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 Low Complexity Fault Tolerant Parallel FFTs using Parallel-SOS-ECC and Parallel correction

L Pavithra[1], R Naveen[2]

M.E VLSI Design, Info Institute of Engineering, Coimbatore, India[1]

Associate Professor, Info Institute of Engineering, Coimbatore, India[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. This makes protection against soft errors a requirement for many applications. Communications and signal processing systems are no exceptions to this trend. For some usage, an attractive choice is to use Algorithmic-Based Fault Tolerance (ABFT) techniques that try to effort 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, perhaps the use of the Parseval 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. With the help of error correction codes and parseval checks are arranged and checked to protect FFTs. The results show that the proposed schemes can further reduce the implementation cost of protection complexity of communications and signal processing circuits increases every year. In this technique, the idea is that each filter can be the equipment of a bit in an ECC and parity check bits can be computed using addition. This technology is used for operation, in which the output of the sum of several input is the sum of the individual outputs. This is true for any linear operations as Discrete Fourier Transform (DFT).This can be simulated with the help of simulation tools like Xilinx and ModelSim.

Keywords: DFT, FFT, ECC, Parity SOS, Parseval check, Soft Errors.

I. INTRODUCTION

The Fast Fourier Transform is a highly efficient procedure for determining the DFT of a finite series and requires less number of computation than that of direct evaluation of DFT. It reduces the computation by taking advantage of the fact that the calculation of the coefficients of the DFT can be carried out iteratively. Due to this, FFT computation approach is used in digital spectral analysis

- Filter simulation
- Autocorrelation and Pattern recognition

The FFT is based on decomposition and breaking the transform into smaller transforms and combining them to get the total transform .FFT reduces the computation time required to compute a DFT and improves the performance by a factor 100 or more over direct evaluation of the DFT [1]. It makes the use of the symmetry and periodicity properties of twiddle factor W_N^K to effectively reduce the DFT computation time.

FFT algorithms are based on the fundamental principle of decaying the computation of Discrete Fourier Transform of a sequence of length N into successively smaller discrete Fourier Transforms [4]. The direct evaluation of DFT requires N^2 complex multiplication and N(N-1) complex additions .Thus for reasonably large values of N direct evaluation of the DFT requires an inordinate amount of computation .By using FFT algorithms the number of computations can be reduced. For example, for an N-point DFT, the number of complex multiplication required using FFT is $N/2\log_2$. If n =16. The number of complex multiplication required for direct evaluation of DFT is 256, whereas by using DFT only 32 multiplication are required.

II. EXISTING METHODOLOGY

A. Error correction code

In Parallel FFT protection using ECCs method, error correction codes (ECCs) has been used to detect and correct the error. The idea is that each FFT can be the equivalent of a bit in an ECC and parity check bits can be computed using addition. This approach can be used for operations, in which the output of the sum of several inputs is the sum of the individual outputs. More number of FFTs are needed to detect as well as correct the errors. So, area and power consumption of this method is too high [1].

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

B. Parity SOS

In Parity-SOS (first approach) fault-tolerant parallel FFTs method, Parseval check is used to detect the errors in individual FFT and only one redundant FFT is enough to correct the error. So that area and power consumption of this method is less compared to existing method .In Parity-SOS-ECC (second approach) fault-tolerant parallel FFTs 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 first method. This method is more effective compared with first method.

