Advances in Computational Sciences and Technology
ISSN 0973-6107 Volume 11, Number 7 (2018) pp. 597-605
© Research India Publications
http://www.ripublication.com

Fault Tolerant Parallel Filters Based on ECC Codes

K.Pravalika[1] and Ch. Rajendra Prasad[2]

*[1]PG Scholar, Department of Electronics and communication engineering,
S R Engineering College, Ananthasagar, Warangal, Telangana, India.*

*[2]Assistant Professor, Department of ECE, S R Engineering College, Warangal,
Ananthasagar, Telangana, India.*

ABSTRACT

With changes in digital technology, systems are developed with more complex designs. The reliability of such systems has decreased when compared to the simple design systems. Several techniques have been proposed to maintain the reliability of the complex design systems. All these techniques are not efficient in providing a fault tolerant system due to more area and power consumption. In this project, one of the Error Correction Code (ECC) techniques is presented, which is the Hamming code concept. The Hamming code is used to detect and correct errors. This type of technique uses the concept of placing filters with the same response in parallel in large numbers so that the design can maintain more efficient fault tolerance. Check filters are used for error detection and correction in the proposed technique. In this brief, that idea is generalized to show that parallel filters can be protected using error correction codes (ECCs) in which each filter is the equivalent of a bit in a traditional ECC. This new scheme allows more efficient protection when the number of parallel filters is large. Parseval or Sum of squares check is one of the protection schemes which is widely used. It is common to find various blocks in parallel. Newly, a technique that utilizes fact to implement Fault Tolerance on parallel filters has been proposed. This technique is first applied for protecting FFT. Hence, two improved protection schemes are proposed and evaluated. These schemes combine the use of error correction codes and parseval checks. Further, these proposed schemes are reducing the implementation cost of production.

Index Terms: Error correction codes (ECCs), fast Fourier transforms (FFTs), soft errors.

I. INTRODUCTION

Finite Impulse Response (FIR) filters are one of two primary types of digital filters used in digital signal processing (DSP) applications, the other type being Infinite Impulse Response (IIR). High performance FIR filters have applications in several video processing and digital communications systems. In some applications, the FIR filter circuit must be able to operate at high sample rates, while in other applications, the FIR filter circuit must be a low-power circuit operating at moderate sample rates. The real time realization of FIR filter with less hardware requirement and less latency has become more and more important. Because the complexity of implementation grows with the length of filter, several algorithms have been made to develop effective architectures for realization of FIR filters in application specific integrated circuits (ASIC) and field programmable gate arrays (FPGA) platforms and one of them is error robust design.

Moreover, this technology represents a number of attractive features such as simplicity, regularity and modularity of architecture. In recent years, fault tolerant-based FIR filter has gained substantial popularity as a primary DSP operation and are rapidly replacing classic analog filters.

II. PARLLEL FILTERS

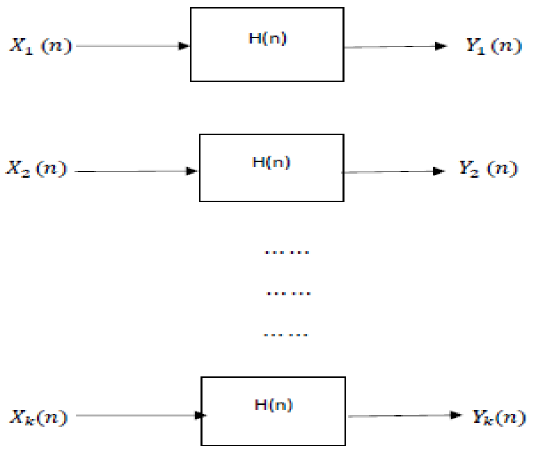


Figure 1: Parallel filters

The filters for certain number of samples is described by the equation

$$y(n) = \Sigma\, x(n - k).\, h(k) \quad [1]$$

In any FIR filter convolution is carried out by y[n] = Σ x [n − l] · h [l]. Where x[n] is the input signal, y[n] is the output, and h[l] is the impulse response of the filter. When the response h[l] is nonzero, only for a finite number of samples, the filter is known as a FIR filter, otherwise the filter is an infinite impulse response (IIR) filter. There are several structures to implement both FIR and IIR filters. This kind of filter is found in some communication systems that use several channels in parallel. In data acquisition and processing applications is also common to filter several signals with the same response. The data and parity check bits are stored and can be recovered later even if there is an error in one of the bits.

