International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 04
February 2018

Parallel Filters Using Error Correction Codes in Modern Signal Processing Circuits

K. Surya Mahesh[1] , Dr. J. Sofia Priyadharshini[2] , Dr.M.V.Subramanyam[3]
suryamahesh009@gmail.com[1], jspd1810@gmail.com[2], mvsraj@yahoo.com[3]
[1]PG Scholar, VLSI, Rajeev Gandhi Memorial College of Engineering & Technology, Nandyal, Kurnool, Andhra Pradesh.
[2]Associate Professor, Dept of ECE, Rajeev Gandhi Memorial College of Engineering & Technology, Nandyal, Kurnool, Andhra Pradesh.
[3]Professor & Principal SREC, Nandyal, Kurnool, A.P

Abstract: *As the complexity of communications and signal processing systems increases, so does the number of blocks or elements that they have. In many cases, some of those elements operate in parallel, performing the same processing on different signals. A typical example of those elements are digital filters. The increase in complexity also poses reliability challenges and creates the need for fault-tolerant implementations. A scheme based on error correction coding has been recently proposed to protect parallel filters. In that scheme, each filter is treated as a bit, and redundant filters that act as parity check bits are introduced to detect and correct errors. In this brief, the idea of applying coding techniques to protect parallel filters is addressed in a more general way. In particular, it is shown that the fact that filter inputs and outputs are not bits but numbers enables a more efficient protection. This reduces the protection overhead and makes the number of redundant filters independent of the number of parallel filters. The proposed scheme is first described and then illustrated with two case studies. Finally, both the effectiveness in protecting against errors and the cost are evaluated for a field-programmable gate array implementation.*

Keywords— Error correction codes(ECCs),filters, soft errors

I. Introduction

Filters are often used in electronic systems to emphasize signals in certain frequency ranges and reject signals in other frequency ranges. In circuit theory, a filter is an electrical network that alters the amplitude and/or phase characteristics of a signal with respect to frequency. Ideally, a filter will not add new frequencies to the input signal, nor will it change the component frequencies of that signal, but it will change the relative amplitudes of the various frequency components and/or their phase relationships. Today filters are widely used in number of applications which based on automotive, medical, and space where reliability of components in digital electronic circuits is critical. Filters of some sort are essential in the operation of most electronic circuits. There are many different bases of classifying filters and these overlap in many different ways; there is no simple hierarchical classification. As the behavioral properties of signal changes the techniques of filtering it will be differ. Being specific with filter, the digital filters have vast applications in digital signal processing. Filtering is also a class of signal processing, the defining feature of filters being the complete or partial suppression of some aspect of the signal. It is therefore in the interest of anyone involved in electronic circuit design to have the ability to develop filter circuits capable of meeting a given set of

specifications. In signal processing, a digital filter is a device or process that removes some unwanted component or feature from a signal. Digital filters are used for two general purposes; separation of signals that have been combined, and restoration of signals that have been distorted in some way. Most often, this means removing some frequencies and not others in order to suppress interfering signals and reduce background noise.

Parallel Processing And communication systems. In many cases, Parallel filters are commonly found in modern signal the filters perform the same processing on different incoming signals as there is a tendency to use multiple-input–multiple output systems.

This parallel operation can be exploited for fault tolerance. In fact, reliability is a major challenge for electronic system. In particular, soft errors are an important issue, and many techniques have been proposed over the years to mitigate them. Some of these techniques modify the low-level design and implementation of the integrated circuits to prevent soft errors from occurring. Other techniques work at a higher abstraction level by adding redundancy that can detect and correct errors.

The protection of digital filters has been widely studied. For example, fault-tolerant implementations based on the use of residue number systems or arithmetic codes have been proposed. The use of reduced precision replication or word-level protection has been also studied Another option to perform error correction is to use two different filter implementations in parallel. All those techniques focus on the protection of a single filter.

This brief studies the protection of parallel filters using more general coding techniques. In particular, a key difference with ECCs is that both filter inputs and outputs are numbers. Therefore, not only a zero or a one can be used for the coding (as done with ECCs). This can be exploited, as shown in the rest of this brief, to provide error correction by adding only two redundant filters regardless of the number of parallel filters. The reduced number of redundant filters does not affect the ability of the scheme to correct errors but reduces the implementation cost. In the rest of this brief, first, the parallel filters and the existing ECC-based protection scheme are described. Then, the proposed coding scheme is presented and illustrated with a few practical case studies. Finally, the case studies are evaluated for a field- programmable gate array (FPGA) implementation and compared with the previously proposed ECC-based technique.

