Low Area Advanced Single Fault Correction Technique for Parallel FFTs

Kalluri Manideepthi* & K. Niranjan Reddy & N. Neelima*****
*PG scholar, Dept. of ECE (VLSI), CMRIT, Kandlakoya (v), Medchal road, Hyderabad, TS, India.
**Associate Professor, Dept. of ECE, CMRIT, Kandlakoya (v), Medchal road, Hyderabad, TS, India.
***Associate Professor, Dept. of ECE, CMRIT, Kandlakoya (v), Medchal road, Hyderabad, TS, India.

Received: May 12, 2018 Accepted: June 13, 2018

ABSTRACT *The quality of communications and signal processing circuits will increase per annum. This can be created attainable by the CMOS technology scaling that permits the mixing of a lot of and a lot of transistors on one device. This increased quality makes the circuits a lot of prone to errors. At identical time, the scaling implies that transistors operate with lower voltages and square measure a lot of at risk of errors caused by noise and producing variations. Soft errors pose an irresponsibleness threat to fashionable electronic circuits. This makes protection against soft errors a demand for several applications. Communications and signal process systems are no exceptions to the current trend. For some applications, an interesting possibility is to use algorithmic-based fault tolerance (ABFT) techniques that try and exploit the recursive properties to sight and proper errors. Signal process and communication applications are compatible for ABFT. One example is quick Fourier transforms (FFTs) that are a key building block in several systems. Many protection schemes have been projected to sight and proper errors in FFTs. Among those, most likely the utilization of the Parseval or add of squares check is that the most generally glorious. In modern communication systems, it's more and more common to seek out several blocks in operation in parallel. Recently, a method that exploits this truth to implement fault tolerance on parallel filters has been projected. During this temporary, this system is 1st applied to guard FFTs. Then, 2 improved protection schemes that mix the utilization of error correction codes and Parseval checks are projected and evaluated. To decrease power utilization and delay due to redundant filters, we proposed another procedure by using Vedic multiplier. To enhance multiplier efficiency, we use Vedic multiplier i.e., Urdhwa Tiryakbhyam Sutra. By utilizing this, we can enhance the functionality of the magnitude square block in the parseval check. The proposed system scheme is first described and then illustrated. This would lead to a lower area overhead as compared to ECC based approach and Parseval checks approach. The results show that the proposed schemes can further reduce area overhead and cost of protection*

Keywords: *Error Correction Codes (ECCs), Fast Fourier Transforms (FFTs), Soft Errors, Vedic Multiplier.*

I. INTRODUCTION

The complexity of communications and signal processing circuits increases every year. This is made possible by the CMOS technology scaling that enables the integration of more and more transistors on a single device. This increased complexity makes the circuits more vulnerable to errors. At the same time, the scaling means that transistors operate with lower voltages and are more susceptible to errors caused by noise and manufacturing variations. The importance of radiation-induced soft errors also increases as technology scales. Soft errors can change the logical value of a circuit node creating a temporary 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. Those vary from modifications within the producing method of the circuits of the circuits to reduce the amount of errors to adding redundancy at the logic or system level to make sure that Errors don't have an effect on the system practicality.

Digital Filters square measure one among the foremost normally used signal process circuits and a number of other techniques are projected to guard them from errors. There square measure range of ways accustomed establishes faults and also the actions necessary to correct the faults at intervals circuit. Digital filters square measure wide used in signal process and communication systems. There square measure completely different fault tolerance approaches to typical process circuits and also the DSP circuits. In some cases, the dependableness of these systems is vital, and fault tolerant filter implementations square measure required. Over the years, several techniques that exploit the filters structure and properties to attain fault tolerance are projected.

One classical example is the use of triple modular redundancy (TMR) that triples a block and votes among the three outputs to detect and correct errors. The main issue with those soft errors mitigation techniques is that they require a large overhead in terms of circuit implementation. For example, 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%. This overhead is excessive for many applications. Another approach is to try

to use the algorithmic properties of the circuit to detect/correct errors. This is commonly referred to as algorithm-based fault tolerance. This strategy can reduce the overhead required to protect a circuit.