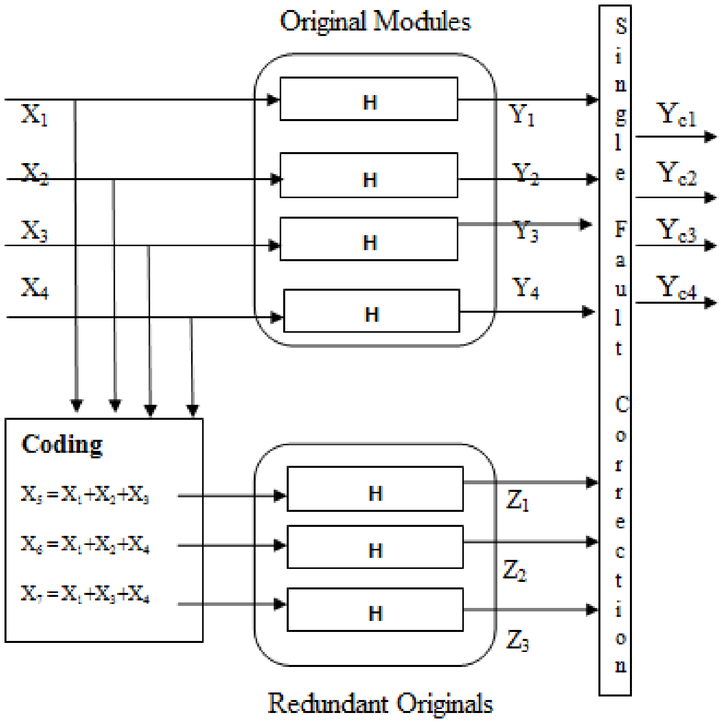


Figure 2: Fault tolerant FIR filter based on hamming code

III. ERROR CORRECTION

Error detection and correction or error controls are techniques that enable reliable delivery of digital data over unreliable communication channels. Many communication channels are subject to channel noise, and thus errors may be introduced during transmission from the source to a receiver. Error detection techniques allow detecting such errors, while error correction enables reconstruction of the original data in many cases.

1) Error detection is the detection of errors caused by noise or other impairments during transmission from the transmitter to the receiver.

2) Error correction is the detection of errors and reconstruction of the original, error-free data.

Convolution codes are important because they are commonly used both as channel coding technique and as building blocks in other techniques as turbo codes and LDPC codes. For each of n input bits to the encoder the output is m bits where $m \geq n$ ((m,n) code) and the code ratio is $R = n/m$, generally a lower code ratio R (more redundancy) increases the error correction/detection possibility. A parity check matrix H with dimensions $(m - n) \times m$ is then used to check if the incoming codeword is valid. A tanner graph can be generated using the parity check matrix H of a code, where the nodes to the left correspond to codeword bits and the right nodes corresponds to check equations.

For example, Hamming code corresponds to check node,

The check equation is

$$z1\,[n] = y1\,[n] + y2\,[n] + y3\,[n] \qquad [2]$$

$$z2\,[n] = y1\,[n] + y2\,[n] + y4\,[n] \qquad [3]$$

$$z3\,[n] = y1\,[n] + y3\,[n] + y4\,[n] \qquad [4]$$

For example, an error on filter y1 will cause errors on the checks of 2, 3, and 4. Similarly, errors on the other filters will cause errors on a different group.

