEE Des. Test Comput.*, Vol.22, No.3, pp.258–266.

[3] Gao Z., Yang W., Chen X., Zhao M., and Wang J., "Fault missing rate analysis of the arithmetic residue codes based fault tolerant FIR filter design" (2012), *in Proc. IEEE IOLTS*, pp.130–133.

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

[5] Jou J. Y., and Abraham J. A., "Fault-tolerant FFT networks" (1988), *IEEE Trans. Comput.*, Vol. 37, No. 5, pp.548–561.

[6] Kim E. P., and Shanbhag N. R., "Soft N-modular redundancy" (2012), IEEE *Trans. Comput.*, Vol.61, No.3, pp. 323–336.

[7] Nicolaidis M., "Design for soft error mitigation" (2005), *IEEE Trans. Device Mater. Rel.*, Vol. 5, No.3, pp. 405–418.

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

[9] Reddy.A.L.N., and Banerjee.P., "Algorithm-based fault detection for signal processing applications" (1990), *IEEE Trans. Comput.*, Vol. 39, No. 10, pp. 1304–1308.

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