C. Block Diagram of ECC

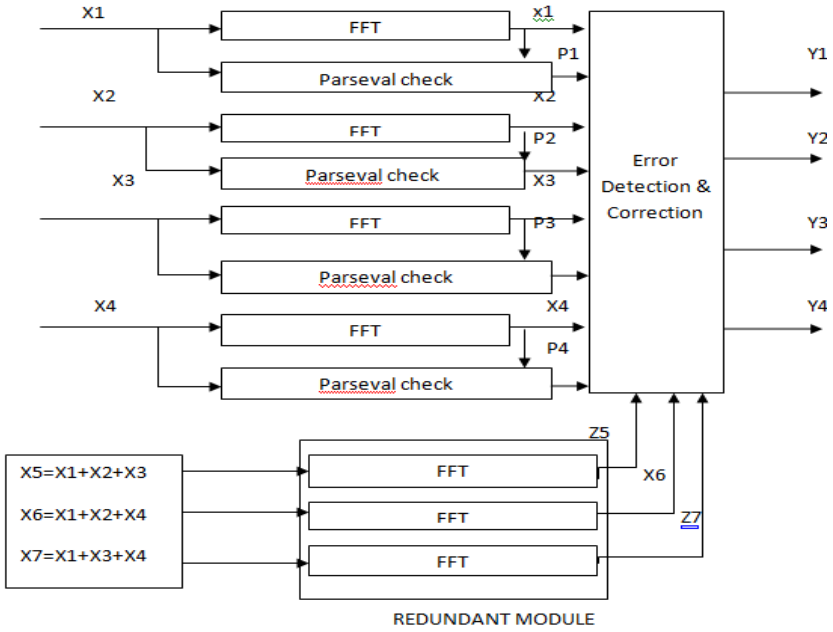

Fig.1 Block Diagram of Error Correction Code

The block diagram consists of four FFT module and three REDUNDANT module. In this the REDUNDANT module is the linear combination of inputs and they are used to check the linear combinations of the outputs. The basic work is to protect the scheme based on ECC's and it is presented with a simple error correction hamming codes. The first input of REDUNDANT module is "**X5=X1+X2+X3"**. In this the linear operation works as DFT hence the output is mentioned as **"Z5"** and it also used to check and it can be written as "**Z5=Z1+Z2+Z3"**. This will be denoted as "**C1"** check. The second input of REDUNDANT module is **"X6=X1+X2+X4"** and the third input of REDUNDANT module is **"X7=X1+X3+X4"**. The same reasoning is applied to the other two REDUNDANT modules that will provide checks as **"C1 & C2"**. Based on the difference observed on each of the checks, the module on which the error has occurred can be determined. The different pattern and the corresponding errors are summarized in Table I.

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

Once the module in error is known, the error can be reformed by reconstructing its output using the remaining modules. If an error affecting "**Z1".** This can be done by following **Z1c[*n*]=Z5[*n*]+Z2[*n*]+Z3[*n*]**. Similar correction equation can be used to correct errors on the other modules. More advanced ECC's can be used to correct errors on multiple modules if that is needed in a given application. The overhead of this technique, as discussed in, lower than TMR as the number of redundant FFT's is related to the logarithm of the number of original FFT's. To protect four

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

FFT's, three redundant FFT's are needed, but to protect eleven, the number of redundant FFTs in only four. This shows how the overhead decreases with the number of FFT's.

In above, it has been mentioned that over the years, many technique 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 the outputs of the FFT except for a scaling factor. This relationship can be used to detect errors with low overhead as one multiplication this needed for each input or output sample (two multiplication and address for SOS per sample). For parallel FFT's, 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 utensil the correction. This can be done using the equivalent of a simple parity bit for all the FFT's. In addition, the SOS check is used on each FFT to detect errors.

When an error is detected, the output of the parity FFT can be used to correct the error. This is better explained with an example.

D. Algorithm Based Fault Tolerance

In classical is the use of triple modular redundancy (TMR) that triples a block and votes among the three outputs to detect and corrects errors. The main issue with those soft errors mitigation techniques is that they require a large overhead in terms of circuits implementation for TMR the overhead is >200% . This is because the unprotected module is replicated three times (which requires a 200% overhead versus the unprotected module).and additionally. Voters are needed to correct the errors making the overhead > 200% [1]. This overhead is excessive for many applications. Another approach is to try to use the algorithmic properties of the circuits to detect/correct errors. This is commonly implied to as Algorithm-Based Fault Tolerance (ABFT).

This strategy can reduce the overhead required to protect a circuit. Signal processing and communication circuits are well suited for (ABFT) as they have regular structures and many algorithmic properties. Over the years, many (ABFT) technique have been proposed to protect the basic blocks that are commonly used in those circuits.