IV. PROPOSED SYSTEM

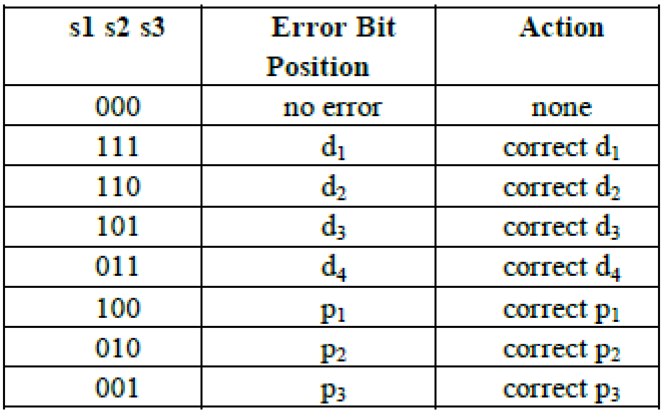
A. Error Correction based on Hamming Codes: The aim of error tolerant design is to protect parallel FFTs from errors. Various schemes have been proposed for error detection and correction in FFTs. One of the basic and simple methods is error correction using hamming codes. Unlike parity code which can detect only odd bit error, the hamming code can detect two bit errors and correct one error. The new technique is based on the use of the ECCs. A simple ECC takes 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. As an example, let us consider a simple Hamming code with k = 4 and n = 7. In this case, the three parity check bits p1, p2, p3 are computed as a function of the data bits d1, d2, d3, d4 as follows: p1 = d1 © d2 ©d3 p2 = d1 ©d2 © d4 p3 = d1 © d3 ©d4. ©- convolution 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 the example considered, an error on d1 will cause errors on the three parity checks; an error on d2 only in p1 and p2; an error on d3 in p1 and p3; and finally an error on d4 in p2 and p3. Therefore, the data bit in error can be located and the error can be corrected. This is commonly formulated in terms of the generating G and parity check H matrixes. For the Hamming code considered in the example.

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

$$H = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}$$

Encoding is done by computing y = x • G and error detection is done by computing s=y•HT, where the operator „•" is based on module two addition (XOR) and multiplication. Correction is done using the vector s, known as syndrome, to identify the bit in error**.**

Table 1: Error position of Hamming codes



s1 s2 s3	Error Bit Position	Action
000	no error	none
111	d_1	correct d_1
110	d_2	correct d_2
101	d_3	correct d_3
011	d_4	correct d_4
100	p_1	correct p_1
010	p_2	correct p_2
001	p_3	correct p_3

Similar to other error correction codes, hamming codes also utilizes the parity bit which is generated for the corresponding input sequence for detecting errors. It achieves higher code rate with minimum distance of three. The number of parity bits depends on the total number of data bits.

B. Fault tolerant FFT based on Parseval's Check: Parseval's method is one of the techniques to detect errors parallel in multiple FFT. This is achieved with Sum of Squares (SOSs) check based on Parseval's theorem. The error free FFT should have its Sum of Squares of the input equaling the Sum of Squares of its frequency domain output. Correlation can be used to identify errors with minimum overhead. For parallel FFTs, the Parseval's check can be combined with the error correction codes to minimize the area overhead. Multiple error detection and correction is achieved through this combination. One of the easy ways is to generate 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 presented in the Fault Tolerant Parallel FFTs Using Error Correction Codes and Parseval Checks, this technique reduced the total number of Sum of Squares used. Another existing work done is by combining SOS checks with hamming codes instead of using Parseval's check individually as shown in Figure 3.

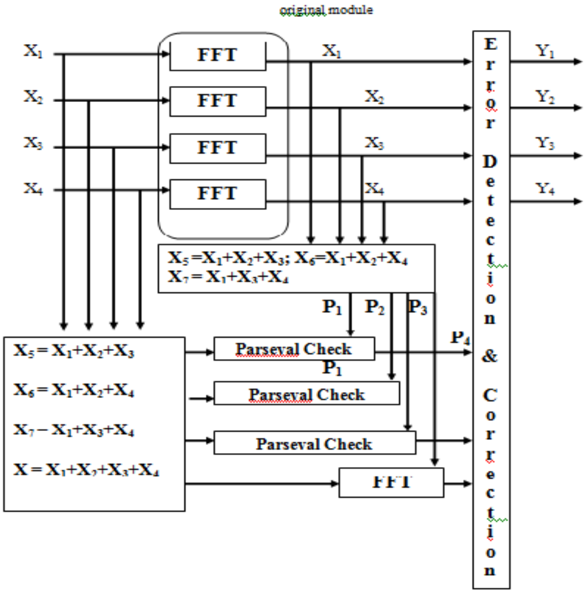


Figure 2: Parity-SOS-ECC fault-tolerant parallel FFTs

Sum of Squares is thus eliminated using only the adder blocks. Technique 1 also uses 3 parallel redundant.

