MAT JOURNALS
Journal of Signal Processing
Volume 4 Issue 3

Improved Fault Tolerance for Parallel FFT

[1]Rinciya Beema, [2]Dr.Gnana Sheela K
[1]PG scholar, [2]Professor
Department of Electronics & Communication Engineering, APJ Abdul Kalam Technological University, Kerala, India
Email: *[1]rinciyabeema94@gmail.com, [2]sheelabijins@gmail.com*
DOI: *http://doi.org/10.5281/zenodo.2222262*

Abstract

Digital filters are mainly used in signal processing and communication systems. Soft errors are major threats of the modern communication system. Now a day's complexity of communication and signal processing increases day by day. So reliability of the system is critical and fault tolerance technique is needed. In parallel FFT protection using ECC and parseval check can be used for error detection and correction. In this project is to reduce the entire area of the fault tolerant system by using pipelined method. In pipelined FFT consist of two butterfly structure. Each stage is replaced by a single butterfly structure. Experimental results show that proposed technique reduces the area and delay of the system. This project is mainly focused on the area of the system. Proposed technique is more efficient and reduces the complexity of the entire system.

Index Terms: *FFT, ECC, Pipelining, Parallel, Soft errors*

INTRODUCTION

Soft error is a major threat of modern electronic circuit. Soft error is a signal or datum which is wrong but it is not assumed to imply such a mistake or breakage [7]. Due to this soft error, communication can be interrupted. In order to reduce this, fault tolerance can be needed. Soft error can change the logical value of the system that makes temporary error in the entire systems. Redundancy check can be used for detecting the error. The technique is very complex. There are different techniques are used for fault tolerance technique. This system can reduce the entire overhead of the system. Error tolerance is a method of detecting and correcting errors in the entire system by parallel fault tolerant method [3]. This technique is mainly used for protecting FFT and it is mainly used parallel filter because complex system contains parallel filters. Mainly used technique is called algorithm based fault tolerance for error correction and detection. Different techniques can be used for protecting parallel filter in which each filter in a bit is equivalent to a traditional ECC. The other improved protection for FFT is combining the use of error correction code and parseval check. Mainly there are three techniques can be used for mitigation of soft errors.

1. Evaluation of Error Correction Code
2. Based on the use of parsevel check combine with FFT
3. ECC combines with parseval check

The above three techniques are mainly used for parallel FFT correction and detection. First method is checking the error by using hamming code followed and it is based on parseval theorem. Then the third technique is combination of these is two techniques. 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. These techniques need large area, power and

need more components. Thus pipelined FFT reduce the area and delay of the entire system,

RELATED WORKS

PARALLEL FFT PROTECTION USING ECC

The system has four original FFT and three temporary modules as shown in fig 1 .The error in this system can be removed by temporary modules represent in the system.