Signal processing and communications circuits are well suited for algorithm based fault tolerance techniques as they have regular structures and many algorithmic properties. Over the years, many algorithm based fault tolerance techniques have been proposed to protect the basic blocks that are commonly used in those circuits. Several works have considered the protection of digital filters. For example, the use of replication using reduced precision copies of the filter has been proposed as an alternative to TMR but with a lower cost. The knowledge of the distribution of the filter output has also been recently exploited to detect and correct errors with lower overheads. The protection of fast Fourier transforms has also been widely studied.

A general scheme based on the use of error correction codes has been proposed. In this technique, 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 technique can be used for operations, in which the output of the sum of several inputs is the sum of the individual outputs. This is true for any linear operation as, for example, the discrete Fourier transforms (DFT). In this brief, the protection of parallel FFTs 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 errors. There are four main contributions in this brief.

1) The evaluation of the ECC technique for the protection of parallel FFTs showing its effectiveness in terms of overhead and protection effectiveness.
2) The technique based on the use of Parseval or sum of squares (SOSs) checks combined with parity FFT. The technique on which the ECC is used on the SOS checks instead of on the FFTs.
3) The technique based on the use of Parseval or sum of squares (SOSs) checks by using Vedic mutiplier.
4) The technique on which the ECC is used on the SOS checks instead of on the FFTs by using Vedic multiplier.

The new techniques provide low area overhead and delay due to system.

1. <u>Parallel FFT Protection using ECCs</u>:

The two proposed techniques provide new alternatives to protect parallel FFTs that can be more efficient than protecting each of the FFTs independently. The existed schemes have been evaluated using FPGA implementations to assess the protection overhead. The results show that by combining the use of ECCs and Parseval checks, the protection overhead can be reduced compared with the use of only ECCs as existed. Fault injection experiments have also been conducted to verify,

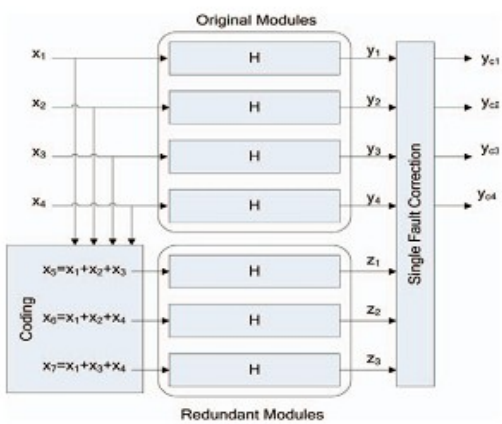

Fig.1. Parallel FFT protection using ECCs.

The ability of the implementations to detect and correct errors. The rest of this brief is organized as follows. Section II presents the two existed schemes. In Section III, Proposed System with Vedic multiplier architecture has explained. In Section IV the simulation result of the system is figured. Finally, the conclusions are drawn in Section V.

II. PROTECTION SCHEMES FOR PARALLEL FFTS

The starting 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. 1. In this example, a simple single error correction Hamming code is used. The original system consists of four FFT modules and three redundant

modules are 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

$$x_5 = x_1 + x_2 + x_3 \quad (1)$$

And since the DFT is a linear operation, its output z_5 can be used to check that

$$z_5 = z_1 + z_2 + z_3 \quad (2)$$

This will be denoted as c1 check. The same reasoning 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 different patterns and the corresponding errors are summarized in Table I. 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 affecting z1, this can be done as follows:

$$z_{1c}[n] = z_5[n] - z_2[n] - z_3[n]. \quad (3)$$

Similar correction equations can be used to correct errors on the other modules. More advanced ECCs 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, is lower than TMR as the number of redundant FFTs is related to the logarithm of the number of original FFTs. For example, to protect four FFTs, three redundant FFTs are needed, but to protect eleven, the number of redundant FFTs in only four. This shows how the overhead decreases with the number of FFTs.

