Design of Low Complexity Fault Tolerant Parallel Ffts Using Partial Summation And Parallel Correction

M. Sakthivel, M.E
Assistant Professor
Department of ECE
Velammal College of Engineering and Technology,
Madurai, Tamil Nadu, India

S. Rohini
IV year,
Department of ECE,
Velammal College of Engineering and Technology,
Madurai, Tamil Nadu, India

P. Lakshana
IV year,
Department of ECE,
Velammal College of Engineering and Technology,
Madurai, Tamil Nadu, India

P. S. Vaishnavi
IV year,
Department of ECE,
Velammal College of Engineering and Technology,
Madurai, Tamil Nadu, India

***Abstract*— Now a days 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. This makes protection against soft errors a requirement for many applications. Communication and signal processing systems are affected by this soft error. Some applications use Algorithmic-Based Fault Tolerance (ABFT) techniques to detect and correct errors. ABFT is most preferred technique for Signal processing and communication systems. One example is Fast Fourier Transforms (FFTs) is an important 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 Parseval or Sum Of Squares is the most widely known. In modern communication system, it is increasingly common to find several blocks operating in parallel. Recently, a technique that exploits this technique to identify fault tolerance on parallel filters has been proposed. Hence Parseval or Sum Of Squares technique is first applied to protect FFTs. Then, two improved protection schemes that combine the use of error correction code and Parseval checks are evaluated. So that the proposed method can further improve the protection of the communication systems.**

I. INTRODUCTION

In every year, the complexity of communication and signal processing circuits increases. However it is made possible by CMOS technology scaling which enables the integration of more and more transistors on a single device. The increase in complexity build the circuits more vulnerable to errors.

Similarly, the scaling operates the transistors with lesser voltage and are more susceptible to errors. These errors are caused by noise and manufacturing variations. The significance of radiation-induced soft error is increased enormously as technology scale[2].

The logical value of the circuit node can be changed by the soft error creating a temporary error which may affect the system operation. A wide variety of techniques can be used to ensure that soft error doesn't affect the operation of the given circuit. These technique provides the use of special manufacturing process for integrated circuits (silicon on insulator). The other way is to design basic circuit blocks or complete design libraries to reduce the probability of soft errors. Lastly it is also problem to add redundancy at the system level for error detection and correction.

II. RELATED WORKS

One classical example to detect and correct error is the use of Triple Modular Redundancy(TMR) which will triple a block and votes among three outputs. The main drawback of this technique is they require a large overhead in terms of circuit implementation. The overhead of TMR is greater than 200% as unprotected module is replicated three times. Another approach is to use algorithmic properties of the circuits in order to detect/correct error. This is referred as Algorithmic Based Fault Tolerance(ABFT)[3].

The technique of ABFT reduces the overhead required to protect the circuit. As the complexity of the signal processing circuits increases, it is possible to find out several filters or FFT operating in parallel. This may occur in filter banks or in Multi Input Multi Output (MIMO) communication systems. In general MIMO Orthogonal Frequency Division Modulation(MIMO-OFDM) includes parallel FFTs/FFTs for modulation/demodulation. More frequently, a scheme based on the Error Correction Code (ECC) has been implemented[7]. The idea behind this technique is that each filter is an equivalent of a bit in an ECC and parity check bits.

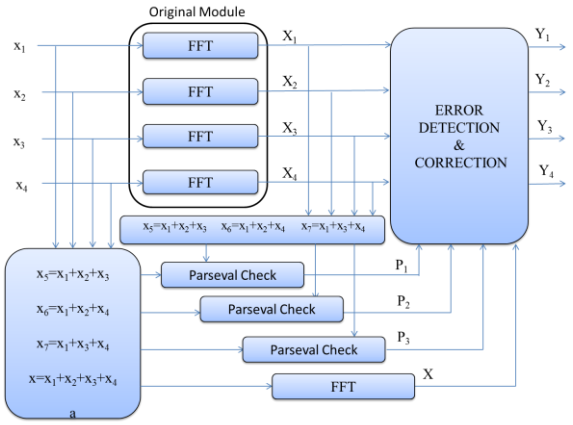

Fig.1 Parity-SOS-ECC fault tolerant parallel FFTs

In particular it is proved that, there can be only single error on the system at any given point of time. This is usual while considering the protection against radiation-induced soft error. The three main contribution are,

1. The evaluation of the ECC method[7] for the protection of parallel FFTs that shows its effectiveness in terms of overhead and protective effectiveness.

