International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue-01
January 2018

Design and Analysis of Fault Tolerant Parallel FFTs Based on Error Correction Codes and Parseval Checks

K. Joshna, G. Anitha Chowdary, K.Sudha Rani
joshna.kande@gmail.com[1] , anitha.gadiparthi@gmail.com[2] , kaligithisudha@gmail.com
[1]Assistant Professor, Dept of ECE, TKR college of Engineering and Technology, Hyderabad, Telangana, India.

Abstract: *Algorithmic-Based Fault Tolerance (ABFT) technique is used to exploit the algorithmic properties to detect and correct errors. One example is Fast Fourier Transforms (FFTs) that is used in spectral analysis in the communication. There are several protection schemes to detect and correct errors in FFTs. The Parseval or sum of square check is the most widely known. It is most common to find several blocks operating in parallel in the modern communication systems. Recently, a method that exploits this truth to implement fault tolerance on parallel filters has been projected. During this a temporary system is first applied to guard FFTs. Then, second a improved protection schemes that combines the utilization of error correction codes and Parseval checks are projected and evaluated. This project outspread with FFT using Vedic multiplier. Simulation results are observed using Xilinx.*

Keywords- *Error correction codes, Fast Fourier transform, sum of square, Vedic multiplier.*

I.INTRODUCTION

Filters are typically used in electronic systems to emphasize signals within the bound frequency ranges and reject signals in alternative frequency ranges. In circuit theory, a filter is associated electrical network that alters the amplitude and section characteristics of a symbol with relevance frequency. Ideally, a filter neither adds new frequencies to the input signal, nor modifies the frequencies of that signal. However it will modify the relative amplitudes of the numerous frequency elements and their section relationships. Now-a-days filters are used in wide range of applications that support automotive, medical, and houses where reliableness of elements in digital electronic circuits is essential. Different types of filters are essential in the operation of most of the electronic circuits. There are totally different basis of classifying filters there is no straightforward hierarchal classification. The filtering techniques consider the issues caused due to change in behavioral properties of signal. Being specific with filter, the digital filters have large applications in digital signal processing. Filtering is additionally a category of signal process. The features of filters are to suppress completely or partially some facts of the signal. To possess the flexibility in electronic circuits we need to develop filter circuits capable of meeting a given set of specifications. In signal process, a digital filter is a device or process that removes some unwanted element or feature. Digital filters are used for two general purposes; separation of signals that are combined, and restoration of signals that are distorted. Most often, this suggests removing certain frequencies so as to suppress interfering signals and scale back original signal.

This parallel operation is exploited for fault tolerance. In fact, dependableness may be a major challenge for electronic system. Specifically, soft errors are a very important issue, and number of techniques is proposed over the years to mitigate them. A number of these techniques modifies the low-level type and implementation of the integrated circuits to stop soft errors from occurring. Different techniques work on the next abstraction level by adding redundancy which will detect and correct errors. The protection of digital filters has been studied widely. For instance, fault-tolerant implementations supported the utilization of residue variety systems or arithmetic codes. The utilization of reduced exactness replication or word-level protection has been additionally studied. Another choice to perform error correction is to use two completely different filter implementations in parallel. All those techniques concentrate on the protection of one filter.

Error coding is employed for fault tolerant computing in computer memory, magnetic and optical information storage media, satellite and part communications, cellular phone networks, and almost any other type of digital communication. Error coding uses mathematical formulas to code information bits at the source into longer bit words for transmission. The "code

word" will then be decoded at the destination to retrieve the information. The additional bits within the code word offer redundancy, according to the coding scheme used, which allow the destination to use the decoding method to determine if the communication medium introduces any errors and in some cases correct them so that the information need not be retransmitted. Totally different error coding schemes are chosen depending on the kinds of errors expected, whether or not information retransmission is feasible.

Faster processors and higher communications technology create a lot of complex coding schemes, with higher error detecting and correcting capabilities, attainable for smaller embedded systems, allowing for more robust communications. However, tradeoffs between bandwidth and coding overhead, coding complexity and allowable coding delay between transmissions, should be considered for every application.