2. <u>Parity-SOS (first technique) fault-tolerant parallel FFTs</u>:

In Section I, it has been mentioned that over the years, many techniques have been proposed to protect FFTs. 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).

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

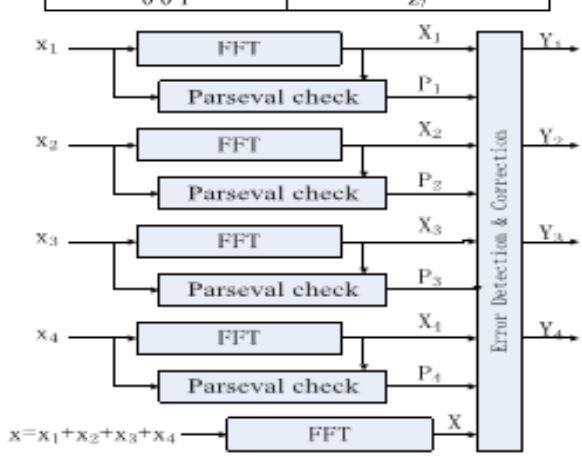

Fig. 2. Parity-SOS (first technique) fault-tolerant parallel FFTs

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 FFT can be used to correct the error. This is better explained with an example. In Fig. 2, the first proposed scheme is illustrated for the case of four parallel 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 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

$$X_{1c} = X - X_2 - X_3 - X_4. \quad (4)$$

This combination of parity FFT and the SOS check 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 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 in Fig. 3.

3. <u>Parity-SOS-ECC (second technique) fault tolerant parallel FFTs:</u>

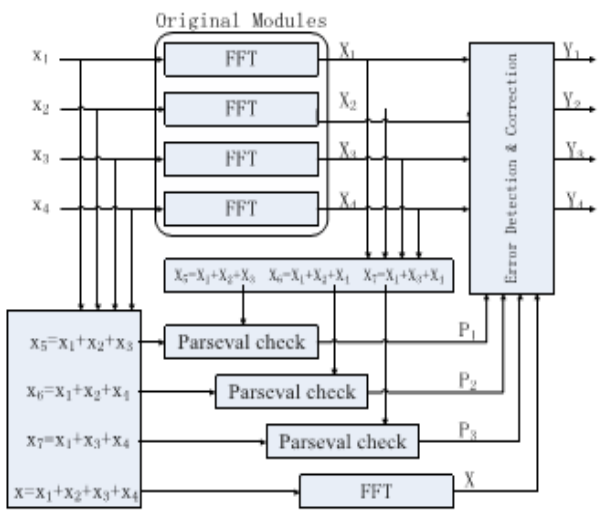

Fig. 3. Parity-SOS-ECC (second technique) fault-tolerant parallel FFTs.

TABLE II
OVERHEAD OF THE DIFFERENT SCHEMES TO PROTECT k FFTs

	FFTs	SOS Checks
ECC	1+log(k)	0
Parity-SOS	1	K
Parity-SOS-ECC	1	1+log(k)
Matrix method with ECC	2	0

The first parity-SOS scheme is to reduce 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 referred 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. This information is summarized in Table II 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. In addition, the second technique also reduces the number of SOS checks. 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 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 in Figs. 2 and 3 can propagate errors to the outputs. Those blocks are protecting 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. 3. The triplication of these blocks has a small impact on circuit complexity as they are much simpler than the FFT computations actually corrected. This means that an evaluation has to be done both in terms of overhead and error coverage.

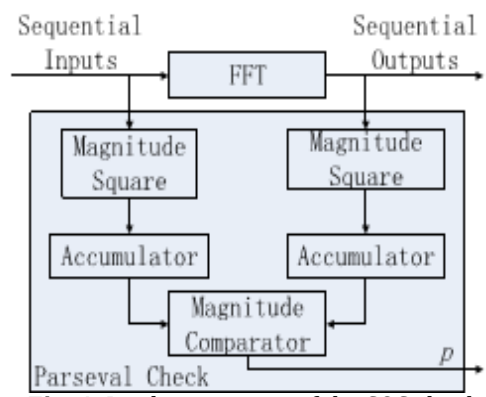

Fig. 4. Implementation of the SOS check

A new technique with Vedic multiplier in place of Booth multiplier, Thus it will decrease area overhead and delay due to the FFTs. So that it will improve the performance efficiency and low area overhead. Below section will describe about Vedic multiplier- Urdhwa Tiryakbhyam sutra.

