# TO DESIGN FAULT TOLERANT AND LOW COST PARALLEL FIR FILTERS USING ERROR CORRECTION CODES

Rakesh Huidrom[1] and Dr. R. Nagaraj Ramrao[2]

Department of Electronics & Communication Engineering, The Oxford College of Engineering, Bangalore, India[1]

Director &Professor, Department of Electronics & CommunicationEngineering, The Oxford College of Engineering, Bangalore, India[2]

Email: rakeshhuidrom79@gmail.com[1]

Abstract: Filters are widely used in dealing out with signal processing and communication systems. The filters so used are digital filters. In those systems, assurances to efficient operation of signal are insignificant and that is why implementation of fault tolerant filters are needed. Enhancing technology make system more complex that include many filters. In those complex systems, it is frequent to have number of filters that functions in parallel. In parallel combination of filters, there apply the same filter to different input signals. In this brief, the idea is to show that error corrected codes (ECCs) can protect parallel filters in which each filter is the equivalent of a bit in a traditional ECC. When the number of parallel filter is large, it provides high fault tolerant to those complex system. The proposed technique uses parallel FIR filters which results in high fault tolerant and low-cost implementation.

Index Terms—Error correction codes (ECCs), filters, soft errors.

## I. INTRODUCTION

Electronic circuits areprogressively present in most of the applications like automotive, medical,and space applications. Those applications depend on functions where reliability is very critical.In those applications, the circuits need to give some level of adaptation to non-critical failure like high fault tolerant.This need is additionally augmented by the intrinsic responsibility challenges of advanced CMOS technologies that embody, e.g., producing variations and soft errors. A variety of techniques is accustomed to defend a circuit from errors.Those vary from modifications within the producing method of the circuits to scale back 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 [1]. to feature redundancy, a general technique called triple modular redundancy (TMR) is used. The TMR, that triplicates the look and adds pick logic to correct errors, is usually used. However, it quite triples the area and power of the circuit, one thing that will not be acceptable in some applications. Once the circuit to be protected has recursive or structural properties, a stronger choice is to take advantage of those properties to implement fault tolerance. One example is signal processing circuits that specific techniques are planned over the years [2].

Digital filters are one of the frequently used in signal processing circuits and several other techniques are projected to guard them from errors. Most of them have centered on finite-impulse response (FIR) filters. As an example, in [3], the employment of reduced exactitude replicas was projected to cut back the price of implementing modular redundancy in FIR filters. In [4], a relationship between the memory parts of associate degree FIR filter and also the input sequence was accustomed to find the errors. Different schemes have exploited the FIR properties at a word level to additionally bring home the bacon fault tolerance [5]. The employment of residue variety systems [6] and arithmetic codes [7] has additionally been projected to guard filters. Finally, the employment of various implementation structures of the FIR filters to correct errors with only 1 redundant module has additionally been projected [8]. All the techniques mentioned thus far, the protection of one filter is taken into account.

However, it's progressively common to search out systems during which many filters operate in parallel. This can be the case in filter banks [9] and in several trendy communication systems [10]. For those systems, the protection of the filters are often addressed at a better level by considering the parallel filters because the block to be protected. This concept was explored in [11], wherever 2 parallel filters with identical response that processed completely different input signals were thought of. It was absolutely shown that with only 1 redundant copy, single error correction are often enforced. Therefore, a big price reduction compared with TMR was obtained.

In this transient, a general technique to defend parallel filters is conferred. As in [11], parallel filters with identical response that uses different input signals are well-thought-out. The new approach is predicated on the appliance of error correction codes (ECCs).InSection II, parallel filters with the same response is shown. In Section III, the proposed scheme is presented. Section IV presents simulation results. Finally, the conclusions are summarized in Section V.

## II. PARALLEL FILTERS WITH THE SAME RESPONSE

A discrete time filter implements is given by the equation:

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

where $x[n]$&$y[n]$ are the input and the output signals, and $h[l]$, impulse response of the filter [12]. When the response h[l] is nonzero, just for a finite variety of samples, the filter is understood as a FIR filter, otherwise the filter is associate as infinite impulse response (IIR) filter. There are many structures to implement both FIR and IIR filters.

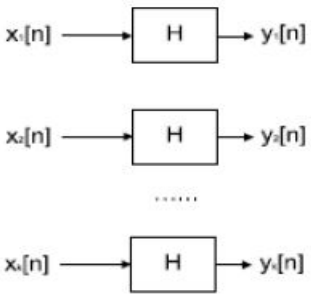


**Fig.1. Parallel filters with the same response**

The parallel filters with the same response for different input signals are shown in the above fig. 1. This type of parallel filters are increasingly used in communication system since they use many channel that run in parallel.