A. Concept of Fault Tolerance

A number of techniques are used to defend a circuit from errors. Those range from modifications in the manufacturing process of the circuits to reduce the number of errors, to add redundancy at the logic or system level to ensure that errors do not affect the system functionality. Digital Filters are one of the most commonly used signal processing circuits and several techniques have been proposed to protect them from errors. There are number of methods used to identify and correct the faults within circuit. There are different fault tolerance approaches to conventional computational circuits and the DSP circuits. In some cases, the reliability of those systems is critical, and fault tolerant filter implementations are needed. Over the years, many techniques that exploit the filters structure and properties to achieve fault tolerance have been proposed. In all the techniques mentioned the protection of a single filter is considered.

Transient errors can often affect more than one bit producing multi-bit errors with a very high probability of error occurrence in adjacent memory cells. Bit interleaving is one technique to remedy multi-bit errors in adjacent memory cells as physically adjacent bits in memory array are assigned to different logical words. The single-error-correction, double-error-detection, and double-adjacent-error-correction (SEC-DED-DAEC) codes have previously been presented to correct adjacent double bit errors. The required number of check bits, the area and timing overheads for encoder and decoder of the SEC-DED-DAEC codes are similar to those of the SEC-DED codes. Consequently, adjacent double bit errors can be resolved with very little additional cost using the SEC-DED-DAEC codes.

The SEC-DED-DAEC codes may be an attractive alternative to bit interleaving in providing greater flexibility for optimizing the memory layout. Furthermore, the SEC-DED-DAEC code can be used in conjunction with bit interleaving and this method can efficiently deal with adjacent multi-bit errors. The FFTs in parallel increases the scope of applying error correction codes together. Generating parity together for parallel FFTs also helps in minimizing the complexity in some ECC. By assuming that there can only be a single error on the system in the case of radiation-induced soft errors and may be two in worst case. The proposed new technique is based on the combination of Partial Summation combined with parity FFT for multiple error correction.

II. LITERATURE SURVEY

[1] 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 applications, an interesting option is to use algorithmic-based fault tolerance (ABFT) techniques that try to exploit the algorithmic properties to detect and correct errors. Signal processing and communication applications are well suited for ABFT. 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, Parseval or sum of squares checks is the most widely known. In modern communication systems, it is increasingly common to find several blocks operating in parallel. Recently, a technique that exploits this fact to implement fault tolerance on parallel filters has been proposed. This technique is first applied to protect FFTs and to improve protection schemes that combine the use of error correction codes and Parseval checks are proposed and evaluated. The results show that

the proposed schemes can further reduce the implementation cost of protection.

[2] In this paper, fault tolerance system based on Error Correction Codes (ECCs) using Verilog is designed, implemented, and tested. It proposes that with the help of ECCs i.e. Error Correction Codes there will be more protected Parallel filter circuit. The filter they have used for error detection and correction are mainly finite-impulse response (FIR) filters. They have been used Hamming Codes for error correction in which they take a block of k bits and produces a block of n bits by adding n−k parity check bits. The parity check bits are XOR combinations of the k data bits. By properly designing those combinations it is possible to detect and correct errors. In this scheme they have been used redundant module in which the data and parity check bits are stored and can be recovered later even if there is an error in one of the bits. This is done by re - computing the parity check bits and comparing the results with the values stored. In this way using hamming codes error can be detected and corrected within the circuit.

[3] In this paper, Triple Modular Redundancy (TMR) and Hamming Codes have been used to protect different circuits against Single Event Upsets (SEUs). In this paper, the use of a Novel Hamming approach on FIR Filters is studied and implemented in order to provide low complexity, reduced delay and area efficient protection techniques for higher data bits. A novel Hamming code is proposed in this paper, to increase the efficiency of higher data bits. In this paper, the proposed technique is used to demonstrate, how lot the overhead due to interspersing the redundancy bits, their subsequent removal, pad to pad delay in the decoder and consumption of total area of FIR filter for higher bits are reduced. These are based on the novel hamming code implementation in the FIR filter instead of conventional hamming code used to protect FIR filter. In this scheme hamming code is used for transmission of 7-bit data word.

[4] In this paper, the design of a FIR filter with self checking capabilities based on the residue checking is analyzed. Usually the set of residues used to check the consistency of the results of the FIR filter are based on theoretic considerations about the dynamic range available with a chosen set of residues, the arithmetic characteristics of the errors caused by a fault and on the characteristic of the filter implementation. This analysis is often difficult to perform and to obtain acceptable fault coverage the set of chosen residues is overestimated. From obtained result this paper shows how an exhaustive fault injection campaigns allows to efficiently select the best set of residues. Experimental results coming from fault injection campaigns on a 16 taps FIR filter demonstrated that by observing the occurred errors and the detection modules corresponding to different residue has been possible to reduce the number of detection module, while paying a small reduction of the percentage of SEUs that can be detected. Binary logic dominates the hardware implementation of DSP systems.