Concept of Fault Tolerance

A number of techniques can be used to protect a circuit from errors. Those range from modifications in the manufacturing process of the circuits to reduce the number of errors to adding 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 faults and the actions necessary to correct the faults within circuit. Digital filters are widely used in signal processing and communication systems. 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 so far, the protection of a single filter is considered.

II. Literature Review

In this paper, fault tolerance based system based on Error Correction Codes (ECCs) using VHDL 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 has been possible. The filter they have used for error detection and correction are mainly finite-impulse response (FIR) filters. They

have been used Hamming Codes for fault correction in which they 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. In this scheme they have used redundant module in which the data and parity check bits are store d 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. [1]

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, reduce delay and area efficient protection techniques for higher bits data. A novel Hamming code is proposed in this paper, to increase the efficiency of higher data bits. In this paper, they have proposed technique used to demonstrate, how the lot of 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 used for transmission of 7-bit data item. [2]

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 of 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. Obtained result and therefore requires that Instead, in this paper they have showed how using 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. [3]

In this paper they have proposed an architecture 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 for our system 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. ECC-Based Protection Of Parallel Filters

The impulse response $h[n]$ completely defines a discrete time filter that performs the following operation on the incoming signal $x[n]$: $$y[n]=\sum_{l=0}^{\infty} x[n-l].h[l]. \quad (1)$$

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 y 1[n], y 2[n], y 3[n], and y 4[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 [14]. Those correspond to the outputs z1[n], z2[n], and z3[n]. Errors can be detected by checking if

$$z_1[n] = y_1[n]+y_2[n]+ y_3[n] \quad (2)$$

$$z_2[n] =y_1[n]+y_2[n]+y_4[n] \quad (3)$$

$$z_3[n] = y_1[n]+ y_3 [n]+ y_4[n] \quad (4)$$

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 y 1 will cause errors on the checks of z1, z2, and z3

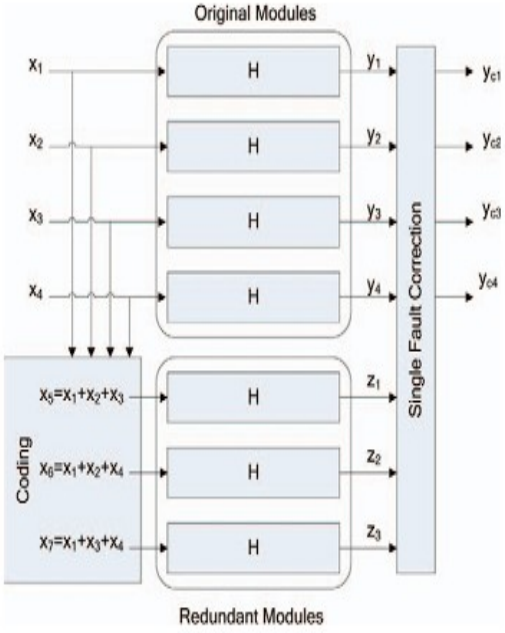


Fig. 1. ECC-based scheme for four filters and a Hamming code.

IV. Coding For Fault-Tolerant Parallel Filters

The proposed scheme is illustrated in Fig. 2 for the case of four parallel filters. The input signals are encoded using a matrix with arbitrary coefficients to produce the signals that enter the four original and two redundant filters. In its more general form, this coding matrix A can be formulated as

$$A = \begin{pmatrix} a11 & a12 & a13 & a14 \\ a21 & a22 & a23 & a24 \\ a31 & a32 & a33 & a34 \\ a41 & a42 & a43 & a44 \\ a51 & a52 & a53 & a54 \\ a61 & a62 & a63 & a64 \end{pmatrix}$$

