MODIFIED PARALLEL FFT FOR MULTIPLE ERROR CORRECTION AND DETECTION

[1]Vinny.P.B, [2]Veena.K,[3]Linu Babu
[1]M.Tech student in VLSI Design
[2,3]Assistant Professor, ECE
[1,2,3]IES College of Engineering, Thrissur, India
[1]vinnbabuputhoor@gmail.com, [2]veenakottekatil@gmail.com,[3]linubabup@gmail.com

Abstract

The FFT is an important factor in the field of communication. Soft errors pose a reliability threat to modern electronic circuits, Communications and signal processing systems are no exceptions. . In modern communication systems, it is increasingly common to find several blocks operating in parallel. Recently, a technique that exploits this fact to implement fault tolerance on parallel filters has been proposed. though soft errors do not pose any physical values but it effect the working of the system. several methods like ABFT, TMR ,ECC, etc exist. it only deals with single error at .the proposed system here deals with multiple error by simply using ECC and SOS parity check

Key words:- soft error, FFT,ECC

1. INTRODUCTION

Reduced feature sizes and voltages in modern semiconductor process technologies have made current and future systems vulnerable to a lot of errors. Size reduction is made possible by the cmos technology scaling .it enables the integration of more and more transistors on a single device. As a result complexity in circuits such as in communication and signal process systems.

The faults will reveal themselves as temporary logic upsets. It can affect the transmitted signals and stored values which lead to incorrect or undesired outcomes in circuit and systems. Several researches have been carried out to find out the faults in electronic systems. Faults itself can be categorized in to two types: permanent and transient irreversible physical defects in the circuit are known as permanent . The next type is soft error which is also known as transient error. It appears during the operation of a circuit and won't create any physical defect . The main reasons for soft errors are cross-talk, any permanent logical error, power supply noise and neutron or alpha radiations during operational life time. Soft errors in semiconductor memories have been a known concern for many years. Soft errors can change the logical value of a circuit node faults creating a temporary error that can affect the system operation

Wide variety of techniques can be used in order to ensure that soft errors do not affect the operation of a given circuit. These include the use of special manufacturing processes for the integrated circuits like, for example, the silicon on insulator Achieving robustness and energy efficiency in nano scale CMOS process technologies is made challenging due to the presence of process, temperature, and voltage variations. The main issue with those soft errors mitigation techniques is that they require a large overhead in terms of circuit implementation

The uses of FFTs are unavoidable in communication systems. As most of the information may be analog in this world and as transmission of information in digital form is more acceptable and efficient, FFTs plays an important role in present communication scenario. For a reliable communication the information that we transmit should be reached at the destination as such. But due to many factors the information that we transmits get altered.FFT exploits the algorithmic properties . it can be considered as the basic building block of the communication systems.

But nowadays in systems FFTs are connected in parallel way to get efficient result. As signal-processing circuits become more complex, it is common to find several filters or FFTs operating in parallel. This occurs for example in filter banks or in multiple-input multiple-output (MIMO) communication systems. This way of operation is found more effective than FFTs connected in serial. it is also subject to soft errors. but this way of operation has advantage that it can model and find out the fault individually.

The paper is organized as, section 2 states the existing error correction scheme. Proposed system with modified system unit explains in section 3 and results are stated in section 4, followed by a section 5 conclusion

2. ERROR PROTECTION SCHEMES

This section describes about the existing system The main aim of the this system is to detect and correct errors in the parallel FFT though a lot of several techniques are available earlier. My proposed protection scheme purely based on the use of ECC and Sum Of Squares(SOS). By combining both FFT can be protected efficiently.

2.1 Proposed Protection Scheme

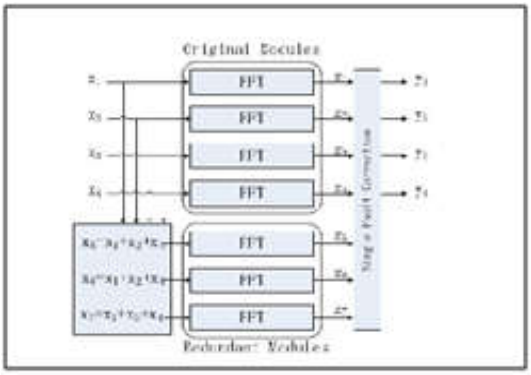

Figure 1. scheme using ECC

The first part of the protection scheme is based on the use of ECC . simple ECC takes a block of k bits and produces a block of n bits by adding n−k parity check bits. As shown in the fig:3.1 the system consists of original module which consists of 4 FFT and for the purpose of error correction and detection another 3 Redundant FFT are there. The inputs to the three redundant modules are linear combinations of the inputs and they are used to check linear combinations of the outputs.