III. PROPOSED SYSTEM

4. Coding for Fault Tolerant Parallel FFTs:

We proposed a low area advanced single fault correction technique for parallel FFTs using Vedic Multiplier. Vedic Multiplier is defined below,

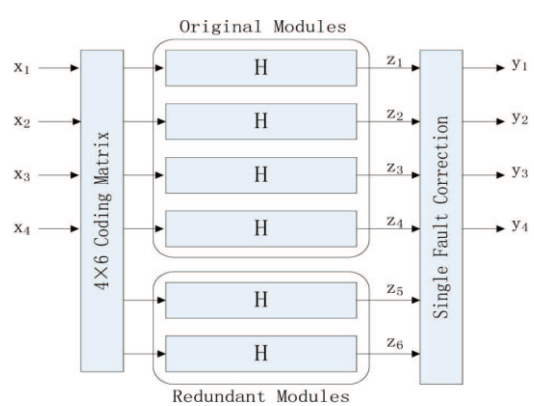

Fig. 5. Proposed coding scheme in general form

5. Vedic Sutra- Urdhwa Tiryakbhyam:

In proposed system we tend to area unit measurement Input Adder Unit, currently it is replaced by sacred text multiplier factor. By doing this we are able to get less power consumption, high accuracy and reduced delay.

The sixteen sacred text Sutras apply to and canopy nearly each branch of arithmetic. They apply even to advanced issues involving an oversized variety of mathematical operations. Among these sutras, Urdhwa Tiryakbhyam Sanskrit literature is that the best for acting multiplication. The use of this Sanskrit literature will be extended to binary multiplication as well .This Sanskrit literature interprets to "Vertical and crosswise". It utilizes solely logical AND operation, 0.5 adders and full adders to perform multiplication wherever the partial merchandise area unit generated before actual multiplication. This protects a substantial quantity of time interval. What is more it's a sturdy methodology of multiplication. Consider two 8-bit numbers, a (a8-a1) and b (b8-b1) wherever one to eight represents bits from the least important bit to the most important bit. The ultimate Product is represented by P (P16-P1). In Fig. 8, the step by step methodology of multiplication of two 8-bit numbers using Urdhwa Tiryakbhyam sutra is illustrated. The bits of the number and number area unit diagrammatic by dots and also the 2 approach are represents the logical AND operation between the bits that provides the partial product terms. In the typical style of Urdhwa Tiryakbhyam sutra based mostly number, solely full-adders and half-adders area unit used for addition of the partial products. But, the aptitude of full-adder is restricted to addition of solely three bits at a time. So, a large number of stages are required to get the final product. Higher order compressors discussed in next section can be employed to add more than 3 bits at a time (up to 7 bits) and hence can reduce the intermediate stages.