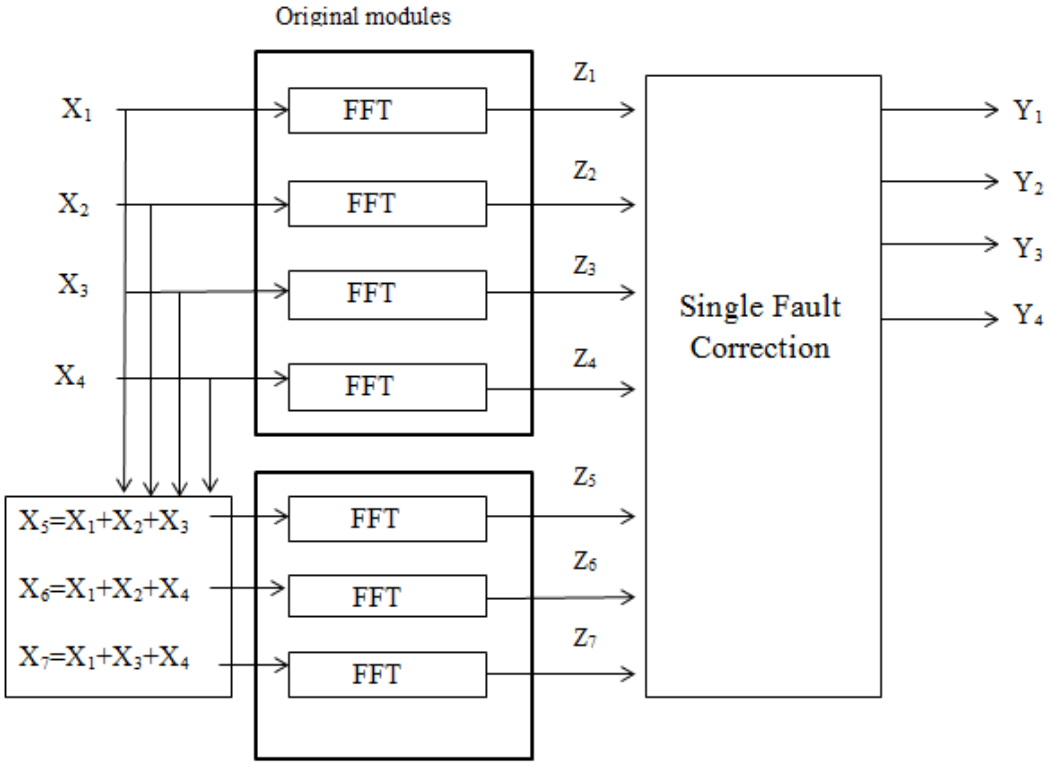


Fig: 1. *Parallel FFT Protection Using ECC.*

This technique is used for protecting the FFT independently. In this system which uses hamming code for finding the error.

The first redundant module 'x5' is the linear combination of the original module 'x1','x2','x3'.so we can write

$x5 = x1 + x2 + x3$. (1)

And the output of the redundant module 'Z5'is also the linear combination of the original module 'Z1','Z2','Z3'.

$So\ Z5 = Z1 + Z2 + Z3$.

In this way the second redundant module 'x6' is the linear combination of the original module 'x1','x2','x4'.Thus it can write

$X6 = x1 + x2 + x4$ (2)

And the output of the redundant module 'Z5'is also the linear combination of the original module 'Z1','Z2','Z4'.

So $Z6 = Z1 + Z2 + Z4$.

The third redundant module 'x7' is the linear combination of the original module 'x1','x3','x4'.so we can write

$X7 = x1 + x3 + x4$ (3)

And the output of the redundant module 'Z5', is also the linear combination of the original module 'Z1','Z3','Z4'.

So $Z7 = Z1 + Z3 + Z4$.

Here, used adder for adding the three original module and compare the output added value with first redundant module and this will be denoted as c1 check. The same technique applied to the other three redundant modules and it is denoted as c2

and c3 check. Using this c1, c2 and c3 we can easily find the error bit position. This pattern summarized in the table 1.In this way we can detecting the error in FFT.

Table: 1. *Error Table*

c1 c2 c3	ERROR BIT POSITION
000	NO ERROR
111	Z1
110	Z2
101	Z3
011	Z4

The error is finding in first module ie (Z1) this can be done as follows.

$Z1 = Z5 - Z2 - Z3$ ie subtracting the redundant module from the original module .

Similarly if the error in the second module ,third module and fourth module can be done as follows

$Z2 = Z6 - Z1 - Z3$

$Z3 = Z5 - Z1 - Z2$

$Z4 = Z7 - ZI - Z3$

The demerit is to need additional FFT for single correction and detection. This increase the complexity of the circuit.

PARITY-SOS-FAULT TOLERANT FFT

This technique is based on parseval theorem.Parseval theoem state that sum of squares of input of FFT is equal to the sum of squares of output of FFT. This method is used for detecting and correcting the FFT. In error free FFT the output sum square is equal to the input sum square.Multiple error correction and detection can done by using this method.

Here the input of the FFT is given to parseval check and it check the sum of squares of input and output. If the input and output square is equal there is no error. This check can be determined by 'P1','P2','P3','P4'.When the error is detected, the output of the parity FFT can be the redundant FFT is added to the sum of input of the original FFT.