[5] In this paper the proposed architecture is for the implementation of fault -tolerant computation within a high throughput multirate equalizer for an asymmetrical wireless LAN. The area overhead is minimized by exploiting the algebraic structure of the Modulus Replication Residue Number System (MRRNS). They had demonstrated that the area cost to correct a fault in a single computational channel is 82.7%. Fault tolerance within MRRNS architecture is implemented through the addition of redundant channels. This paper has presented a detailed analysis of the cost of implementing single fault correction capability in a FIR filter using the MRRNS. The fault-tolerant architecture makes use of the algebraic properties of the MRRNS, and has been shown to provide significant area savings when compared with general techniques. This architecture also requires few additional components to be designed, as identical redundant channels are used, and the polynomial mapping stages are simply expanded from the original components.

III. ERROR TOLERANT TECHNIQUES FOR PARALLEL FFTS

Error Correction based on Hamming Codes

The impulse response $h[n]$ completely defines a discrete time filter that performs the following operation on the incoming signal $x[n]$:

$$y[n] = \sum_{l=0}^{\infty} x[n-l].h[l] \cdots\cdots\cdots\cdots\cdots (1)$$

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue-01
January 2018

This property can be exploited in the case of parallel filters that operate on different incoming signals, as shown on Fig. 1. In this case, four filters with the same response process the incoming signals $x\,1[n]$, $x\,2[n]$, $x\,3[n]$, and $x\,4[n]$ to produce four outputs $y1[n]$, $y2[n]$, $y3[n]$, and $y4[n]$. To detect and correct errors, each filter can be viewed as a bit in an ECC, and redundant filters can be added to form parity check bits. This is also illustrated in Fig.1, where three redundant filters are used to form the parity check bits of a classical single error correction Hamming code. Those correspond to the outputs $z_1[N]$, $z_2[N]$, AND $z_3[N]$. Errors can be detected by checking if

$$z_1[n]=y_1[n]+y_2[n]+y_3[n]$$
$$z_2[n]=y_1[n]+y_2[n]+y_4[n]$$
$$z_3[n]=y_1[n]+y_3[n]+y_4[n]$$

When some of those checks fail, an error is detected. The error can be corrected based on which specific checks failed. For example, an error on filter $y1$ will cause errors on the checks of $z1$, $z2$, and $z3$.

TABLE I
ERROR LOCATION IN HAMMING CODE

$c_1c_2c_3$	Error Bit Position
000	No Error
111	y_1
110	y_2
101	y_3
011	y_4
100	z_1
010	z_2
001	z_3

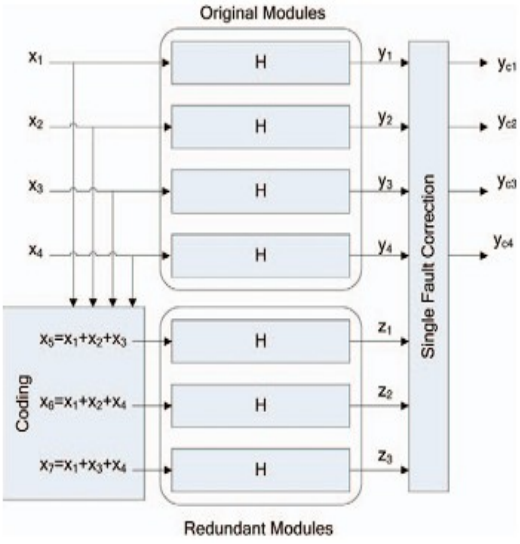


Fig. 1: ECC-based scheme for four filters and a Hamming code.

The proposed 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.

Fault tolerant FFT based on Parseval's check

Parseval's technique is one amongst the techniques to detect errors in parallel multiple FFT. This is often achieved with sum of Squares (SOSs) check [5] supported Parseval's theorem. The error free FFT should have its sum of Squares of the input equaling the total of Squares of its frequency domain output. This correlation usually establish errors with minimum overhead. For parallel FFTs, the Parseval's checks are often combined with the error correction codes to attenuate the realm overhead. Multiple error detection and correction is achieved through this method. One amongst the straightforward ways is to come up with the redundant input for single FFT with all the four FFT inputs. To correct error the parity FFT output is XORed with fault free outputs of the FFTs. Compared to the previous schemes bestowed within the Fault Tolerant Parallel FFTs victimization Error Correction Codes and Parseval Checks [1], this technique reduced the whole variety of sum of Squares used. Another existing work done is by combining SOS checks with hamming codes rather than exploitation Parseval's check on an individual as shown in Fig2.