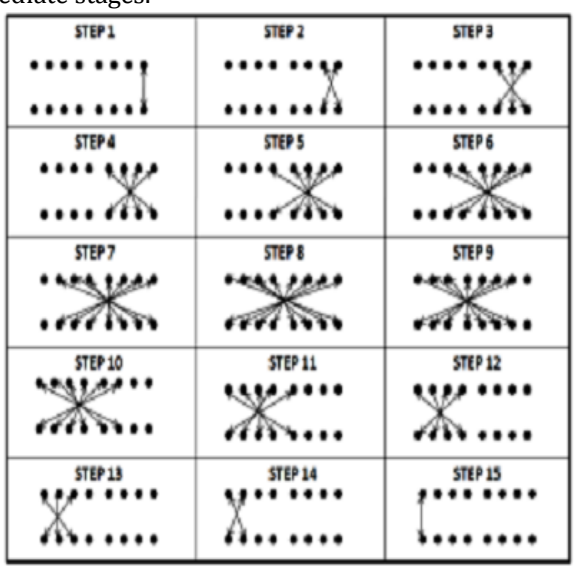

Fig.6. 8-Bit binary multiplication using Urdhwa Tiryakbhyam Sutra

The multiplier is based on an algorithm Urdhwa Tiryakbhyam (Vertical & Crosswise) of ancient Indian Vedic Mathematics. Urdhwa Tiryakbhyam Sutra is a general multiplication formula applicable to all cases of multiplication. It literally means "Vertically and crosswise". It is based on a novel concept through which the generation of all partial products can be done with the concurrent addition of these partial products. The parallelism in generation of partial products and their summation is obtained using Urdhawa Triyakbhyam. The algorithm can be generalized for n x n bit number. Since the partial products and their sums are calculated in parallel, the multiplier is independent of the clock frequency of the processor. Thus the multiplier will require the same amount of time to calculate the product and hence is independent of the clock frequency.

The net advantage is that it reduces the need of microprocessors to operate at increasingly high clock frequencies. While a higher clock frequency generally results in increased processing power, its disadvantage is that it also increases power dissipation which results in higher device operating temperatures.

By adopting the Vedic multiplier, microprocessors designers can easily circumvent these problems to avoid catastrophic device failures. The processing power of multiplier can easily be increased by increasing the input and output data bus widths since it has a quite a regular structure. Due to its regular

structure, it can be easily layout in a silicon chip. The Multiplier has the advantage that as the number of bits increases, gate delay and area increases very slowly as compared to other multipliers. Therefore it is time, space and power efficient. It is demonstrated that this architecture is quite efficient in terms of silicon area/speed.

IV. SIMULATION RESULT:

A. First Technique:

The written Verilog HDL Modules have successfully simulated and verified using Modelsim III 6.4b and synthesized using Xilinx ISE 10.1.

Simulation:

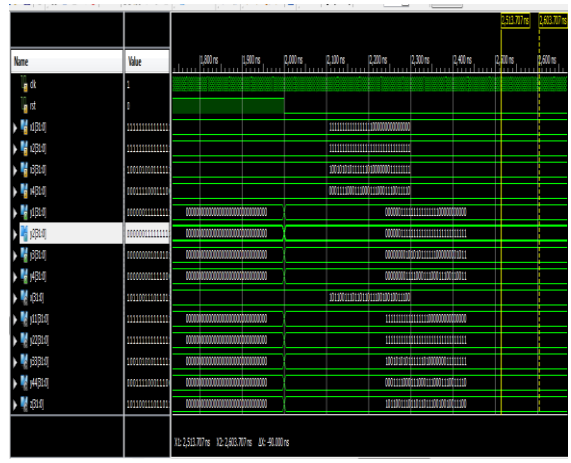

Fig.7. Simulated output wave form (First Technique)

In Fig.7, the inputs are given to the parallel FFTs, and then the outputs generated from those are fed to the parity SOS to detect and correct the errors. This combination of a Parity FFT and the SOS check reduces the number of additional FFTs to just one and may, therefore, reduce the protection overhead. 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. Here, in place of Booth's multiplier we used Vedic multiplier. In proposed system we tend to area unit measurement Input Adder Unit, currently it is replaced by sacred text multiplier factor. By doing this we are able to get less power consumption, high accuracy and reduced delay.

Synthesis Results:

The created venture is mimicked and checked their usefulness. Once the useful confirmation is done, the RTL display is taken to the union procedure utilizing the Xilinx ISE instrument. In union process, the RTL model will be changed over to the door level netlist mapped to a particular innovation library. Here in this Spartan 3E family, a wide range of gadgets were accessible in the Xilinx ISE apparatus. Keeping in mind the end goal to combination this outline the gadget named as "XC3S500E" has been picked and the bundle as "FG320" with the gadget speed, for example, "- 4".