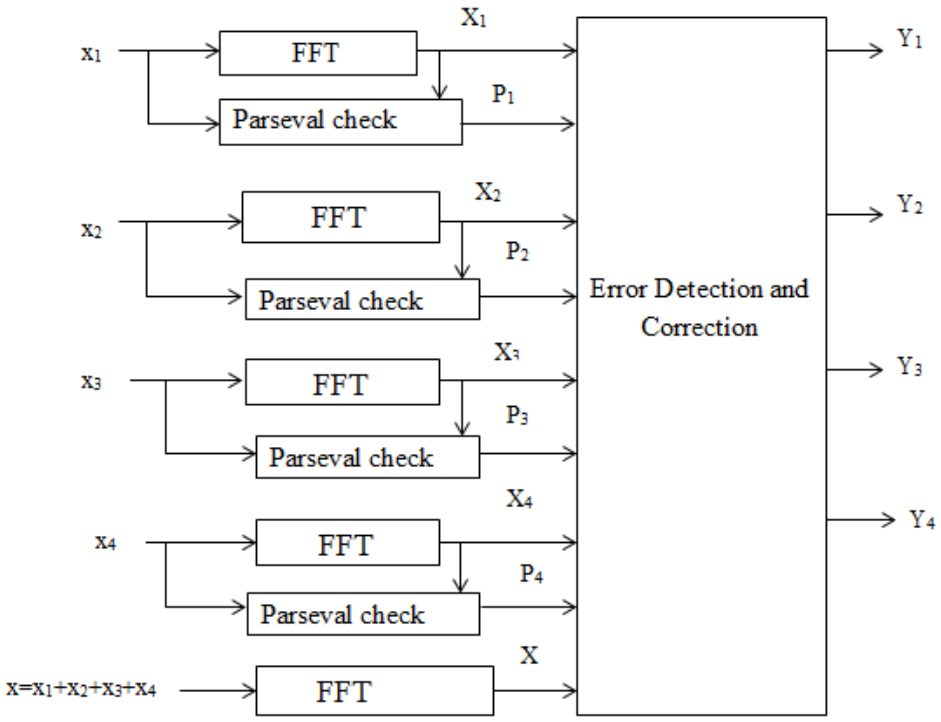


Fig: 2. *Parity SOS Fault Tolerant*

For example if the error is detecting in P1. The error can be corrected using the equation

$$P1 = X - X2 - X3 - X4$$

If the error is detecting in P2. The error can be corrected using equation

$$P2 = X - X1 - X3 - X4$$

If the error is detecting in P3. The error can be corrected using equation

$$P3 = X - X1 - X2 - X4$$

If the error is detecting in P4. The error can be corrected using equation

$$P4 = X - X1 - X2 - X3$$

The combination of parity FFT and SOS check reduce the number of FFT by just one. And it increasing the accuracy of detecting error.

PARITY SOS-ECC-FAULT TOLERANT PARALLEL FFT

This method is the combination of ECC and the Parseval check as shown in fig 3. In this method, ECC for SOS is used. The additional parity FFT is used for detecting and correcting error.

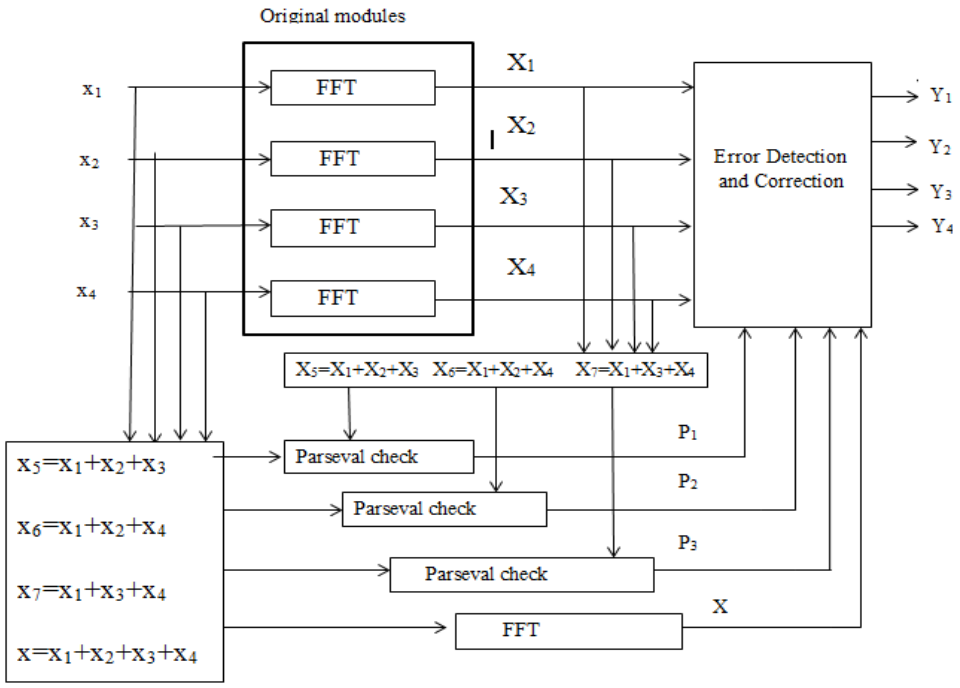


Fig: 3. *Parity SOS-ECC*

Here the linear combination of the input FFT and output of the FFT is given to the parseval check, and it checks the equality of input sum square sum and output sum square. If there is error detected, it can be corrected using parity FFT. The error location can be detected using error table as same as that of the first technique.

Thus soft error can added the element for protection. In parity FFT, the error will not propagate to the data output and will not trigger any correction. In SOS check the error will trigger correction, if there is no error on FFT. So it will produce corrected result. The final observation is that ECC detect all the error that exceed a given threshold but SOS doesn't detect all the error. So the accuracy of this above three technique is not much guaranteed and also there is chance of large overhead.