Several works have examined the protection of digital filters. The use of the replication using reduced precision copies of the filter has been proposed as an alternative to TMR but with a lower cost [7]. The knowledge of the distribution of the filter output has also been recently taken advantage of to detect and correct error with lower overhead the protection of Fast Fourier Transforms (FFTs) has also been widely studied. In this brief, the protection of parallel FFT's is studied. In particular, 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 error. There are three main contributes in this brief, they are:-

1) The evaluation of the ECC technique for the protection of parallel FFT's showing its effectiveness in terms of overhead and protection effectiveness.
2) The proposal of a new technique based on the use of parseval or sum of square (SOSs) checks combined with a parity FFT.
3) The proposal of a new technique on which the ECC is used on the SOS check instead of on the FFT's.

E. BLOCK DIAGRAM OF PARITY SOS

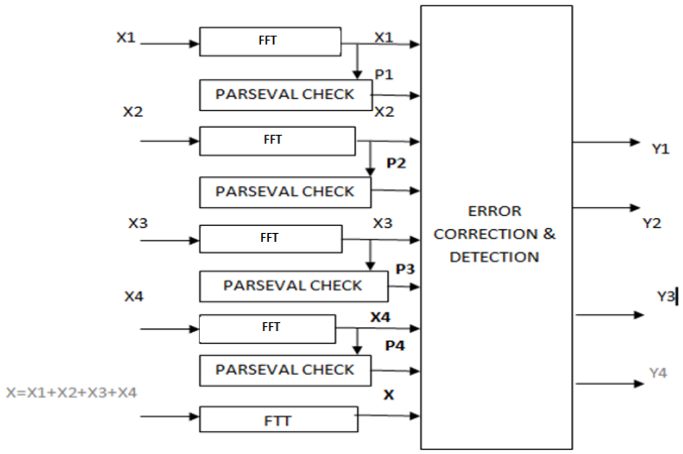

Fig 2 Block Diagram Of Parity SOS

In this block diagram consists of five FF T modules and four parseval check blocks. The inputs are "**X1,X2,X3,X4"** and it is also seperatly proposed to each and every parseval check blocks as **"P1,P2,P3,P4"**. The

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

combination of sum of the inputs are as **"x=x1+x2+x3+x4"** and it is given as the input for the last FFT block to detect the errors and this block is also otherwise called as redundant block. All the inputs are combined with their parseval block check bits and its fed into the error correction and detection block and all the four inputs produces individual four outputs from this error correction and detection block output terminals are mentioned as **"Y1, Y2, Y3 , Y4".** This error correction and detection blocks is used to correct whether the inputs have any error and it detect the input and produce an corrected error free output to the which is given .The first proposed scheme is illustrated for a case of four parallel FFT's.

A redundant (the parity) FFT is added that has the sum of the inputs to the original FFT's 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. If an error occurs in the first FFT, P1 will be set and the error can be corrected by doing **"X1c=X-X2-X3-X4"** . The grouping of a parity FFT and the SOS check reduces the number of additional FFT's therefore, reduce the protecting overhead. In the following, this scheme will be referred to as parity- SOS (or first proposed technique).Another possibility to combine the SOS check and the ECC approach is 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. This second technique is shown above in the block diagram.

F. Multiple Input and Multiple Output (MIMO)

MIMO is a communication system which participates in a particular MIMO orthogonal frequency division modulation (MIMO OFDM) system use parallel iFFT's/FFT's for modulation / demodulation MIMO-OFDM is implemented on long-term evolution mobile systems and also on WiMax. The presence of parallel filters or FFT's creates an opportunity to implement ABFT technique [1] for the entire group of parallel modules instead of for each one independently. This has been studied for digital filters initially in where two filters were considered. More recently a general scheme based on the use of error correction codes (ECC) has been proposed. In this approach the idea is that each filter can be the equivalent of a bit in an ECC and parity check bits can be computed using addition.

This approach can be used for operations, in which the output of the sum of several input is the sum of the individual outputs. This is true for any linear operation as, the Discrete Fourier Transform (DFT).The two proposed schemes and the ECC scheme have been implemented on an FPGA and evaluated both in terms of overhead and error coverage. A four-point decimation-in-frequency FFT core is used to compute the FFT iteratively. This core has been developed to implement MIMO-OFDM for wireless systems [2].