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue-01
January 2018

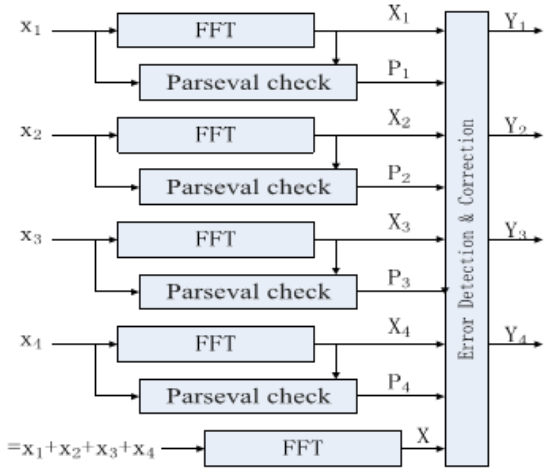


Fig. 2: Parity-SOS (first technique) fault-tolerant parallel FFTs.

This method combines the feature of parity calculation of hamming codes and error detection process of Sum of Squares. Concurrent Error Detection (CED) schemes for the FFT are the Sum of Squares (SOS) check based on Parseval theorem. The use of Parseval check is exponentially reduced to the direct comparisons of FFTs inputs and outputs used to protect parallel FFTs.

IV. PROPOSED PROTECTION SCHEMES FOR PARALLEL FFTS

The place to begin for our work is that the protection scheme based on the utilization of ECCs that was for digital filters. This scheme is shown in Fig1. In this example, a straightforward single error correction codes is employed. The initial system consists of 4 FFT modules and 3 redundant modules is value-added to sight and correct errors. The inputs to the 3 redundant modules area unit linear combinations of the inputs and that area unit used to check linear combinations of the outputs. For example, the input to the primary redundant module is

$$X_5=X_1+X_2+X_3$$

And since the DFT is a linear operation, its output Z_5 can be used to check that

$$Z_5=Z_1+Z_2+Z_3$$

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:

$$Z_{1C}[N]=Z_5[N]-Z_2[N]-Z_3[N]$$

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. 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.

In fig.2 the first proposed scheme is illustrated for the case of four parallel FFTs. A redundant 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 and correction using P1,P2,P3,P4 can be done by re-computing the FFT in error using the output of the parity FFT X and rest of the FFT outputs. For Example, if an error occurs in the first FFT, P1 will be set and the error can be corrected by

$$X_{1c}=X-X_2-X_3-X_4$$

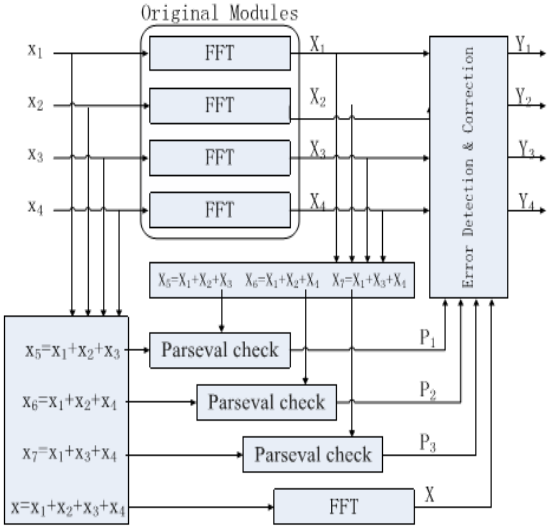


Fig. 3: Parity-SOS-ECC (second technique) fault-tolerant parallel FFTs.

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 in the parity-SOS scheme, an additional parity FFT is used to correct the errors. This second technique is shown in Fig. 3. The main benefit over the first parity SOS scheme is to reduce the

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue-01
January 2018

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).