PROPOSED SYSTEM

BUTTERFLY 1 STRUCTURE

Fig 4 (a) shows the Butterfly I structure, it consists of two inputs Ar, Ai and a control signal C1. There are two feedback register for storing subtracted value. Here, these two input comes from the FFT input data. The output data Br, Bi goes to the next stage which is normally the Butterfly II. The control signal C1 has two cases when C1=0, multiplexers direct the input data to the feedback registers. The other case is when C1=1 the multiplexers select the output of the adders and subtracters. Here used D flip flop for stores the values.

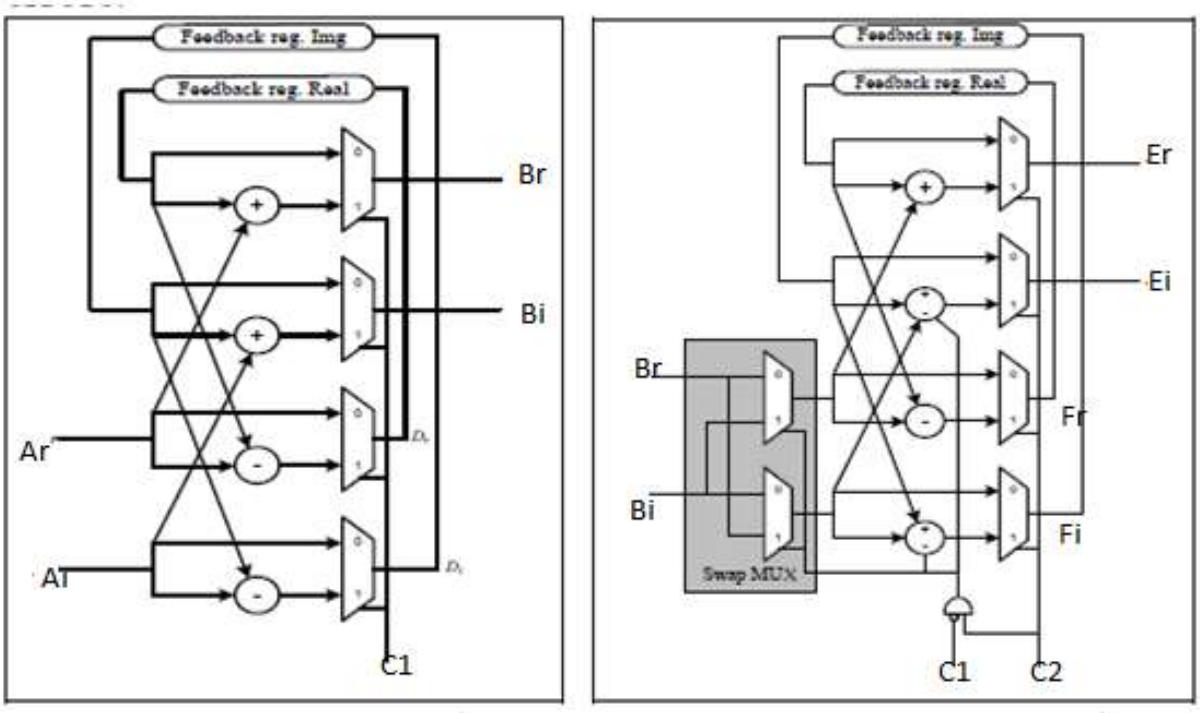
The process of the Butterfly I is start when C1=0, then it store the first input in to the feedback register followed by C1=1, then it select added or subtracted value .The subtracted values are Dr and Di stores in the D flip-flop. The result of the butterfly is Br, Bi, Dr, Di. Br, Bi fed to the output result of the Butterfly I the other result Dr, Di goes to the feedback registers.

BUTTERFLY 2 STRUCTURE

Fig. 2.4 (b) shows the Butterfly II structure. The input data Br, Bi comes from the stage 1. The output data from the Butterfly II are Er, Ei, Fr and Fi. Fr and Fi are the subtracted output goes to the fee back register, Er, Ei fed to the stage 3.Here twiddle factor multiplication can done by

using multiplier. Multiplication of –j means swapping of the real part and imaginary part and sign inversion.

The swapping is done by the multiplexers Swap-MUX efficiently and the sign inversion is handled by using the adding and the subtracting operations by mean of Swap-MUX. Two control signal C1 and C2 is used for the entire operation .When C1=0 and C2=1 swapping can be performed. Then the real and imaginary values are swapped.