2. The existing technique based on the combination of parseval or SOS check and parity FFT.

3. The existing technique on which the ECC is used on the SOS check instead of on the FFT.

As shown in Fig.1 it has been studied that this method is used to detect and correct single bit error. Also this scheme is evaluated using FPGA implementation in order to assess protection overhead. Hence the result shows that protection overhead can be reduced compared with the use of ECCs.

To detect and correct errors fault injection experiment is to be conducted to verify the ability of the implementation.

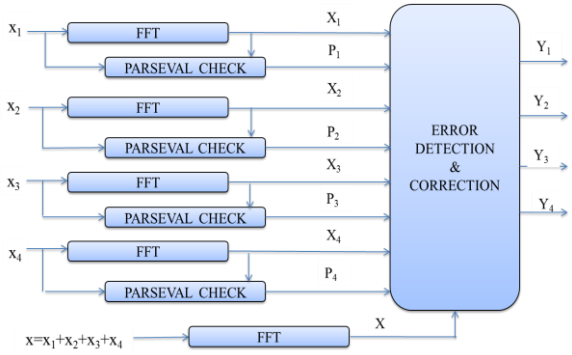

Fig.2 Parity-SOS fault tolerant parallel FFTs

III. PROPOSED SCHEMES FOR PARALLEL FFTS

The Starting point of proposed scheme is the protection based on the use of ECCs for digital filter[7].In this scheme Double error correction Hamming code is used. The original system includes four FFT modules and three Redundant modules for error detection and correction. In this method, we are using parseval check method to detect the errors in the original module. By using this method we can able to detect two error in the original module as shown in Fig.3. After that errors in the original modules are corrected by using ECC. The inputs of the three redundant modules are linear combination of the inputs and is used to check the linear combination of the outputs. For example, the input to the first redundant module is

$$X_5=X_1+X_2+X_3 \quad [1]$$

Since the DFT is a linear operation, its output Z_5 checks that

$$Z_5=Z_1+Z_2+Z_3 \quad [2]$$

This will be denoted as C_1check. The same logic is used for the other two redundant module which provides C_2 and C_3 The module on which the error occurred can be determined by the differences observed on each of the checks. The Table I shown describes the different pattern and corresponding errors. Once the module in error occurs , the error can be corrected by reconstructing its output based on the remaining modules. For example, for an error affecting z_1 it can be done by,

$$Z_{1c}[n]=Z_5[n]-Z_2[n]-Z_3[n] \quad [3]$$

Same correction equations can be used to correct the error on the another module and advanced ECCs can be used to check errors on multiple modules. The overhead of this technique [7] is lesser than TMR since the number of redundant FFTs are compared to the logarithm of the number of original FFTs. For example three redundant FFTs are needed to protect four FFTs, but protect Eleven, the number of redundant FFTs is only four. This shows that How overhead decreases with the number of FFTs.

Table I: Error Location in the Hamming Code

$C_1\ C_2\ C_3$	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

In Section I, many techniques have been proposed to protect the FFTs. One such technique is the Sum Of Squares (SOSs) check which can be used to detect errors. As shown in Fig.2 the SOS check based on the Parseval theorem states that the SOSs of the input to the FFT equals the SOSs of the output of FFT except for scaling factor. With one overhead, this relationship can be used to detect error as one multiplication is needed for each input or output sample.

For parallel FFTs, the two approaches combined for protection overhead. That is the SOS is combined with ECC approach. This will contain two separate sections. One is error detection and another one is error correction. The error detection handled by SOS, and the error correction can be implemented by ECC. Each FFT uses SOS check for error detection which can by using parity bit for all FFTs. The parity FFT can be used to correct the error when an error is detected.

The proposed scheme is illustrated in the case of four parallel FFTs. The redundant FFT added, has the sum of inputs to the original FFTs as input. A Parseval check is also added to each of the original FFT. If an error is detected the correction can be performed by reconstructing the FFT using the output of the parity FFT and also the rest of the FFT outputs. In case an error occurs the first FFT P_1 set and error is corrected by

$$X_{1c}=X-X_2-X_3-X_4 \quad [4]$$

The combined model of parity FFT and SOS check reduces the number of further FFTs to just one and it will also reduce the protection overhead. This method is also referred as parity-SOS. The technique illustrated above explains that soft error can also affect the element included for protection. In ECC technique, the protection of the elements was explained [7]. In case of redundant or parity FFTs, error has no affect since it won't propagate to the data output and won't trigger a correction. Similarly the SOS check will trigger a correction though there is no error on the FFT. This will create an unnecessary block may propagate error to outputs and such block can be protected using correction but produces correct result. Hence error on the detection and correction TMR.