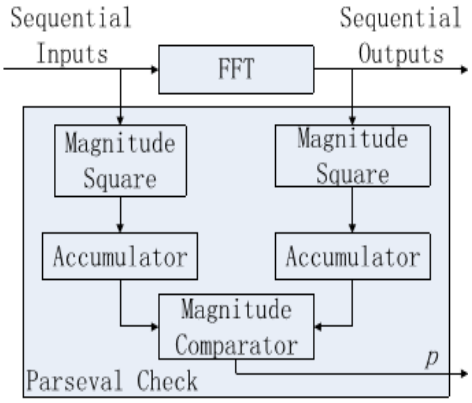


Fig. 4: Implementation of the SOS check

V. Vedic Sutra – Urdhwa Tiryakbhyam

In Fast Fourier Transform (FFT) we need to perform number of complex multiplications and complex additions. In this project a normal multiplier has been used for this multiplication which uses more computational time. In order to reduce the computational time and to increase the speed of operation we can replace this normal multiplier with the Vedic Multiplier. By doing this we can obtain less power consumption, high accuracy and reduced delay.

The sixteen sacred text Sutras apply to 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, half 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. Consider two 8-bit numbers, a (a8-a1) and b (b8-b1) where one to eight represents bits from the least significant bit to the most significant bit. The ultimate Product is represented by P (P16-P1). In Fig.5, the step by step methodology of multiplication of two 8-bit numbers using Urdhwa Tiryakbhyam sutra is illustrated. The number of bits are unit diagrammatic by dots and also the two dot approach 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 are a 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.

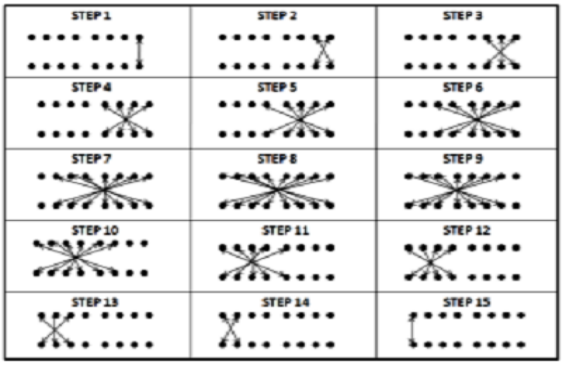


Fig.5: 8-bit binary multiplication using Urdhwa Tiryakbhyam Sutra

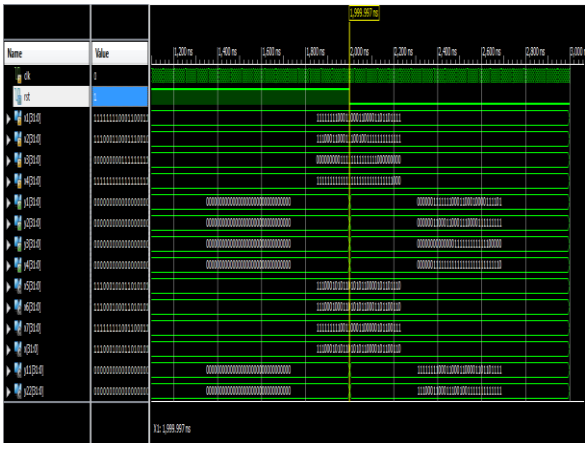
So, a large number of stages are required to get the final product. Higher order compressors discussed in next section can be employed to add more than 3 bits at a time (upto 7 bits) and hence can reduce the intermediate stages.

VI. RESULTS

The Modules are designed using VERILOG HDL and are successfully simulated, verified and synthesized using Xilinx ISE 13.2.

Proposed Result:

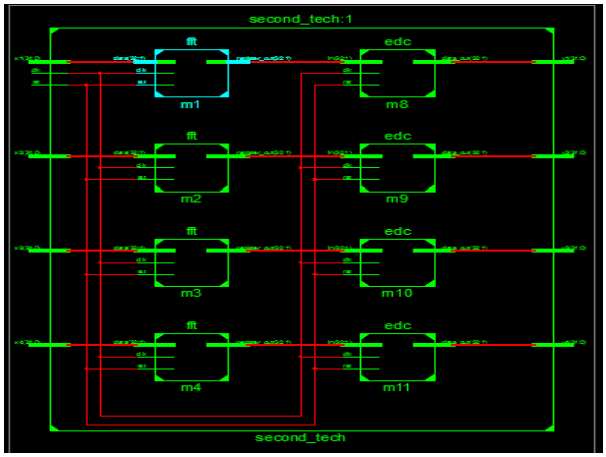
Simulation:



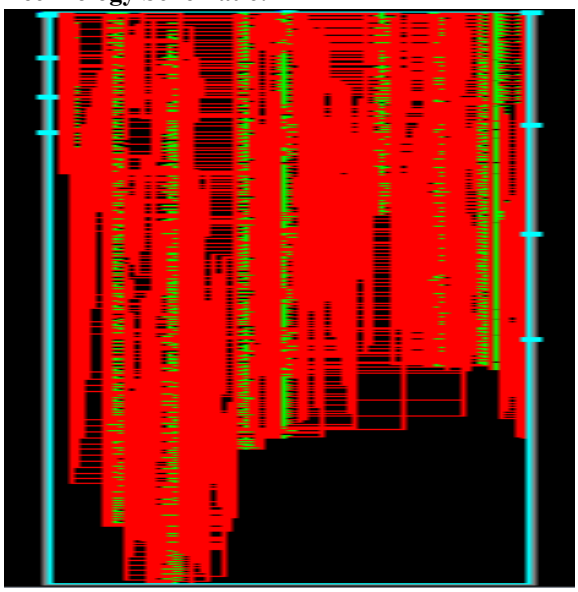
International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue-01
January 2018

Synthesis Results:

RTL Schematic:



Technology Schematic:



Design Summary:

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slices	266	4656	5%
Number of Slice Flip Flops	228	9312	2%
Number of 4 input LUTs	429	9312	4%
Number of bonded IOBs	254	232	109%
Number of GCLKs	1	24	4%

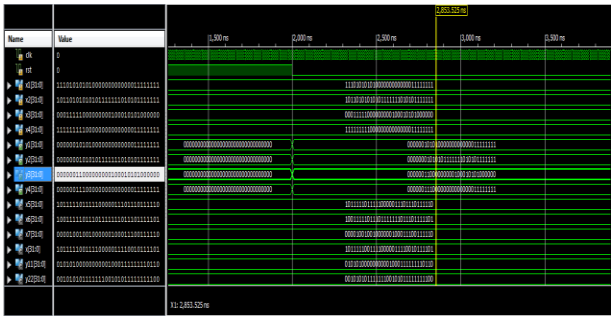
Timing Report:

```
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)
```

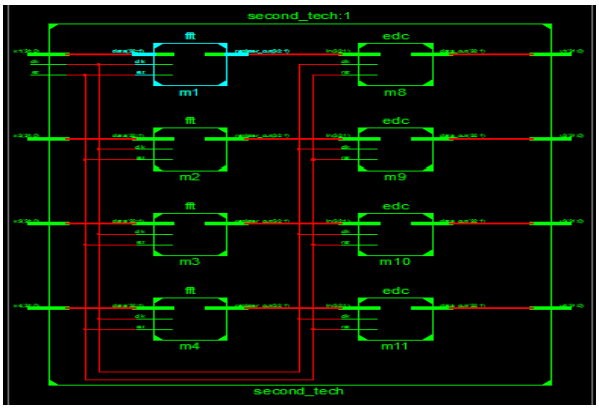
Extension Result:

Simulation:

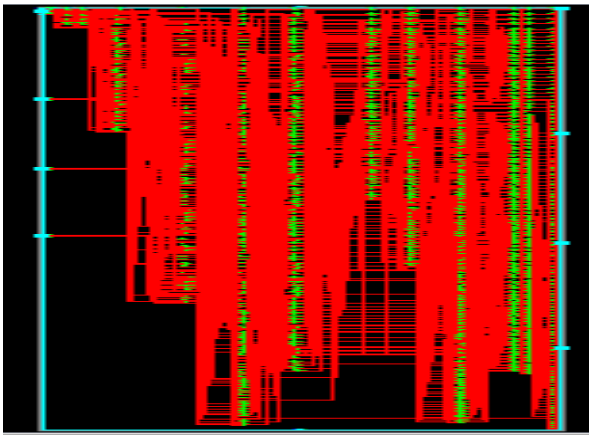


Synthesis Results:

RTL Schematic:



Technology Schematic:



International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue-01
January 2018

Design Summary:

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:

```
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)
```

VII. CONCLUSIONS

Detecting and correcting errors such as critical reliability are difficult in signal processing which increases the use of fault tolerant implementation. In modern signal processing circuits, it is common to find several filters operating in parallel. Proposed is an area efficient technique to detect and correct single errors. The Parseval or sum of square check is the most widely known. It is most common to find several blocks operating in parallel in the modern communication systems. 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, two improved protection schemes that mix the utilization of error correction codes and Parseval checks are projected and evaluated. This project outspread with FFT using Vedic multiplier. For the further improvement of the multiplier efficiency, we use Vedic multiplier i.e., Urdhwa Tiryakbhyam Sutra. By using this, we can improve the functionality of the magnitude square block in the Parseval check.