(a) (b)

Fig: 4. *Butterfly structure (a) Butterfly I structure (b) Butterfly II structure.*

PARITY SOS-ECC-FAULT TOLERANT PIPELINED FFT

This method is the combination of ECC and the Parseval check as shown in fig 5. In this method, it can using ECC for SOS instead of using SOS per FFT. The additional parity FFT is used for detecting and correcting error. Here we used pipelined FFT instead of using parallel FFT. The main advantage of the system is the reduction of the entire area and delay of the system.

Here the linear combination of the input FFT and output of the FFT is given to the parseval check, and it checks the equality of input sum square sum and output sum square. If there is error detected, it can be corrected using parity FFT. The error location can be detected using error table as same as that of the existing technique.

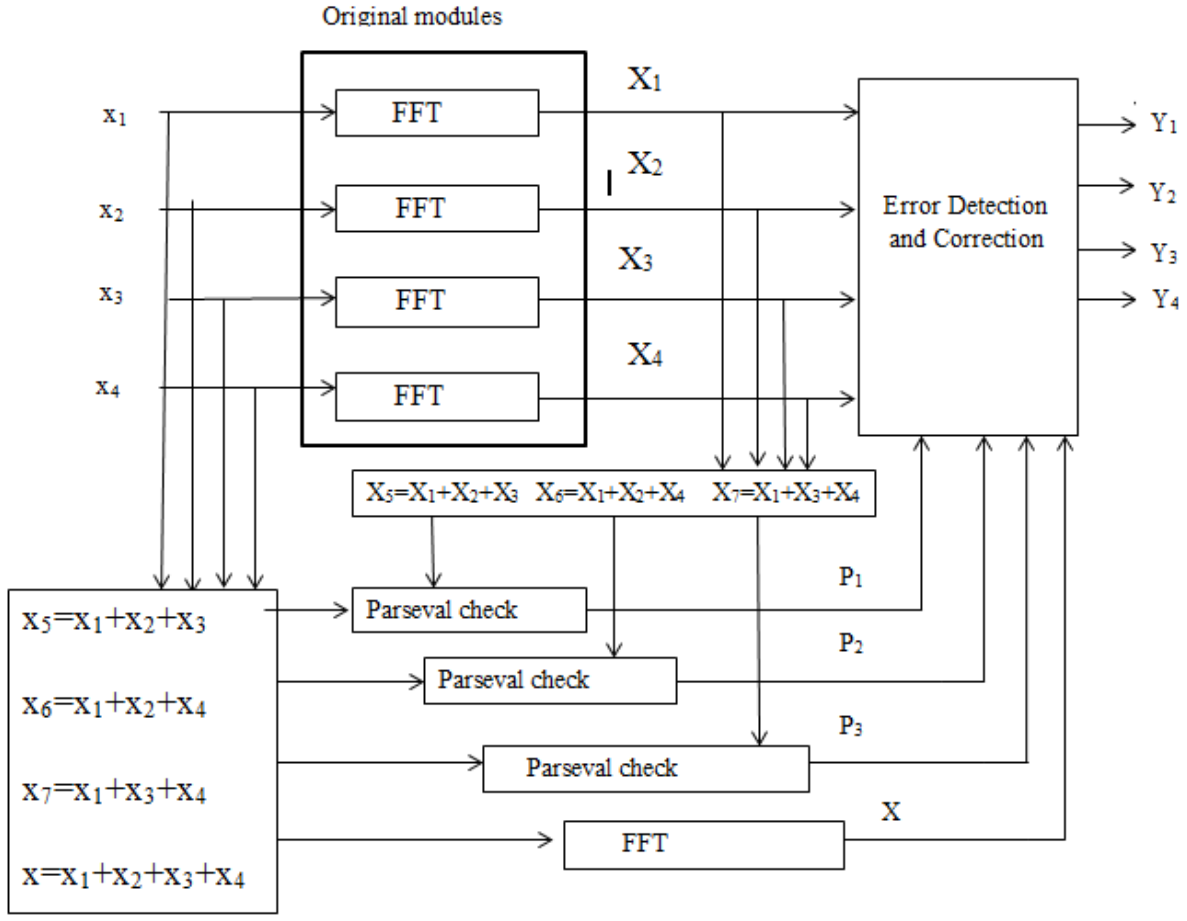


Fig: 5. *Pipelined Parity SOS-ECC*

RESULTS AND DISCUSSION

In this research work, an improved fault tolerant implementation of parallel FFT is presented and performed. The typical fault tolerant system detects and correct the output of the FFT. If error is detected it corrected by suitable techniques. The result of the proposed method is compared with existing method by using LUT's.