The number of FFT points is programmable and the rotation coefficients are calculated on-line for each stage and stored in registers. For the evaluation, a 1024 points FFT is configured with five stages calculation (log41024 = 5), so in total 5∗1024 = 5120 cycles are needed to calculate the FFT for 1024 input samples. The inputs are 12-bit wide and the outputs are 14-bit wide. For the redundant FFT, the bit widths are extended to 14 and 16 bit, respectively, to cover the larger dynamic range (as the inputs are the sum of several signals). Since both the inputs and outputs to the FFT are sequential, the SOS check is also done sequentially using accumulators that are related at the end of the block.

III. PROPOSED METHODOLOGY

A. BLOCK DIAGRAM OF PARITY SOS ECC

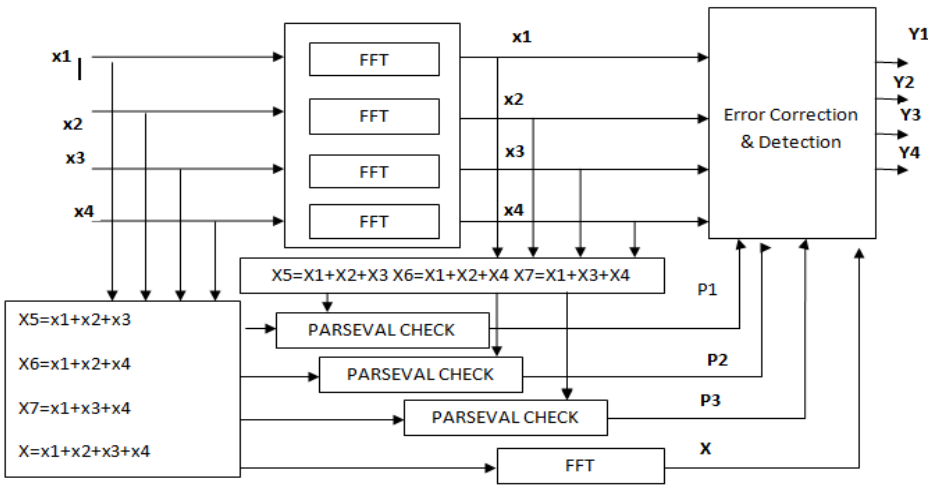

Fig.3 Block Diagram Of Parity SOS ECC

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

The main advantage over the first parity-SOS design is to decrease the number of SOS checks needed. The error location process is the same as for the ECC scheme in Fig.1 and correction is as in the parity-SOS scheme. In the following, this scheme will be implied to as parity-SOS-ECC (or second proposed technique). The overheads of the two proposed schemes can be initially estimated using the number of additional FFTs and SOS check blocks needed. The set of *k* original FFT modules assuming *k* is a power of two. It can be observed that the two proposed schemes decrease the number of additional FFTs to just one. In addition, the second technique also decrease the number of SOS checks. In above mentioned a detailed evaluation for an FPGA implementation is discussed to illustrate the relative overheads of the proposed techniques. In all the techniques discussed, soft errors can also affect the elements added for protection.

For the ECC technique, the protection of these elements was discussed. 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 cause a correction[4]. In the case of SOS checks, an error will cause 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 in Figs.2 and 3 can propagate errors to the outputs. In our implementations, those blocks are protected with TMR. The same applies for the adders used to compute the inputs to the redundant FFTs in Fig.1 or to the SOS checks in Fig.2. The triplication of these blocks has a small impact on circuit complexity as they are much simplest 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 assurance 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.

B. EVALUATION

The two proposed arrangement and the ECC arrangement have been carried out on an FPGA and checked both in terms overhead and error coverage. A four-point decimation-in-frequency FFT core is used to compute the FFT iteratively. This core has been developed to implement MIMO-OFDM for wireless systems. The implementation of the four-point FFT core is shown in Fig. 4. The number of FFT points is programmable and the rotation coefficients are calculated on-line for each stage and stored in registers. For the evaluation, a 1024 points FFT is configured with five stages calculation (log41024 = 5), so in total 5∗1024 = 5120 cycles are needed to calculate the FFT for 1024 input samples [3].