Future Scope: In Future, utilization of DCT instead of FFT will be carried out. Since SOS-ECC technique can detect and correct only single bit fault, this will be extended to multi bit faults by using the trellis code and hence area will be further reduced.

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.

[2] R. Baumann. 2005. Soft errors in advanced computer systems. IEEE Des. Test Comput. 22(3): 258-266.

[3] M. Ergen. 2009. Mobile Broadband-Including WiMAX and LTE. New York, NY, USA: Springer- Verlag.

[4] Z. Gao *et al*. 2015. Fault tolerant parallel filters based on error correction codes. IEEE Trans. Very Large Scale Integr. (VLSI) Syst. 23(2): 384-387.

[5] R. W. Hamming. 1950. Error detecting and error correcting codes. Bell Syst. Tech. J. 29(2): 147-160.

[6] T. Hitana and A. K. Deb. 2004. Bridging concurrent and non-concurrent error detection in FIR filters. in Proc. Norchip Conf. pp. 75-78.

[7] J. Y. Jou and J. A. Abraham. 1988. Fault-tolerant FFT networks. IEEE Trans. Comput. 37(5): 548-561.

[8] N. Kanekawa, E. H. Ibe, T. Suga and Y. Uematsu. 2010. Dependability in Electronic Systems: Mitigation ofHardware Failures, Soft Errors, and Electro-Magnetic Disturbances. New York, NY, USA: Springer-Verlag.

[9] E. P. Kim and N. R. Shanbhag. 2012. Soft N-modular redundancy. IEEE Trans. Comput. 61(3): 323-336.

[10] M. Nicolaidis. 2005. Design for soft error mitigation. IEEE Trans. Device Mater. Rel. 5(3): 405-418.

[11] S. Pontarelli, G. C. Cardarilli, M. Re and A. Salsano. 2008. Totally fault tolerant RNS based FIR filters. In Proc. 14th IEEE Int. On-Line TestSymp. (IOLTS). pp. 192-194.

[12] A. L. N. Reddy and P. Banerjee. 1990. Algorithm based fault detection for signal processing applications. IEEE Trans. Comput. 39(10): 1304- 1308.

[13] P. Reviriego, C. J. Bleakley and J. A. Maestro. 2012. A novel concurrent error detection technique for the fast

Fourier transform. In Proc. ISSC, Maynooth, Ireland. pp. 1-5.

[14] P. Reviriego, S. Pontarelli, C. J. Bleakley and J. A.Maestro. 2012. Area efficient concurrent error detection and correction for parallel filters. IET Electron. Lett. 48(20): 1258-1260.

[15]S. Sesia, I. Toufik, and M. Baker. 2011. LTE-The UMTS Long Term Evolution: From Theory to Practice, 2nd ed. New York, NY, USA: Wiley. [16]B. Shim and N. 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, April.

[17]A. Sibille, C. Oestges and A. Zanella. 2010. MIMO: From Theory to Implementation. San Francisco, CA, USA: Academic.

[18]G. L. Stüber, J. R. Barry, S. W. McLaughlin, Y. Li, M. A. Ingram and T. G. Pratt. 2004. Broadband MIMO-OFDM wireless communications. Proc. IEEE. 92(2): 271-294.

[19]P. P. Vaidyanathanm. 1994. Multirate Systems and Filter Banks. Englewood Cliffs, NJ, USA: PrenticeHall, 1993. S.-J. Wang and N. K. Jha. Algorithmbased fault tolerance for FFT networks. IEEE Trans. Comput., vol. 43, no. 7, pp. 849–854.

[20]S.-J. Wang and N. K. Jha. 1994. Algorithm-based fault tolerance for FFT networks. IEEE Trans. Comput. 43(7): 849-854.

[21]Richard M. Jiang. 2007. An Area-Efficient FFT Architecture for OFDM Digital Video Broadcasting. IEEE Transactions on Consumer Electronics. 53(4).