This design is synthesized and its results were analyzed as follows,

RTL Schematic:

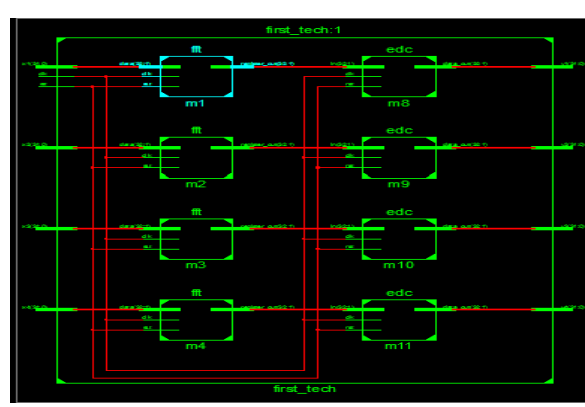

Fig.8. RTL Schematic for Parity-SOS (first technique) Fault-tolerant parallel FFTs by using Vedic multiplier.

The above Fig.8 represents the block diagram for Parity SOS technique. It shows the number of blocks used in formation of this technique.

Technology Schematic:

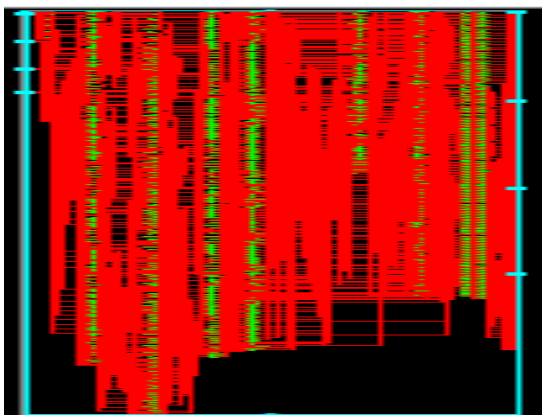

Fig.9. Technology Schematic for Parity-SOS (first technique) Fault-tolerant parallel FFTs by using Vedic Multiplier.

Technology Schematic for Parity SOS technique is the practical implementation of RTL modules. RTL are decomposed into lookup tables in FPGA. Fig.9 shows the technology schematic.

Design Summary:

Table III

Design Summary for the output of First Technique

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slices	254	4656	5%
Number of Slice Flip Flops	228	9312	2%
Number of 4 input LUTs	409	9312	4%
Number of bonded IOBs	254	232	109%
Number of GCLKs	1	24	4%

The above table III represents the design summary for parity SOS technique gives the area consumed in FPGA device for implementing the technique.

Timing Report:

The below screenshot represents the timing report for Parity SOS technique shows the delay used for the technique.

```
Offset:             4.040ns (Levels of Logic = 1)
  Source:           m8/data_26 (FF)
  Destination:      y1<25> (PAD)
  Source Clock:     clk rising

  Data Path: m8/data_26 to y1<25>
                              Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     FDR:C->Q             1   0.514   0.357  m8/data_26 (m8/data_26)
     OBUF:I->O                3.169          y1_25_OBUF (y1<25>)
    ----------------------------------------
    Total                     4.040ns (3.683ns logic, 0.357ns route)
                                      (91.2% logic, 8.8% route)
```

B. Second Technique:

Simulation:

In Fig.10, The inputs are given to the parallel FFTs, outputs generated from these are fed to the parity SOS-ECC to detect and correct the errors**.** The main benefit over the first parity SOS scheme is to reduce the number of SOS checks needed. The error location process is the same as for the ECC scheme and correction is as in the parity-SOS scheme. In the following, this scheme will be referred to as parity-SOS-ECC (or second proposed technique). The results show that the second technique, which uses 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 protection, fault injection experiments show that the ECC scheme can recover all the errors that are out of the tolerance range. This project is extended with FFT using Vedic multiplier.