$$x5 = x1 + x2 + x3$$
$$x6 = x1 + x2 + x3$$
$$x7 = x1 + x3 + x4$$

The parity check bits are XOR combinations of the k data bits. By properly designing those combinations it is possible to detect and correct errors

As DFT is a linear operation, its output can be used to check the property stated below

$$z5 = z1 + z2 + z3$$
$$z6 = z1 + z2 + z3$$
$$z7 = z1 + z3 + z4$$

The three equations will be denoted as c1,c2,c3. Based on the differences observed on the each of the has occurred can be determined.

Table 1. Error Correction Codes

C1 C2 C4	ERROR BIT POSITION
000	No Error
111	Z1
110	Z2
101	Z3
011	Z4
100	Z5
010	Z6
001	Z7

Once the module in error is known, the error can be corrected by reconstructing its output using the remaining modules. For example, for an error affectin $z1$

$$Z1_C[n] = z5[n] - z2[n] - z3[n]$$

Similar correction equations can be used to correct errors on the other module. Another important approach used in protection scheme 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. 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

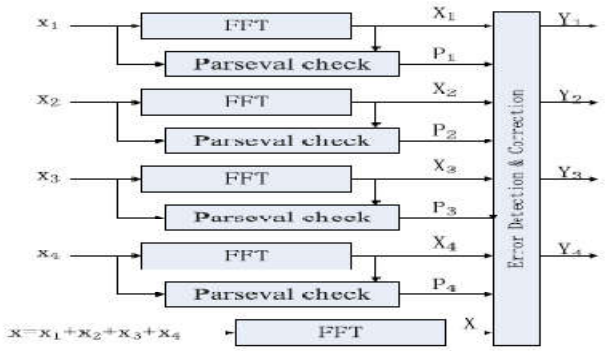

Figure 2. Parity SOS Technique

The SOS check is applied on each FFT to detect errors. When an error is detected, the output of the parity FFT is used to correct the error. A redundant (the parity) FFT is added that has the sum of the inputs to the original FFTS as input. In case an error is detected (using *P*1, *P*2, *P*3, *P*4*)*, 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. For example, if an error occurs in the first FFT, *P*1 will be set and the error can be corrected by doing

$$X1_c = X - X2 - X3 - X3$$

In this paper, protection scheme combine BOTH SOS check and the ECC approach . Instead of using an SOS check per FFT, use an ECC for the SOS checks. Then as in the parity-SOS scheme, an additional parity FFT is used to correct the errors,

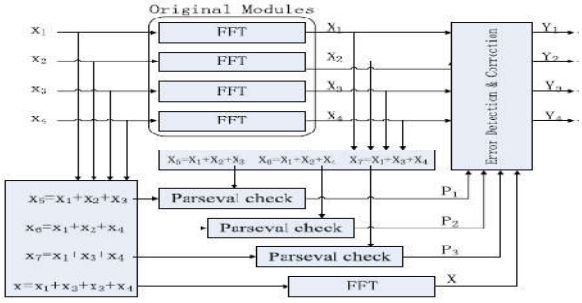

Figure 3. Existing Technique

3. PROPOSED SYSTEM

In the above proposed scheme it can only detect and correct single errors in the FFT . it won't work if there is more than one error. so in the modification part the attempt is made to detect and correct multiple errors.

Since SOS can be applied individually to each FFT unlike ECC, it can easily detect multiple errors .whenever there is error is detected ,parity bits (P1,P2,P3,P4) corresponding to that FFT is set, error correction scheme is based on ecc . it follows some criteria to correct the

Table 2. Multiple Error Correction

P4P3P2P1	BIT POSITION
0000	No error
0001	Z1
0010	Z2
0100	Z3
1000	Z4
1010	Z2,z4
0011	Z2,z1
0101	Z1,z3
1100	Z4,z3