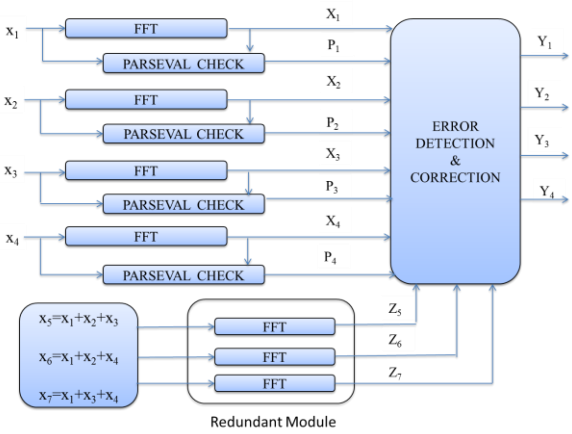

Fig.3 Block diagram for proposed system

Finally it was observed that the ECC scheme can detect all errors exceeding a given threshold. On the other case parseval check (SOS) detect many errors but doesn't guarantee the detection of all errors. Thus to compare the three techniques fault injection experiment is to be done which determines the percentage of errors actually corrected. Hence evaluation has to be done in terms of both overhead and error coverage.

IV.EVALUATION

The implementation of four point FFT core is shown in fig.4. In this number of FFT point is programmable and it is also used to calculate rotation coefficient online for each stage and stored in registers. The inputs and outputs are 12bits and 14bits wide. The bit width are extended to 14 and 16 bit for the redundant FFT to cover the larger dynamic range since inputs and outputs of the FFT are sequential, the SOS check is performed sequentially compared at the end of the block. This is shown in figure 5.

The impact of round off on the fault coverage is minimized if the output of the accumulator is 39 bit wide. Several values of the number of parallel FFTs are considered for the evaluation. This is done in order to compare different techniques as the function of number of parallel FFTs.

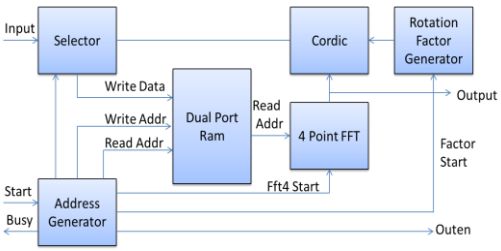

Fig.4 Architecture of FFT Implementation

The tools used in the scheme are Modelsim and Xilinx. The Modelsim is used for test the code and analysis is done by using Xilinx. The Xilinx finds out the area and power consumption. In this scheme the area and power consumption is large compared to the existing one.

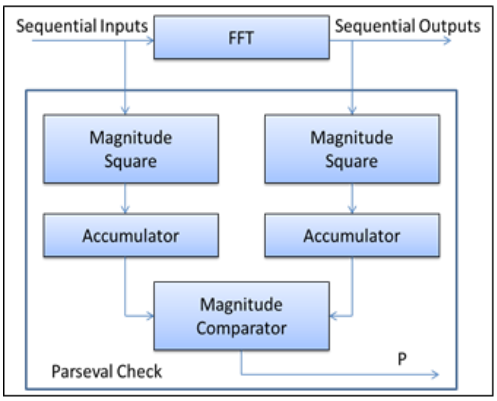

Fig.5 Implementation of SOS Check

Table II. Resource usages for Parallel FFTs

Description	Existing Method	Proposed Method
Total Equivalent Gate Count for Design	6,912	11,837
Total Estimated Power Consumption	164	257
Number of Slices Occupied	464	475
Total Number of 4-Input LUTs	887	818
Additional JTAG Gate Count for IOBs	7680	7728

From the analysis the total equivalent gate count used in the proposed method is much large when compared to the existing method. Also it was observed that the total power consumption in the proposed method is much larger than existing method. Compared to the existing one the area and power consumption is increased to 58% and 71% in the proposed system. Since the proposed method concentrates on two error detection and correction, when compared to the existing method which concentrates in single bit error detection and correction. Hence the tabulated values are larger than the existing method.