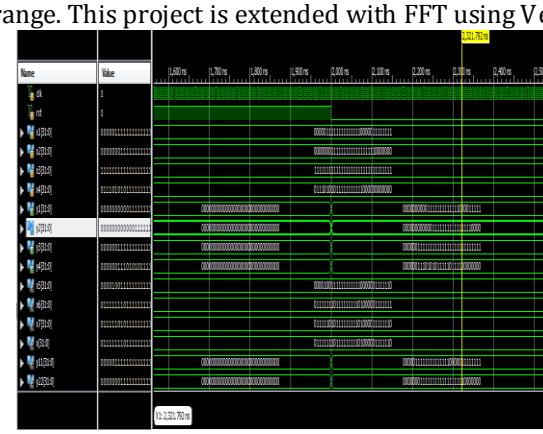

Fig.10. Simulated output wave form (Second Technique)

RTL Schematic:

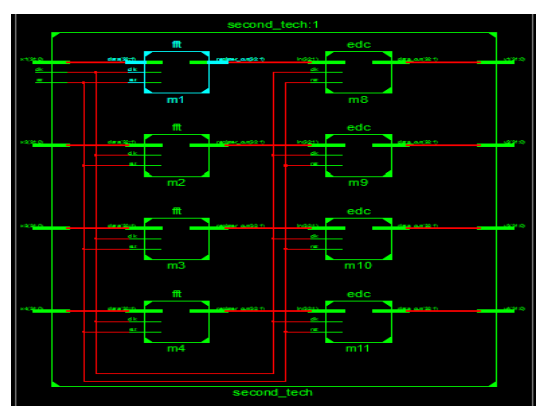

Fig.11. RTL Schematic for Parity-SOS-ECC (second technique) Fault-tolerant parallel FFTs

The above Fig.11 represents the block diagram for Parity SOS-ECC technique. It shows the number of blocks used in formation of this technique.

Technology Schematic:

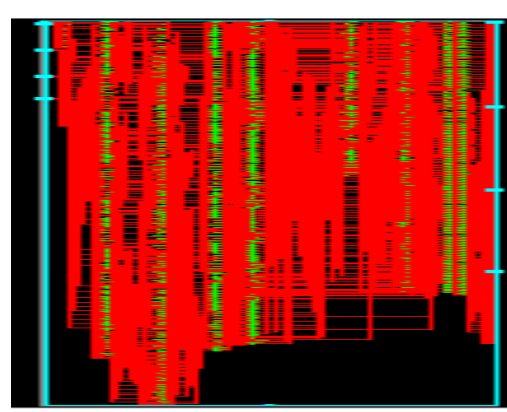

Fig.12. Technology Schematic for Parity-SOS-ECC (second technique) Fault-tolerant parallel FFTs by using Vedic multiplier

Technology Schematic for Parity SOS-ECC technique is the practical implementation of RTL modules. RTL are decomposed into lookup tables in FPGA. Fig.12 shows the technology schematic.

Design Summary:

The above table IV represents the design summary for parity SOS-ECC technique gives the area consumed in FPGA device for implementing the technique.

TABLE IV

Design Summary for the output of Second Technique

Device Utilization Summary (estimated values)			[-]
Logic Utilization	Used	Available	Utilization
Number of Slices	225	4656	4%
Number of Slice Flip Flops	229	9312	2%
Number of 4 input LUTs	421	9312	4%
Number of bonded IOBs	234	232	100%
Number of GCLKs	1	24	4%

Timing Report:

The below screenshot represents the timing report for Parity SOS-ECC technique shows the delay used for the technique.

```
Offset:              4.040ns (Levels of Logic = 1)
  Source:            m8/data_26 (FF)
  Destination:       y1<25> (PAD)
  Source Clock:      clk rising

  Data Path: m8/data_26 to y1<25>
                               Gate     Net
    Cell:in->out      fanout   Delay    Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     FDR:C->Q             1   0.514    0.357  m8/data_26 (m8/data_26)
     OBUF:I->O                3.169           y1_25_OBUF (y1<25>)
    ----------------------------------------
    Total                     4.040ns (3.683ns logic, 0.357ns route)
                                      (91.2% logic, 8.8% route)
```

TABLE V

Comparison Table:

SYSTEM	AREA			DELAY(ns)
	SLICES	LUTS	Flip-Flops	
EXISTING	266	429	228	4.0401ns
PROPOSED	254	409	228	4.04ns