Table: 2. *Comparative Analysis of Parallel and Pipelined FFT*

METHODS	NO. OF SLICES	DELAY
ECC	24262	5.89
PARITY SOS	40472	6.12
PARITY SOS ECC	42260	5.90
PARITY SOS ECC PIPELINED	11842	4.89

In pipelined parity SOS ECC method, reduction of area of the entire system is achieved by reducing the number of LUT's. This system increases the processing speed of the entire system.

SYNTHESIS RESULT

PARALLEL FFT PROTECTION USING ECC

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice LUTs	23903	63400	37%
Number of fully used LUT-FF pairs	0	23903	0%
Number of bonded IOBs	1152	210	548%
Number of DSP48E1s	56	240	23%

Fig: 6. *Synthesis result of ECC*

PARITY-SOS FAULT TOLERANT FFT

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice LUTs	44181	63400	69%
Number of fully used LUT-FF pairs	0	44181	0%
Number of bonded IOBs	1152	210	548%
Number of DSP48E1s	148	240	61%

Fig: 7. *Synthesis result of parity SOS*

PARITY SOS-ECC-FAULT TOLERANT PARALLEL FFT

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice LUTs	41541	63400	65%
Number of fully used LUT-FF pairs	0	41541	0%
Number of bonded IOBs	1152	210	548%
Number of DSP48E1s	184	240	76%

Fig: 8. *Synthesis result of parity-SOS-ECC*

PIPELINED PARITY-SOS-ECC

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Registers	1733	126800	1%
Number of Slice LUTs	11901	63400	18%
Number of fully used LUT-FF pairs	962	12672	7%
Number of bonded IOBs	194	210	92%
Number of BUFG/BUFGCTRLs	1	32	3%
Number of DSP48E1s	61	240	25%

Fig: 9. *Synthesis result of Pipelined parity SOS ECC*

SIMULATION RESULT

PARALLEL FFT PROTECTION USING ECC

Fig 10 shows the simulation result of a single input of 8 point FFT with 14 outputs. Error correction has achieved as shown in the figure.

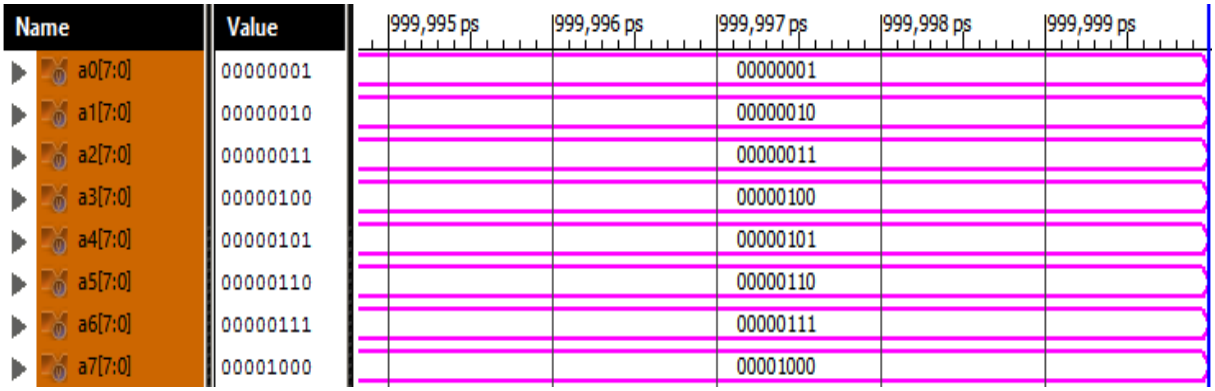


Fig: 10. *ECC Method Simulation*

PARITY-SOS-FAULT TOLERANT FFT

Figure 11 shows the simulation result of single input of parity SOS fault tolerant FFT. Error detection and correction has achieved.