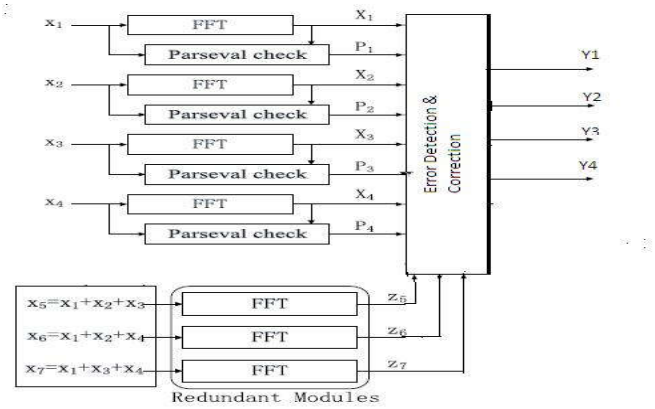

Figure 4. Proposed Technique

In all the techniques discussed, soft errors can also affect the elements added for protection. For the ECC technique, the protection of these element. In the case of the redundant or parity FFTs, an error will have no effect as it will not propagate to the data outputs and will not trigger a correction. In the case of SOS checks, an error will trigger a correction when actually there is no error on the FFT. This will cause an unnecessary correction but will also produce the correct result. Finally, errors on the detection and correction blocks can propagate errors to the outputs. In our implementations, those blocks are protected with TMR. The triplication of these blocks has a small impact on circuit complexity as they are much simpler than the FFT computations

A final observation is that the ECC scheme can detect all errors that exceed a given threshold (given by the quantization used to implement the FFTs) .On the other hand, the SOS check detects most errors but does not guarantee the detection of all errors. Therefore, to compare the three techniques for a given implementation, fault injection experiments should be done to determine the percentage of errors that are actually corrected. This means that an evaluation has to be done both in terms of overhead and error coverage.

4. RESULTS AND DISCUSSION

The following section gives the experimental studies of the paper trying to implement in ModelSim software in VHDL language, section by section. Xilinx software is used to fnd out the area power delay of the sytems so designed.

4.1 Existing System Output

Fault in FFT usually occurs due to degradation and aging .soft error and manufacturing effects also induce fault in FFT . Therefore to create a faulty output . One of the inputs given should be different. The fig given below shows the output of faulty FFT. In here
x0'th input is complimented the fault s putted at third one so its output is different. The output obtained is corrected one

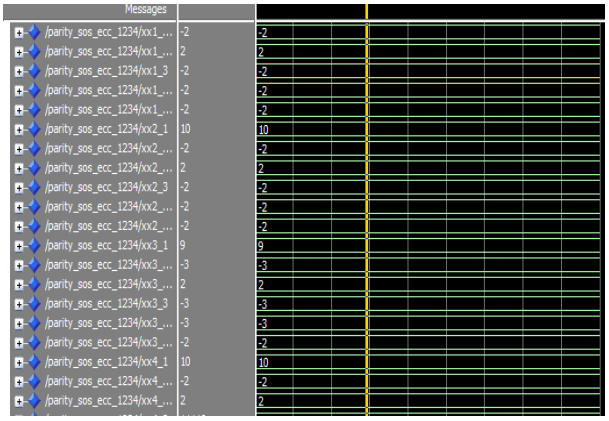

Figure 5. Fault FFT

The next step of my project is to correctly detect the faulty FFT. By utilizing the datas in the TABLEI, error location can be found out easily. Since the fault Is inputted at the third FFT . There is a specific number corresponding each FFT which is fault.

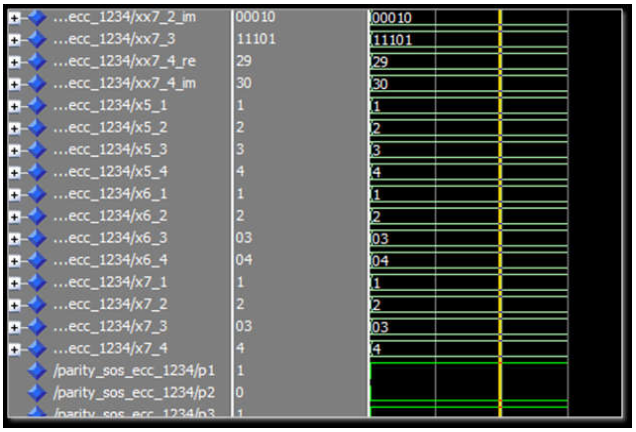

Figure 6. Fault Detection

4.2 System Output

Modification is to detect multiple errors and correcting it. The fig given below depicts the multiple error. Here error is inputted manually at the position z2 and z4 like that in the proposed system