The inputs are 12-bit wide and the outputs are 14-bit wide. For the redundant FFT, the bit ranges are prolonged to 14 and 16 bit, respectively, to cover up the higher dynamic range (as the inputs are the sum of several signals). Since both the inputs and outputs to the FFT are sequential, the SOS check is also done sequentially using accumulators that are compared at the end of the block. . To minimize the impact of round offs on the fault coverage, the outputs of the accumulator are 39-bit wide [5]. For the evaluation, several values of the number of parallel FFTs are considered. This is done to compare the different techniques as a action of the number of parallel FFTs in the original system.

The error detection and correction blocks are implemented as multiplexers that select the correct output depending on the error pattern detected [8]. As mentioned before, these blocks are tripled to ensure that errors that affect them do not exploiting the final outputs. The FFT and the different protection techniques have been implemented using Verilog.

Then, the arrangement has been mapped to a Virtex-4 xc4vlx80 FPGA setting the maximum effort on minimizing the use of resources. The results obtained are summarized in simulation results. The first table provides the resources needed to implement a single FFT and an SOS check. The results show that the FFT is more complicated than the SOS check as familiar. The differentiation will be much larger when a wholly parallel FFT implementation is used. The simulation results show the results when different number of parallel FFTs are protected.

The objective is to illustrate how the respective overheads of the different techniques vary with the number of parallel FFTs. In parentheses, the cost relative to an unprotected implementation is also provided. The results show that all techniques have a cost factor of <2. This demonstrates that the ECC-based technique proposed in [7] is also competitive to protect 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. On the other hand, the parity-SOS scheme needs less resources than the ECC scheme when the number of FFTs is 4, 6, or 8 but more when the number of FFTs is 11.

This can be explained as in the ECC arrangement, the number of additional FFTs grows logarithmically with the number of FFTs, while in the parity-SOS technique, the number of SOS checks grows linearly.

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

This means that as the number of FFTs to protect increases, the ECC arrangements becomes more competitive. For the parity-SOS-ECC arrangements, the number of SOS checks also grows logarithmically and they are simplist 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 arrangements 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. For larger values, the ECC arrangements outperforms the parity-SOS technique in our implementation. As a summary, the results show that the parity-SOS arrangements outperforms only the ECC arrangements for small number of parallel FFTs, and the parity-SOS-ECC arrangements always provides the best results. As mentioned before, a key aspect of any fault tolerant scheme is to validate that it can effectively correct errors. To that end, fault injection experiments have been done on the two proposed arrangements and the ECC only.

In each simulation run, one error is inserted to mimic the performance of soft errors that occur in isolation. In particular, 20 000 errors have been randomly injected on the registers for the Fourier coefficients and on the RAMs for the results of each stage of the FFT calculation, respectively [6]. For ECC protected parallel FFTs, a tolerance level of 1 is used for the equation checks. For example, all faults that introduce errors out of range of [−1, 1] were detected and corrected, which is the same as that reported for parallel FIR filters .For the parity-SOS and parity-SOS-ECC schemes, the fault coverage is determined by the tolerance level τ used in the Parseval check (the absolute difference between the input power and the output power) In the experiments, we have set $\tau = 1$, and the fault coverage is ∼99.9%, which is similar to the results reported in error detection technique in Fast Fourier Transform. This means that approximately 1 out of 1000 errors will not be corrected. Since soft errors are rare events, the residual error rate will be very low and therefore ,acceptable for many communication and signal processing applications.