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 04
February 2018

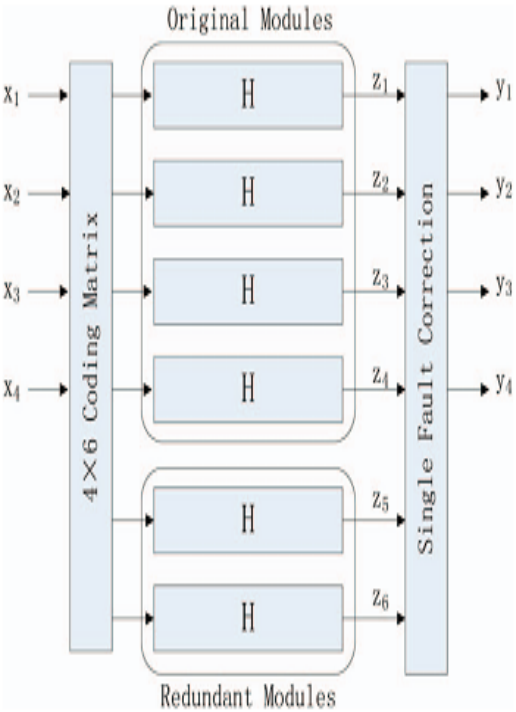


Fig. 2. Proposed coding scheme in general form

With this coding scheme, the outputs of the filters, i.e., y1[n], y2[n], y3[n], and y4[n], can be obtained as follows:

$$\begin{pmatrix} y1 \\ y2 \\ y3 \\ y4 \end{pmatrix} 1235 = (A1235)^{-1} \begin{pmatrix} z1 \\ z2 \\ z3 \\ z5 \end{pmatrix}$$

where $A1235$ is a submatrix of A, including the first, second, third, and fifth rows. This process can be repeated with different submatrixes of A, for example, with $A1236$, $A2345$, and $A2346$. In the error-free case, all the recovered versions of $y1[n]$, $y2[n]$, $y3[n]$, and $y4[n]$ will be the same. When there are differences, an error is detected. For example, suppose that

$$\begin{pmatrix} y1 \\ y2 \\ y3 \\ y4 \end{pmatrix} 1235 \neq \begin{pmatrix} y1 \\ y2 \\ y3 \\ y4 \end{pmatrix} 1236 \quad \begin{pmatrix} y1 \\ y2 \\ y3 \\ y4 \end{pmatrix} 2345 = \begin{pmatrix} y1 \\ y2 \\ y3 \\ y4 \end{pmatrix} 2346$$

which means that there is an error among filters { 1 2 3 5 6} and that filters {2 3 4 5 6} are correct. therefore, the faulty filter is filter 1. Then, the error can be corrected by taking the final outputs from a set that does not include filter 1.

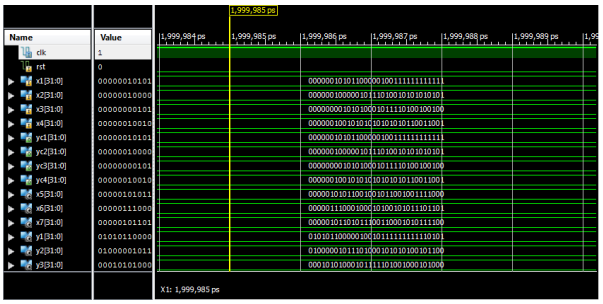
The error correction and detection logic can be simplified assuming that there is only a single error. In that case, checking only that, for each recovered set, the sums of the values $y1[n] + y2[n] + y3[n] + y4[n]$ are equal is enough. In more detail, four checks are needed, each involving five filters and excluding one. For example, if branch 1 is excluded, the error checking would be

$$\begin{cases} s_1^1 = \overline{w}\ 2345\,(z2\ z3\ z4\ z5)^T \\ s_1^2 = \overline{w}\ 2346\,(z2\ z3\ z4\ z6)^T \\ e1 = s_1^1 - s_1^2 \end{cases}$$

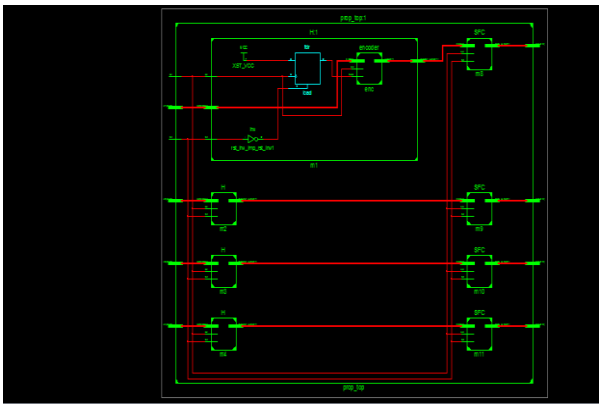
in which $w\bar{}2345 = [1111](A2345) - 1$, and $w\bar{}2346 = [1111]\,(A2346) - 1$.

