INTERNATIONAL RESEARCH JOURNAL IN ADVANCED ENGINEERING AND TECHNOLOGY (IRJAET) www.irjaet.com

ISSN (PRINT) : 2454-4744 | ISSN (ONLINE): 2454-4752

Vol. 1, Issue 4, pp.298 - 303, November, 2015

FAULT TOLERANT PARALLEL FILTERS USING EFFICIENT CODING SCHEMES

M.SHANMUGA PRIYA[1], MR.S.SATHISH. M.E,MBA,(PHD).[2]

[1]PG Student, Jayalakshmi Institute of Technology, Dharmapuri,
[2]Asst professor, Jayalakshmi Institute of Technology, Dharmapuri.

ARTICLE INFO

Article History:

Received 1st Dec, 2015
Received in revised form 3rd Dec,2015
Accepted 4th Dec, 2015
Published online 6th Dec, 2015

Keywords:

Parallel filters
Coding
Soft errors

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 paper, 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 an FPGA implementation.

1. INTRODUCTION

Parallel filters are commonly found in modern signal processing and communication systems [1]. In many cases, the filters perform the same processing on different incoming signals as there is a tendency to use multiple input multiple output systems [2]. This parallel operation can be exploited for fault tolerance. In fact, reliability is a major challenge for electronic systems [3]. 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. One classical example is the use of Triple Modular Redundancy (TMR) in which the design is tripled and a majority vote of the outputs is used to correct errors. Another example is the use of Error Correction Codes (ECCs) to protect the bits stored in memories [5].

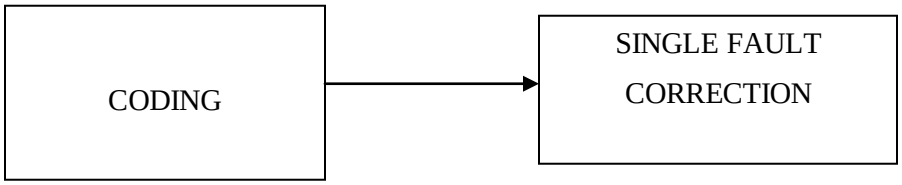

Fig.1. General block diagram

In this case, a number of parity checks are computed and stored in the memory so that errors can be detected and corrected when the data is read. Finally, for applications that have a regular structure and properties, those can be exploited to detect and correct errors with a lower cost than TMR. This is the case for many signal processing circuits [6]. In many cases, ECCs or specific protection techniques are combined with TMR to achieve a complete protection. For example, the ECC encoders and decoders may be protected with TMR to ensure that they are not affected by errors. In those cases, TMR is used to protect a small part of the circuit that cannot be protected by the ECC or the specific technique. 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 also been studied. Another option to perform error correction is to use two different filter implementations in parallel [11]. All those techniques focus on the protection of a single filter.

2. 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]:

$$y[n] = \sum_{l=0}^{\infty} x[n-l] \cdot h[l]$$

The impulse response can be infinite or be non-zero for a finite number of samples. In the first case the filter is an Infinite Impulse Response (IIR) filter and in the second a Finite Impulse Response (FIR) filter. In both cases, the filtering operation is linear such that:

$$y_1[n] + y_2[n] = \sum_{l=0}^{\infty} \left(x_1[n-l] + x_2[n-l]\right) \cdot h[l]$$

This property can be exploited in the case of parallel filters that operate on different incoming signals. In this case, four filters with the same response process the incoming signals x1[n], x2[n], x3[n], x4[n] to produce four outputs y1[n], y2[n], y3[n], y4[n]. To detect and correct errors, each filter can be viewed as a bit in an Error Correction Code (ECC) and redundant filters added to form parity check bits.