For example, if we wish to find out any combination of outputs $Y_i[n]$ then, we can obtain by combining the corresponding inputs $X_i[n]$ i.e.

$$Y_1[n]+Y_2[n] = \sum_{l=0}^{\infty}(X1[n-l] + X2[n-l]).h[l] \quad (2)$$

This simple observation is utilized in the subsequenttodevelop the projected fault tolerant implementation.

## III. PROPOSED METHOD

The proposed technique is based on utilization of the ECCs. It takes a block of k-bits as inputs and producing a block of n-bits as output by adding n-k parity check bits [13]. The inputs k-bits are combined with XOR combination for the parity check bits. So, it is mandatory to design properly those combinations so that it will detect and correct errors automatically.

Let's consider a simple Hamming code [14] with k=4 and n=7. For this special case, we need 3 parity check bits i.e. $p_1$, $p_2$, $p_3$ are associated with the functions of the data bits i.e. $d_1$, $d_2$, $d_3$, $d_4$.

$$p1=d_1 \oplus d_2 \oplus d_3$$

$$p3=d_1 \oplus d_2 \oplus d_4$$

$$p4=d_1 \oplus d_3 \oplus d_4 \quad (3)$$

So, the resulting data and parity check bits are stored in the memory. If any error is found, then it can be recovered from the memory and correct those errors.

Table 1. Error Location in Hamming Code

| $S_1S_2S_3$ | Error bit correction | Action |
|---|---|---|
| 000 | No error | None |
| 111 | d1 | Correct d1 |
| 110 | d2 | Correct d2 |
| 101 | d3 | Correct d3 |
| 011 | d4 | Correct d4 |
| 100 | p1 | Correct p1 |
| 010 | p2 | Correct p2 |
| 001 | p3 | Correct p3 |

$$G=\begin{bmatrix} 1\,0\,0\,0 & 1\,1\,1 \\ 0\,1\,0\,0 & 1\,1\,0 \\ 0\,0\,1\,0 & 1\,0\,1 \\ 0\,0\,0\,1 & 0\,1\,1 \end{bmatrix}$$

1 1 1 0 1 0 0

H= 1 1 0 1 0 1 0

1 0 1 1 0 0 1

Within the example thought of, a fault ond1 can cause errors on the 3 parity checks; a fault on d2only in p1 and p2; a fault on d3 in p1 and p3; and at last afault on d4 in p2 and p3. Therefore, the information bit in error islocated and therefore the error is corrected. This is often unremarkablyformulated in terms of the generating G and redundant check Hmatrices as shown above.

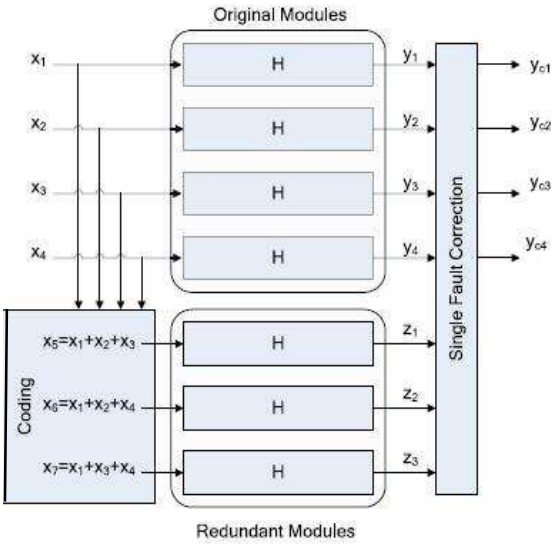


Fig. 2. ECC-based scheme for four filters and a Hamming code.

This ECC-based scheme reduces the protection overhead compared with the use of TMR. Table I summarizes the number of redundant filters needed for different parallel filter configurations. It can be observed that the number grows with the logarithm in base two on the number of filters. The cost reductions were confirmed by some case study implementations

Table 2. - Number of Redundant Filters in the ECC-based approach

| No. of Parallel Filters | No. of Redundant Filters |
|---|---|
| 4 | 3 |
| 8 | 4 |
| 16 | 5 |
| 32 | 6 |