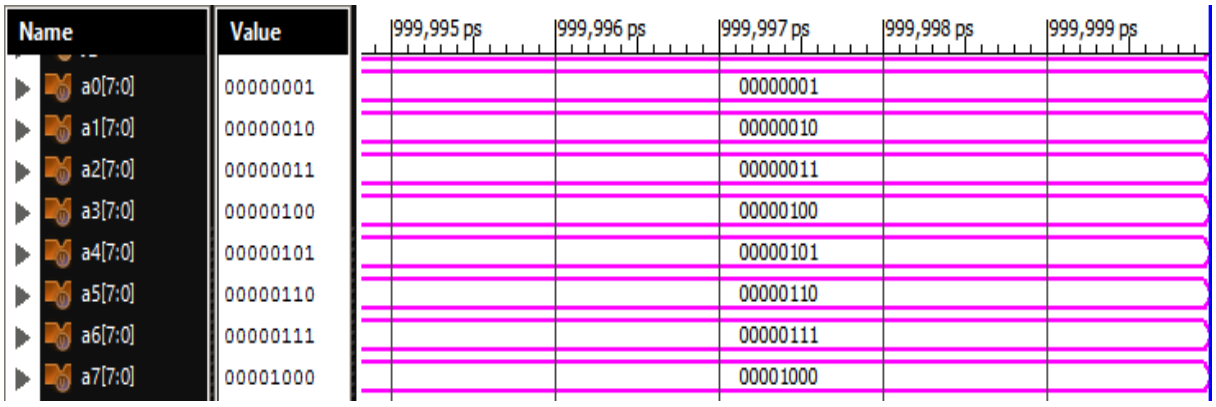


Fig: 11. *Parity SOS Simulation*

PARITY SOS-ECC-FAULT TOLERANT PARALLEL FFT

The simulation result of single input of parity SOS ECC is shown in fig 12. Error detection and correction has achieved in this method.

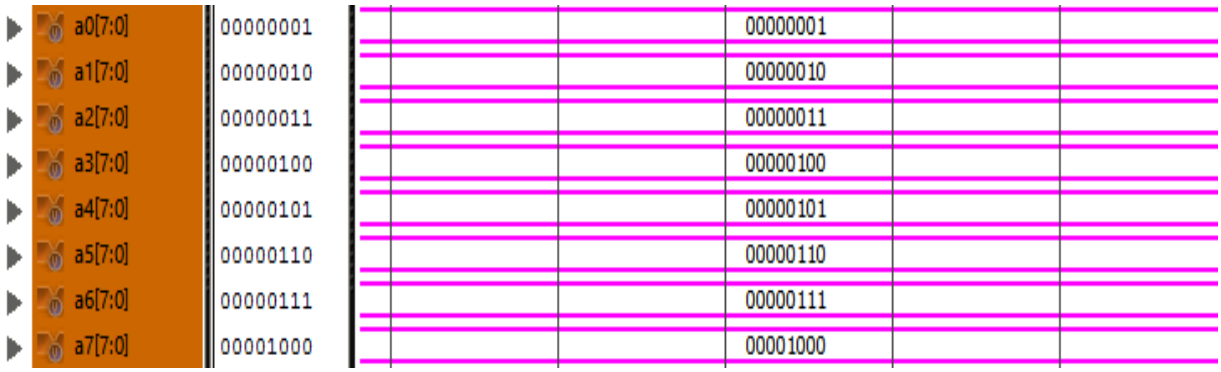


Fig: 12. *Parity SOS-ECC Simulation*

PARITY SOS-ECC-FAULT TOLERANT PIPELINED FFT

The simulation result of pipelined parity SOS ECC fault tolerant system is shown in fig 13. Error detection and correction has achieved in this method.