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 location of the errors can 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. The results show that the second technique, which uses 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 protection, fault injection experiments show that the ECC scheme can recover all the errors that are out of the tolerance range. The proposed Vedic multiplier circuit using Urdhwa Tiryakbhyam Sutra can be implemented in arithmetic and logical units of a DSP processor replacing the traditional circuits. This project is extended with Vedic multiplier. For the further improvement of the multiplier efficiency, we use Vedic multiplier - Urdhwa Tiryakbhyam Sutra. By using this, we can improve the functionality of the magnitude square block in the parseval check. This technique can reduce the area and power for parallel filters with large number of FIR filters. The conventional multiplier has been replaced by Booths multiplier which has further reduced the area and has increased the operating frequency.

Applications:

- Digital signal processing and Image Processing
- High Speed FFT processor using Vedic Multiplier
- High Speed Accelerator
- Spectrum Analyzer

Future Scope:

In Future, use of Vedic multiplier in Digital signal processing will be extended. Since SOS-ECC strategy will takes huge territory, control utilization and number of excess channels increments per each piece will diminish by Vedic Multiplier. Fast and secured error correction will happen.

REFERENCES

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 Trans. VLSI, vol. 24, no. 2, February 2016.
2. N. Kanekawa, E. H. Ibe, T. Suga, and Y. Uematsu, Dependability in Electronic Systems: Mitigation of Hardware Failures, Soft Errors, and Electro-Magnetic Disturbances. New York, NY, USA: Springer-Verlag, 2010.
3. R. Baumann, "Soft errors in advanced computer systems," IEEE Des. Test Comput., vol. 22, no. 3, pp. 258–266, May/Jun. 2005.
4. M. Nicolaidis, "Design for soft error mitigation," IEEE Trans. Device Mater. Rel., vol. 5, no. 3, pp. 405–418, Sep. 2005.
5. A. L. N. Reddy and P. Banerjee, "Algorithm-based fault detection for signal processing applications," IEEE Trans. Comput., vol. 39, no. 10,1304–1308, Oct. 1990.
6. 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, pp. 336–348, Apr. 2006.
7. J. Y. Jou and J. A. Abraham, "Fault-tolerant FFT networks," IEEE Trans. Comput., vol. 37, no. 5, pp. 548–561, May 1988.
8. S.-J. Wang and N. K. Jha, "Algorithm-based fault tolerance for FFT networks," IEEE Trans. Comput., vol. 43, no. 7, pp. 849–854, Jul. 1994.
9. P. P. Vaidyanathanm, Multirate Systems and Filter Banks. Englewood Cliffs, NJ, USA: Prentice-Hall, 1993.
10. A. Sibille, C. Oestges, and A. Zanella, MIMO: From Theory to Implementation. San Francisco, CA, USA: Academic, 2010.
11. S. Sesia, I. Toufik, and M. Baker, LTE—The UMTS Long Term Evo-lution: From Theory to Practice, 2nd ed. New York, NY, USA: Wiley, Jul. 2011.
12. P. Reviriego, S. Pontarelli, C. J. Bleakley, and J. A. Maestro, "Area efficient concurrent error detection and correction for parallel filters," IET Electron. Lett., vol. 48, no. 20, pp. 1258–1260, Sep. 2012.
13. Z. Gao et al., "Fault tolerant parallel filters based on error correction codes," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 23, no. 2, pp. 384–387, Feb. 2015.
14. R. W. Hamming, "Error detecting and error correcting codes," Bell Syst. Tech. J., vol. 29, no. 2, pp. 147–160, Apr. 1950.
15. P. Reviriego, C. J. Bleakley, and J. A. Maestro, "A novel concurrent error detection technique for the fast Fourier transform," in Proc. ISSC, Maynooth, Ireland, Jun. 2012, pp. 1–5.
16. T. Hitana and A. K. Deb, "Bridging concurrent and non-concurrent error detection in FIR filters," in Proc. Norchip Conf., Nov. 2004, pp. 75–78.
17. 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