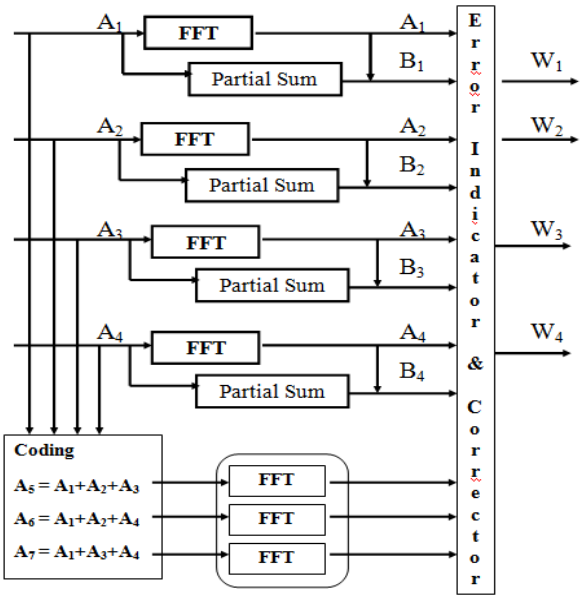


Figure 3: Parity-PS-ECC fault-tolerant parallel FFTs

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 Pa theorem. The use of parseval check is exponentially reduced to the direct comparisons of FFTs inputs and outputs used to protect parallel FFTs. The technique analyzed in the previous work has certain limitation due to the complexity of handling larger number of FFTs and Sum of Squares block. Instead of using Sum of Squares, Partial Summation (PS) is used for calculating its parity at the input and the output side of the FFT. It sums all possible node values of 4-point FFT along with the twiddle factors. Multiplication operation which leads to the complexity in Figure 3. illustrates the Parity Partial Summation block for less error prone applications. For example when the error occurs in A1 and A2 then it can be detected by the partial summation used individually for FFT blocks. The first check equation is selected in such a way that the both error block signals are not present.

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 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 or to the SOS checks. The triplication of these blocks has a small impact on circuit complexity as they are much simpler 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 guarantee 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.

V. RESULTS

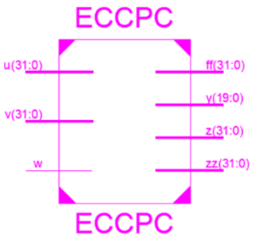
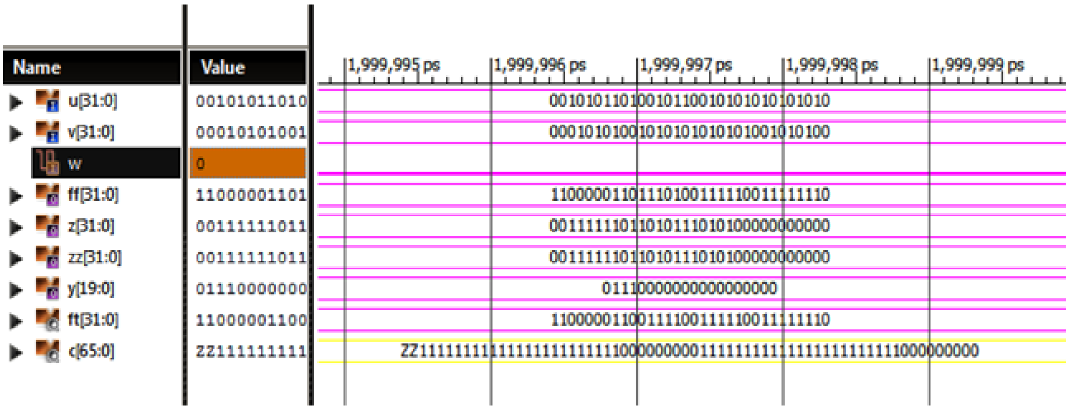