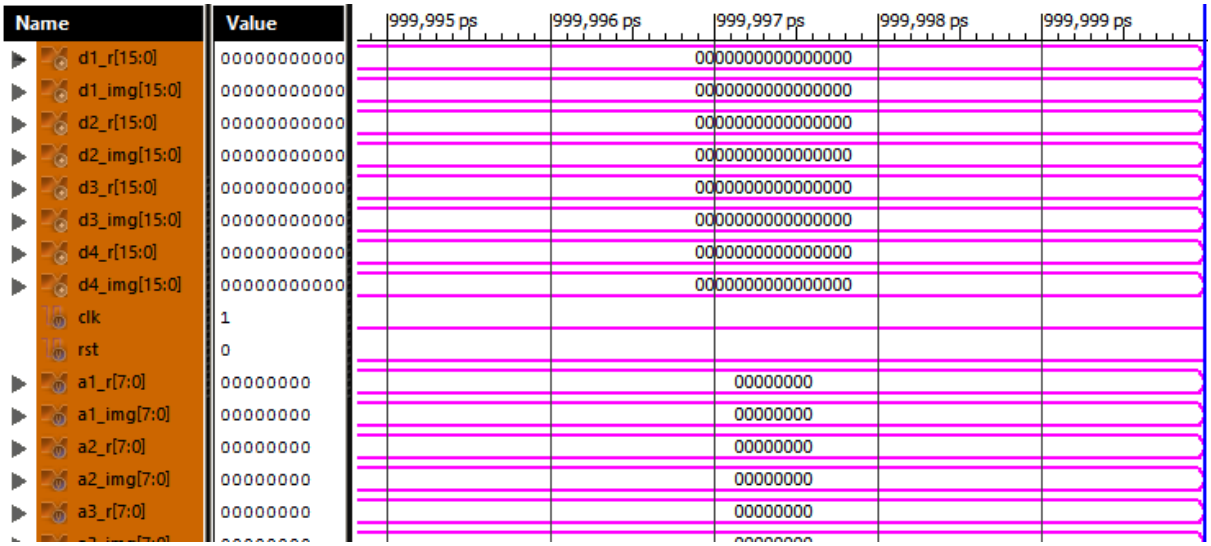


Fig: 13. *Pipelined ECC-SOS Method Simulation*

CONCLUSION

In this method, protection of pipelined FFT against soft error is studied and verified the result using Xilinx ISE design. The proposed pipelined FFT is used in analysis. The proposed technique is error correction and detection using pipelined FFT. The technique is the combination of ECC and SOS techniques. This technique which can reduces the circuit complexity by reducing the area and delay of the system. Thus reduce the entire area and circuit complexity of the system. In future it is possible to develop a system which checks multiple errors in the FFT.

REFERENCES

1. R.W Hamming, "Error correcting and error detecting codes", Bell system technical journal, vol 29, no.2, Apr 1950
2. Algirdas, Avizienis, Fellow, Ieee, "fault-tolerant systems IEEE," transactions on computers, vol. C-25, no. 12, December 1976
3. Jing-Yang Jou, Jacob A Abraham, "Fault-Tolerant FFT Networks," IEEE transactions on computers, vol. 31, no. 5, may 1988
4. A. L. Narasimha Reddy, P. Banerjee, "Algorithm based fault detection for signal processing application," IEEE transactions on computers, vol. 39, no. 10, October 1990
5. T. Hitana, Abhijit K. Deb, "Bridging Concurrent and Non-concurrent error Detection in FIR Filters," IEEE LECS Lab, Royal institute of technology, Electrum 229 S-16440 kista, Sweden, Nov 2004
6. G. L. Stüber, J. R. Barry, S. W. McLaughlin, Y. Li, M. A.Ingram, and T. G. Pratt, "Broadband MIMO-OFDM wireless communications," Proc. IEEE, vol. 92, no. 2, pp. 271–294, Feb. 2004
7. M. Nicolaidis, "Design for soft error mitigation," IEEE Trans. Device Mater. Rel., vol. 5, no. 3, pp. 405–418, Sep. 2005.
8. R. Baumann, "Soft errors in advanced computer systems," IEEE Des.Test Comput., vol. 22, no. 3, pp. 258–266, May/Jun. 2005.
9. 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
10. Mojtaba Valinataj, Saeed Safari, "Fault Tolerant Arithmetic Operations with Multiple Error Detection and Correction," 22nd IEEE International

Symposium on Defect and fault Tolerance in VLSI Systems, 2007
11. S. Pontarelliyz, G.C. Cardarilliy, "Totally Fault Tolerant RNS based FIR Filters", 14th IEEE International On-Line Testing Symposium 2008
12. E. P. Kim and N. R. Shanbhag, "Soft N-modular redundancy," IEEE Trans. Comput., vol. 61, no. 3, pp. 323–336, Mar. 2012.
13. Shubham C. Anjankar1, Dr. Mahesh T. Kolte, "Fault Tolerant and Correction System Using Triple Modular Redundancy," International Journal of Emerging
Engineering Research and Technology Volume 2, Issue 2, May 2014
14. Haryono, Jazi Eko Istiyanto, "Five Modular Redundancy with Mitigation Technique to Recover the Error Module," International Journal of advanced studies in Computer Science and Engineering IJASCSE, Volume 3, Issue 2, 2014
15. 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
16. K.Mohana Krishna1, Mrs.A.Maria Jossy, "Fault Tolerant Parallel Filters Based On Bch Codes," Int. Journal of Engineering and Research ISSN : 2248-9622, Vol. 5, Issue 4, (Part -7) April 2015, pp.99-103
17. M. Anogeeth1, Mr. Chenthi, "area efficient and fault tolerant parallel fir filter on ecc,"4th International Conference on science and management 15th May 2016
18. Zhen Gao, Pedro Reviriego, Zhan Xu, Xin Su," Fault Tolerant Parallel FFTs Using Error Correction Codes and Parseval Checks," IEEE Transactions On Very Large Scale Integration Systems, Vol. 24, No. 2, February 2016.

Cite this article as: Rinciya Beema, & Dr.Gnana Sheela K. (2018). Improved Fault Tolerance for Parallel FFT. Journal of Signal Processing, 4(3), 8–17.
http://doi.org/10.5281/zenodo.2222262