IV. SIMULATION AND RESULTS

A.Simulation result of ECC

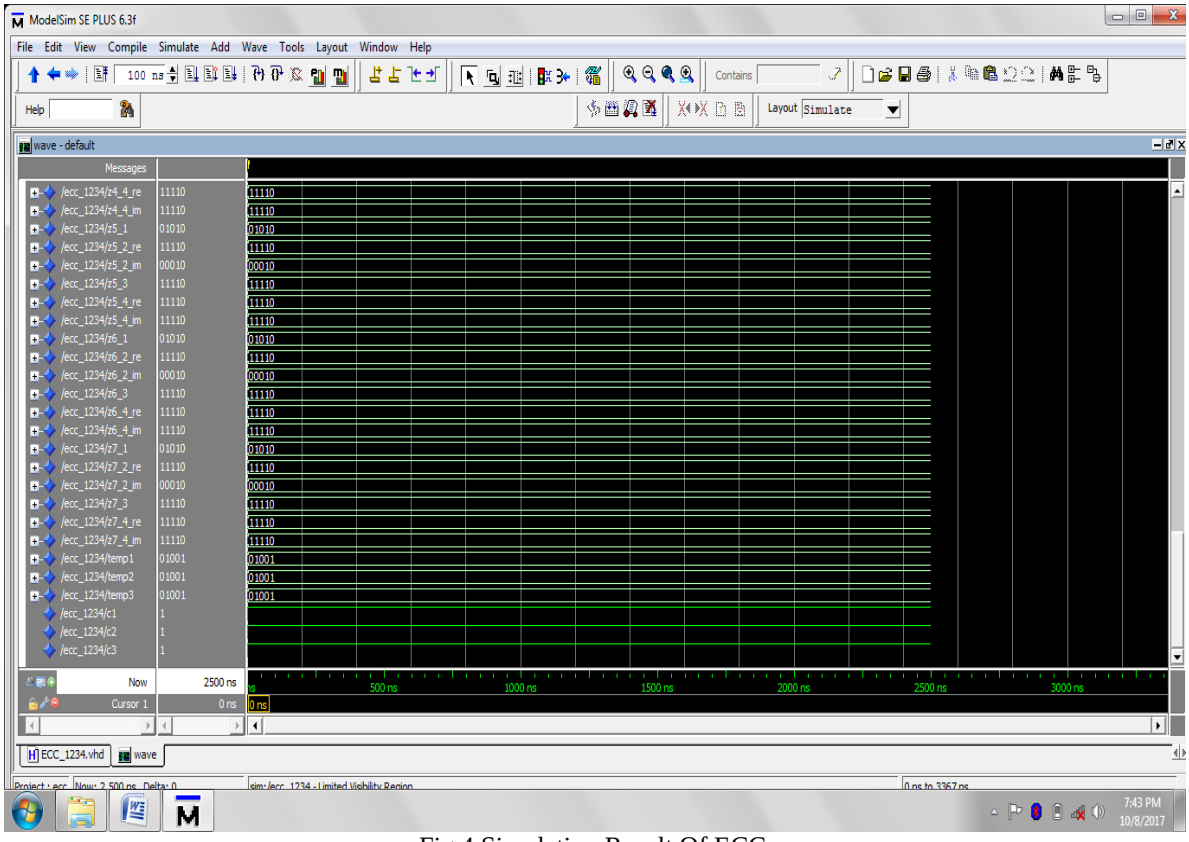

Fig.4 Simulation Result Of ECC

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

B.SIMULATION RESULT OF PARITY SOS

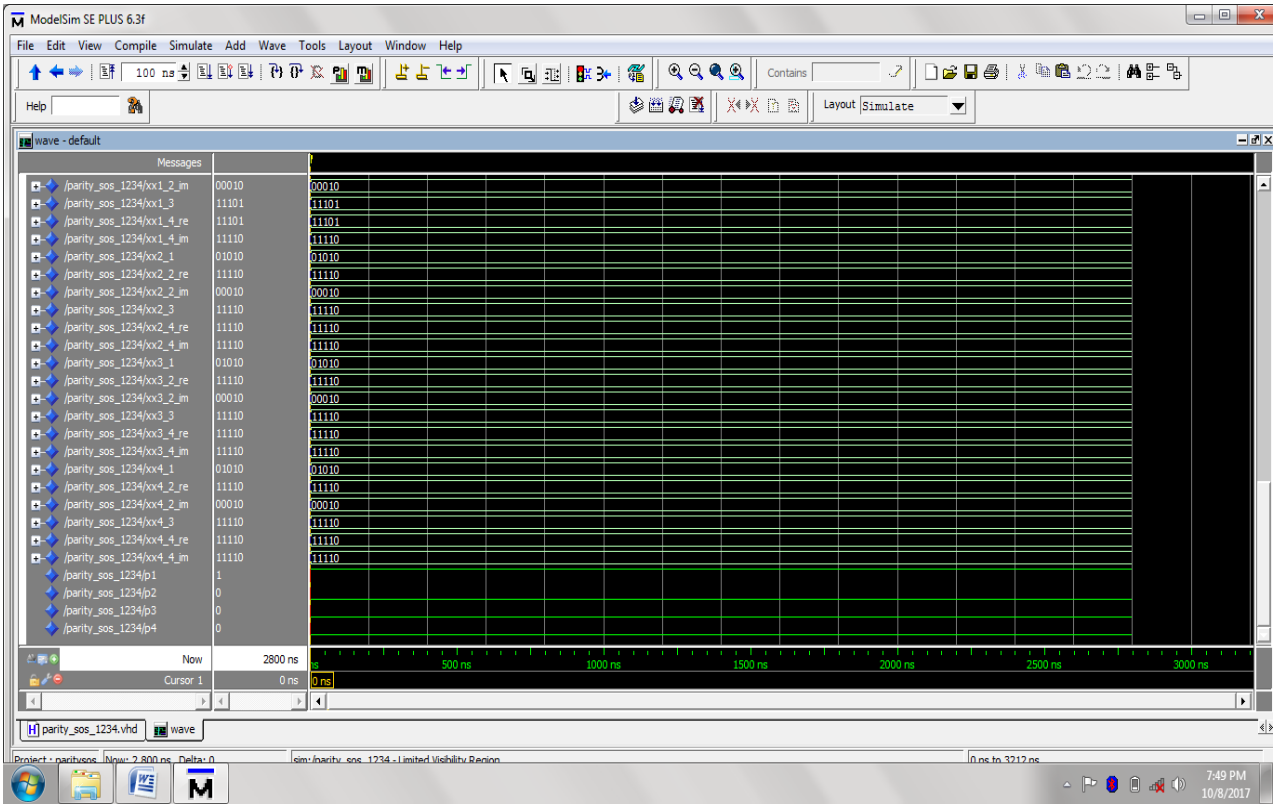

Fig.5 Simulation Result Of Parity SOS

C.SIMULATION RESULT OF PARITY-SOS-ECC

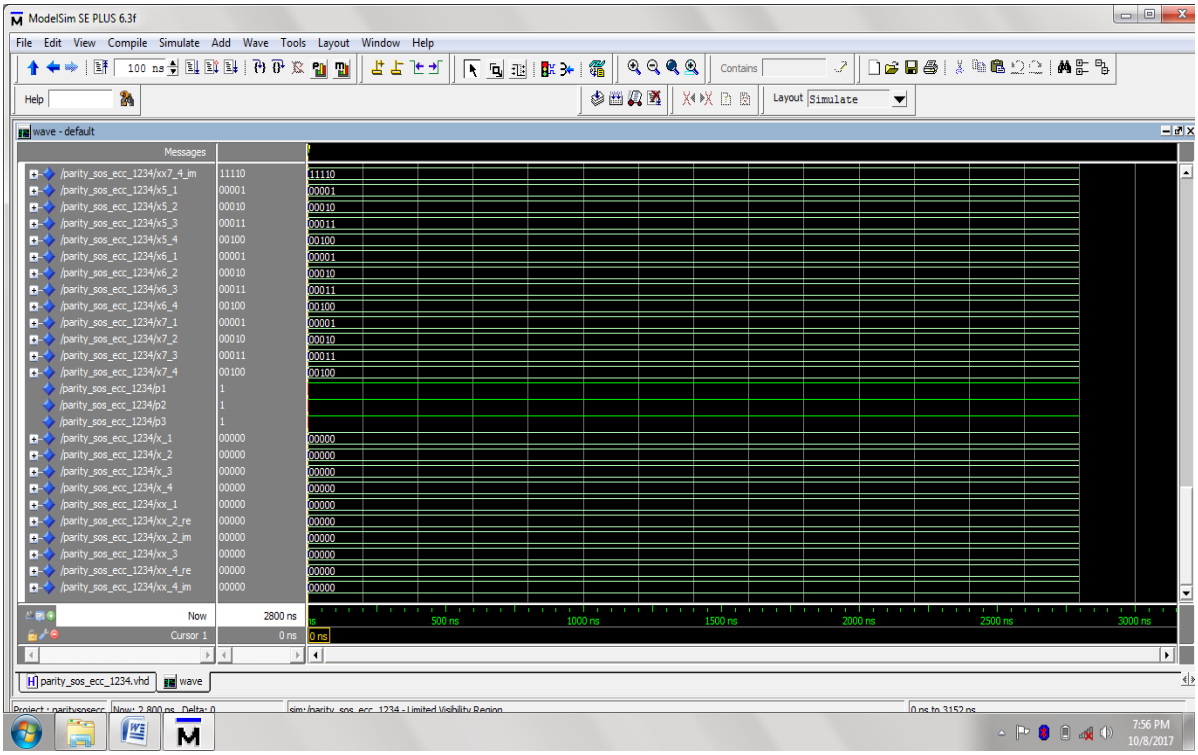

Fig.6 Simluation Result Of Parity SOS ECC

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 II Comparision and performance of ECC, Parity SOS & Parity-SOS-ECC

	ERROR CORRECTION CODE	PARITY SOS	PARITY-SOS-ECC
AREA	16.735	10.561	9.584
POWER CONSUMPTION	263Mw	253mW	252mW
DELAY	2.670ns	10.561ns	7.049ns

This section deals with simulation result and discussion of the existing **Error correction codes** and **PARITY SOS.** The software tool used is **XILINX**. The simulation and **power** analysis are **263 & 253Mw** respectively and **area** analysis are **16.735 & 10.561**and **delay** analysis are **2.670ns & 8.187ns** respectively are obtained with the help of XILINX tool.