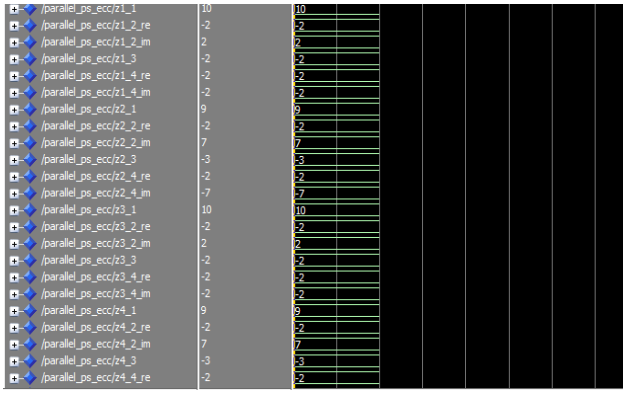

Figure 7. Multiple Error

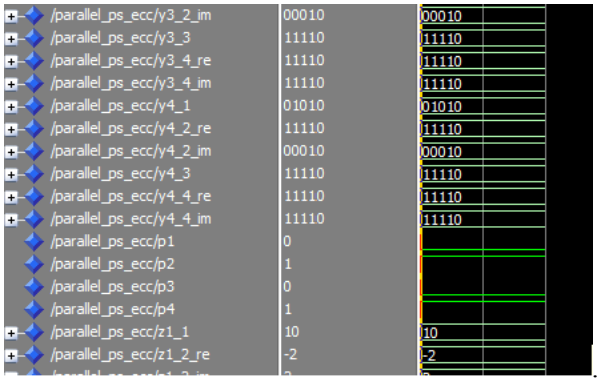

Figure 8. Fault Detection

Using the datas in the TABLE 2 ,the error position can be located easily . process following is same as that of before in the proposed system .based on the data s given on the table error bit position can be easily found out.Performance analysis of the two systems are given below.

Table 3: Performance Analysis

	Proposed System	Existing System
Power	253.mhz	219mhz
Delay	3.49 ns	3.98ns

5. CONCLUSION

The proposed techniques are based on combining an existing ECC approach with the traditional SOS check. The SOS checks are used to detect and locate the errors and a simple parity FFT is used for correction. The detection and location of the error scan be done using an SOS check per FFT or alternatively using a set of SOS checks that form an ECC. The proposed techniques have been evaluated both in terms of implementation complexity and error detection capabilities

ACKNOWLEDGMENT

I express my sincere thanks to my guide Ms Veena.K and Ms Linu Babu for her valuable guidance and useful suggestions, which helped me throughout this work.

REFERENCES

1] Zhen Gao, Pedro Reviriego," Fault Tolerant Parallel FFTs Using Error Correction Codes and Parseval Checks " IEEE Transactions on Very Large Scale Integration (VLSI) Systems Vol 24, no: 2, (**2016)**pp 769-777.

2] N. Kanekawa, E. H. Ibe, T. Suga, and Y. Uematsu,"Dependability in Electronic Systems: Mitigation of Hardware Failures, Soft Errors, and Electro-Magnetic Disturbance" New York, NY, USA **(2010**)Springer-Verlag

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

4] M. Nicolaidis, "Design for soft error mitigation," IEEE Trans. Device Mater. Rel., vol. 5, no. 3, (**1990),**pp. 405–418.

5] 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.

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

7] 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), **(2008)** pp. 192–194.

8] 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, **(2006)** pp. 336–348.

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

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

11] S.-J. Wang and N. K. Jha, "Algorithm-based fault tolerance for FFT networks," IEEE Trans. Comput., vol. 43, no. 7, **(1994)**, pp. 849–854.

12] P. P. Vaidyanathanm, Multirate Systems and Filter Banks.Englewood Cliffs, NJ, USA: **(1994**)Prentice-Hall,

13] E. P. Kim and N. R. Shanbhag, "Soft N-modular redundancy," IEEE Trans. Comput., vol. 61, no. 3, (2012),pp. 323–336.

14] J. Y. Jou and J. A. Abraham, "Fault-tolerant FFT networks," IEEE Trans. Comput., vol. 37, no. 5, (1988), pp. 548–561.

15] S.-J. Wang and N. K. Jha, "Algorithm-based fault tolerance forFFT networks," IEEE Trans. Comput., vol. 43, no. 7, (1994) pp. 849–854.