Figure 5: RTL Schematic



```
Name       Value
u[31:0]    00101011010      00101011010010110010101010101010
v[31:0]    00010101001      00010101001010101010101001010100
w          0
ff[31:0]   11000001101      11000001101110100111110011111110
z[31:0]    00111111011      00111111011010111010100000000000
zz[31:0]   00111111011      00111111011010111010100000000000
y[19:0]    01110000000      01110000000000000000
ft[31:0]   11000001100      11000001100111100111110011111110
c[65:0]    ZZ111111111      ZZ111111111111111111111111110000000001111111111111111111111111000000000
```

Figure 6: Output

VI. CONCLUSION

A new method to implement fault-tolerant parallel filters has been presented in this brief. The proposed scheme exploits the linearity of filters to implement an error correction mechanism. 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 approach is based on applying SOS-ECC check to the parallel FFT outputs to detect and correct errors. 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 a parsvel’s check per FFT or alternatively using a set of SOS checks that form an ECC. This technique can detect and correct only single bit error and it reduces area results in high speed compared to existing techniques.

REFERENCES

[1] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*, Englewood Cliffs, N.J., USA: Prentice Hall, 1993.

[2] A. Sibille, C. Oestges and A. Zanella, *MIMO: From Theory to Implementation*, New York, NY, USA: Academic, 2010.

[3] 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.

[4] M. Nicolaidis, “Design for soft error mitigation,” *IEEE Trans. DeviceMater. Rel.*, vol. 5, no. 3, pp. 405–418, Sep. 2005.

[5] C. L. Chen and M. Y. Hsiao, “Error-correcting codes for semiconductormemory applications: A state-of-the-art review,” *IBM J. Res. Develop.*,vol. 28, no. 2, pp. 124–134, Mar. 1984.

[6] A. Reddy and P. Banarjee "Algorithm-based fault detection for signal processing applications," *IEEE Trans. Comput.*, vol. 39, no. 10, pp. 1304–1308, Oct. 1990.

[7] S. Pontarelli, G. C. Cardarilli, M. Re, and A. Salsano, "Totally fault tolerant RNS based FIR filters," in *Proc. IEEE IOLTS*, 2008, pp. 192–194.

[8] Z. Gao, W. Yang, X. Chen, M. Zhao and J. Wang, "Fault missing rate analysis of the arithmetic residue codes based fault-tolerant FIR filter design," in *Proc. IEEE IOLTS*, 2012, pp. 130–133.

[9] B. Shim and N. Shanbhag, "Energy-efficient soft error-tolerant digital signal processing," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 14, no. 4, pp. 336–348, Apr. 2006.

[10] Y.-H. Huang, "High-efficiency soft-error-tolerant digital signal processing using fine-grain subword-detection processing," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 18, no 2, pp. 291–304, Feb. 2010.

[11] P. Reviriego, C. J. Bleakley, and J. A. Maestro, "Structural DMR: A technique for implementation of soft-error-tolerant FIR filters," *IEEE Trans. Circuits Syst. II: Exp. Briefs*, vol. 58, no. 8, pp. 512–516, Aug. 2011.

[12] P. Reviriego, S. Pontarelli, C. 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. Syst.*, vol. 23, no. 2, pp. 384–387, Feb. 2015.

[14] R. W. Hamming, "Error correcting and error detecting codes," *Bell Sys. Tech. J.*, vol. 29, pp. 147–160, Apr. 1950.

[15] A. Chatterjee, andM. A. d'Abreu, "The design of fault-tolerant linear digital state variable systems: Theory and techniques," *IEEE Trans. Comput.*, vol. 42, no. 7, pp. 794–808, Jul. 1993.