This also deals with simulation result and discussion of the proposed **PARITY-SOS-ECC**. The software tool used is XILINX & MODELSIM. The simulation and **power** analysis as **252mW** and **area** analysis as **9.584** and delay analysis as **7.049ns** are obtained with the help of XILINX tool.

V. CONCLUSION

In this brief, the protection of parallel FFTs implementation against soft errors has been studied. Two techniques have been proposed and evaluated. 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 position of the errors can be done using an SOS check per FFT. The proposed techniques have been evaluated both in terms of implementation complexity and error detection capabilities. The results show that the second technique, which uses a parity FFT and a set of SOS checks that form an ECC, provides the best results in terms of implementation complexity. In terms of error protecting, fault booster analysis show that the ECC approach can get back all the errors that are out of the tolerance range. The fault coverage for the parity-SOS scheme and the parity-SOS-ECC scheme is ~99.9% when the tolerance level for SOS check is 1.

REFERENCES

[1] Byonghyo Shimand Naresh R. Shanbhag,(2006),"Energy-Efficient Soft Error-Tolerant Digital Signal Processing",IEEE trans.Very Large Scale Integr.(VLSI) Syst.,Vol.14,No.4,pp.336-348.
[2] Baumann R.(2005),"Soft Errors in Advanced Computer Systems",IEEE Des.Test Comput.,Vol.22,No.3,pp.258- 266.
[3] Eric P.Kim and Shanbhag N.R(2012),"Soft N modular redundancy" ,IEEE Trans .Comput.,Vol.61,No.3,pp.323-336.
[4] Gordon L.Stuber,Barry J.R.,McLaughlin S.W.,Li Y.,Ingram M.A.amd Pratt T.G.(2004),"Broadband MIMO-OFDM Wireless communications",Proc.ieee,. Vol.92,No.2,pp.271-294.
[5] Hitana T. and Deb A.K.(2004),"Bridging Concurrent and Non-concurrent error Detection in FIR filters",in Proc.Norchip Conf.,pp.75-78.
[6] Michael Nicolaidis (2005),"Design for Soft Error Mitigation",IEEEtrans.device.matter.rel,Vol.5,No.3,pp.405-418.
[7] Pontarelli S.Cardarilli G.C.Re M. and Salsano A.(2008),"Totally fault tolerant RNS based FIR filters",IEEE Int.On-line Test Symp(IOLTS),pp.192-194.
[8] Sying Jyan Wang and Niraj K.Jha(1994) ,"Algorithm based fault tolerance to FFT networks", IEEE Trans.comput,Vol.43,No.7,pp.849-854.