Encoding is computed as y = x • G and error detection by $s = y \cdot H^T$, wherever the operator • is based on module 2 addition (XOR) and multiplication. Correction is performed utilizing the vector s, referred to as syndrome, to identify the bit in error. The correspondence of values of s to error position is captured in Table I. Once the inaccurate bit is identified, it's corrected by merely inverting the bit. This code theme is applied to the parallel filters considered by process a group of check filters zj .For the case of four filters y1, y2, y3, y4 and therefore the Hamming code, the check filters would be

$$z_1[n]=\sum_{i=0}^{ps}(x1[n-1]+x2[n-1]+x3[n-1]).h[i]$$

$$z_2[n]=\sum_{i=0}^{ps}(x1[n-1]+x2[n-1]+x4[n-1]).h[i]$$

$$z_3[n]=\sum_{i=0}^{ps}(x1[n-1]+x3[n-1]+x4[n-1]).h[i]$$

and the checking is performed by testing if

$z_1[n]=y_1[n]+y_2[n]+y_3[n]$

$z_2[n]=y_1[n]+y_2[n]+y_4[n]$

$z_3[n]= y_1[n]+y_3[n]+y_4[n]$

For example, if a fault is detected on $y_1$, it can be corrected using

$y_{c1}[n]= z_1[n]-y_2[n]-y_3[n]$.

## IV. SIMULATION RESULTS

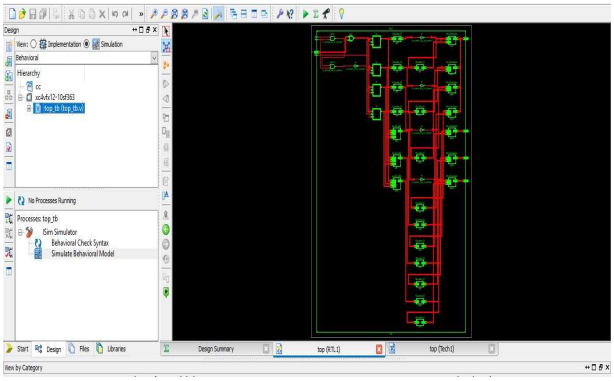

Fig. 3. RTL Schematic.

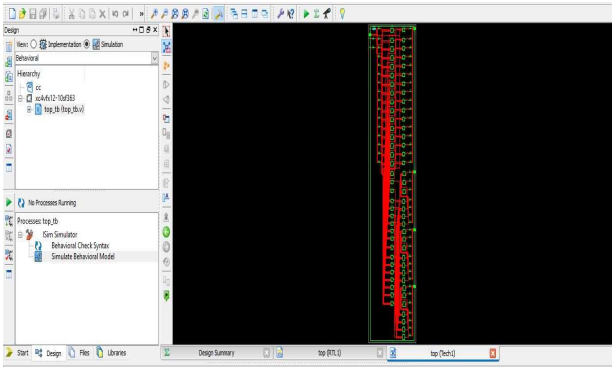

Fig. 4. Technology Schematic.

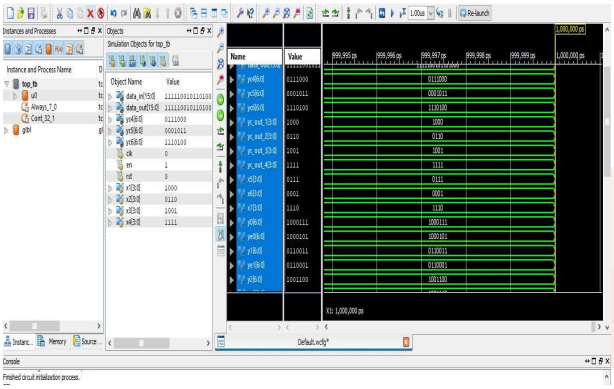

Fig. 5. Test Bench of The Proposed Technique.

The proposed method has been implemented using Verilog code and Xilinx 14.6 and mapped to a Xilinx vertex 4XC4VLX80 device.

## V. CONCLUSION

A new technique to implement fault-tolerant parallel filters has been given during this temporary. The proposed technique exploits the dimensionality of filters to implement anerror correction mechanism. Especially, two redundant filters whose inputs are linear of the original filter inputs accustomed to find and fixed the errors.The code writing of these linear combination was developed as a general drawback to then show however it will with efficiency be enforced.ThePractical implementation was illustrated with 2 case studies that were evaluated for associate FPGA implementation and compared with an antecedently projected technique. That technique depends on the employment of codes such every filter is treated as a bit within the ECC. The results show that the proposed technique outperforms the existing technique (lower prices achieving similar fault-tolerant capability).Finally, the case studies are evaluated for a field-programmable gate array (FPGA) implementation and compared with the previously proposed ECC-based technique. Therefore, the proposed technique is helpful to implement faulttolerant parallel filters.

## VI. 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. 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. 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.