V. RESULTS

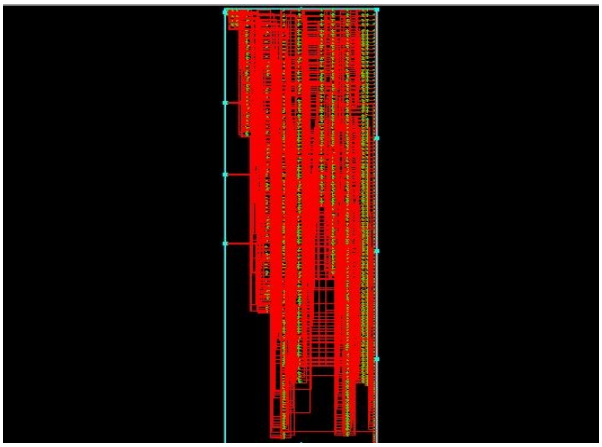
Simulation.



RTL Schematic.



Technology Schematic.



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	190	123%
Number of GCLKs	1	24	4%

Timing Report.

```
Delay:               7.209ns (Levels of Logic = 6)
  Source:            m4/enc/out_24 (FF)
  Destination:       m11/data_21 (FF)
  Source Clock:      clk rising
  Destination Clock: clk rising

  Data Path: m4/enc/out_24 to m11/data_21
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     FD:C->Q              2    0.514   0.532  m4/enc/out_24 (m4/enc/out_24)
     LUT4:I0->O           2    0.612   0.449  m11/Mxor_old_p8_10_xor0000_xo<0>1 (m11/old_p8_10_
     LUT2:I1->O           1    0.612   0.360  m11/Mxor__xor0001_xo<0>11 (m11/Mxor__xor0001_xo<0
     LUT4_D:I3->O        22    0.612   1.058  m11/Mxor__xor0001_xo<0>14 (m11/_xor0001)
     LUT2:I1->O           5    0.612   0.690  m11/data_10_mux0000111 (m11/N26)
     LUT4:I0->O           1    0.612   0.000  m11/data_21_mux0000_G (N479)
     MUXF5:I1->O          1    0.278   0.000  m11/data_21_mux0000 (m11/data_21_mux0000)
     FDR:D                     0.268          m11/data_21
    ----------------------------------------
    Total                      7.209ns (4.120ns logic, 3.089ns route)
                               (57.1% logic, 42.9% route)
```

Comparison.

parameters	Existing method	Proposed method
time	Total:18.034ns (12.237ns logic,5.797ns route)	Total:17.281ns (11.903ns logic, 5.378ns route)
LUT	790	421
Flip flops	249	229

family	**Spartan3E**	**Spartan3E**

VI. Conclusion

In this paper a new scheme to protect parallel filters that are commonly found in modern signal processing circuits has presented. The approach is based on applying ECCs to the parallel filters outputs to detect and correct errors. The scheme can be used for parallel filters that have the same response and process different input signals. An objectives has also been discussed to show the effectiveness of the scheme in terms of error correction and problem definition also shows the overheads. The proposed scheme can also make system cost lower. Proposed work will result in more efficient fault tolerant system using parallel IIR filters based on ECCs, which will meet the goal to achieve low power consumption, increase area of application and high speed.

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 ElectroMagnetic Disturbances, New York, NY, USA: Springer Verlag, 2010.

[4] M. Nicolaidis, "Design for soft error mitigation," IEEE Trans. Device Mater. Rel., vol. 5, no. 3, pp. 405–418, Sep. 2005.

[5] C. L. Chen and M. Y. Hsiao, "Error-correcting codes for semiconductor memory 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] T. Hitana and A. K. Deb, "Bridging concurrent and non-concurrent error detection in FIR filters," in Proc.
Norchip Conf., 2004, pp. 75–78.

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

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