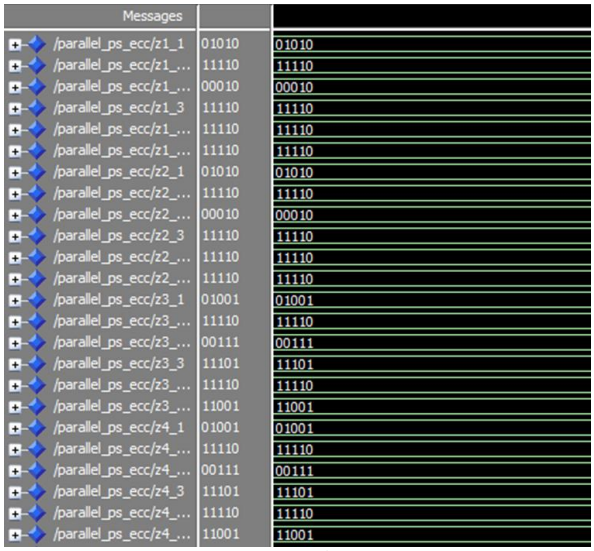

Fig.6 Output of Error Detection

Assume that the input to the FFTs (x_1, x_2, x_3, x_4) is 1,2,3,4. These are denoted in the binary format. We use XOR operation for the addition process. The parseval check is done to ensure that the output of the Parallel FFTs is equal to the given input. In case if the input is not equivalent to the output, then error is detected and it is corrected by using ECC scheme. For example if we create an error in parseval check, the output of this say P_1, P_2, P_3, P_4 If the output of the parseval check is zero then there is no error and the output is one then error occurs.

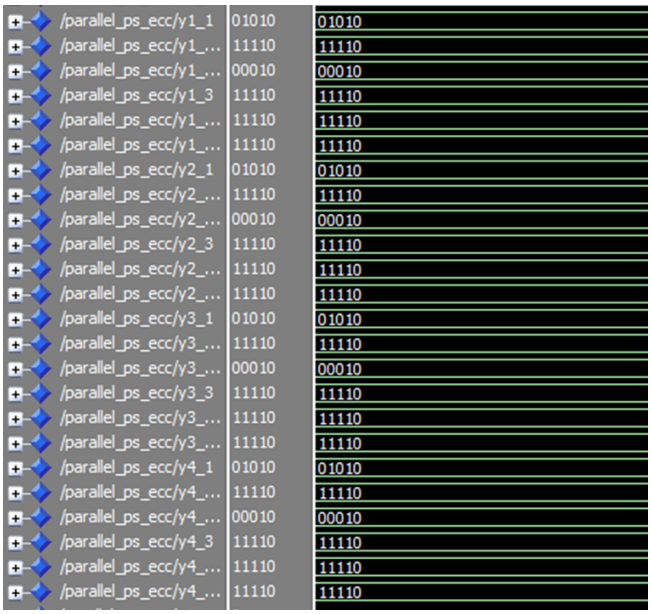

Fig.7 Output of Error Correction

From this the error is detected and corrected for any two bits. In fig.6, for example the error is detected in third and fourth bit, since the expected output is ten but in this case it is nine. This is then corrected by the Error Correction Code and would get the required output as shown in fig 7.Thus by this way the error is detected and corrected.

V.CONCLUSION

In this project, we presented a new scheme to detect and correct two bit error to protect parallel filters that are commonly found in modern signal processing circuits. This method use ECC approach to the parallel filters outputs to detect and correct errors. The scheme can be used for parallel filters that have the same response and process different input signals.

A case study has also been discussed to show the effectiveness of the scheme in terms of error correction and also of circuit overheads. The technique provides larger benefits when the number of parallel filters is large. The proposed scheme can also be implemented in IIR filters. Future work will consider the appraisement of the advantages of the proposed technique for IIR filters.

The extension of the error detection and correction scheme to parallel filters that have the same input and different impulse responses is also take as a new topic in later for future work. This scheme will be comfortable one when the number of parallel filters is small and also benefit for large number of parallel filter in this case of the cost of this scheme is larger. Another interesting topic to continue this brief is it able to fine and rectify multi bit errors efficiently.

REFERENCE

[1]. 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 transactions on very large scale integration (vlsi) systems, vol. 24, no. 2, February 2016.

[2]. R.Baumann, “Soft errors in advanced computer systems”, IEEE Des. Test Compute., vol. 22, no. 3, pp. 258–266, May/Jun. 2005.

[3]. 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.

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

[5]. Eric P. Kim, Student, IEEE, and Naresh R. Shanbhag, Fellow, IEEE “Soft NModular Redundancy”, in 2010 proc. international conf. on acoustics, speech, and signal processing (icassp), mar. 2010.

[6]. P. Reviriego, S. Pontarelli, C.J. Bleakley and J.A. Maestro “Area efficient concurrent error detection and correction for parallel filters”, article in electronics letters September 2012.

[7]. Zhen Gao, Pedro Reviriego, Wen Pan, Zhan Xu, Ming Zhao, Jing Wang, and Juan Antonio Maestro “Fault Tolerant Parallel Filters Based on Error Correction Codes”, IEEE transactions on very large scale integration (vlsi) systems, February 2015.