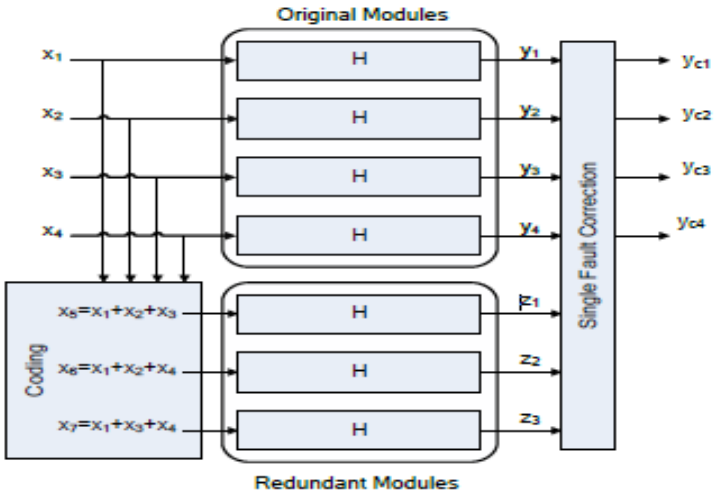

Fig.2. ECC based scheme for four filters and a Hamming code

In those works, the processing of the linear system is modified to incorporate error detection and correction mechanisms. This is different from the approach proposed in this paper where inputs are encoded but the processing of the filters is not modified. In the following, the use of a coding scheme for parallel filters in which the redundant filters are constructed as linear combinations of the original filters with arbitrary coefficients is explored.

3. CODING FOR FAULT TOLERANT PARALLEL FILTERS

The proposed scheme is illustrated in Figure 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:

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \\ a_{51} & a_{52} & a_{53} & a_{54} \\ a_{61} & a_{62} & a_{63} & a_{64} \end{pmatrix}$$

Therefore the error check will ensure error detection when the sums of the columns in each of the matrixes are different and non-zero. Since the coding matrix A is known, matrix C can be determined in advance, and stored for error detection. The complexity of the error detection and correction relies on the design of A. in order to avoid additional computations to reconstruct those outputs. This greatly simplifies the implementation. In order to illustrate the use of the proposed coding scheme in a practical application, the next section presents two case studies that are then evaluated both in terms of implementation cost and protection effectiveness.

$$\begin{cases} \overline{c_1} = \left[0, \left(\sum_{i=1}^{4} b_{i1} - \sum_{i=1}^{4} c_{i1}\right), \left(\sum_{i=1}^{4} b_{i2} - \sum_{i=1}^{4} c_{i2}\right), \left(\sum_{i=1}^{4} b_{i3} - \sum_{i=1}^{4} c_{i3}\right), \sum_{i=1}^{4} b_{i4}, -\sum_{i=1}^{4} c_{i4}\right] \\ \overline{c_2} = \left[\left(\sum_{i=1}^{4} b_{i1} - \sum_{i=1}^{4} c_{i1}\right), 0, \left(\sum_{i=1}^{4} b_{i2} - \sum_{i=1}^{4} c_{i2}\right), \left(\sum_{i=1}^{4} b_{i3} - \sum_{i=1}^{4} c_{i3}\right), \sum_{i=1}^{4} b_{i4}, -\sum_{i=1}^{4} c_{i4}\right] \\ \overline{c_3} = \left[\left(\sum_{i=1}^{4} b_{i1} - \sum_{i=1}^{4} c_{i1}\right), \left(\sum_{i=1}^{4} b_{i2} - \sum_{i=1}^{4} c_{i2}\right), 0, \left(\sum_{i=1}^{4} b_{i3} - \sum_{i=1}^{4} c_{i3}\right), \sum_{i=1}^{4} b_{i4}, -\sum_{i=1}^{4} c_{i4}\right] \\ \overline{c_4} = \left[\left(\sum_{i=1}^{4} b_{i1} - \sum_{i=1}^{4} c_{i1}\right), \left(\sum_{i=1}^{4} b_{i2} - \sum_{i=1}^{4} c_{i2}\right), \left(\sum_{i=1}^{4} b_{i3} - \sum_{i=1}^{4} c_{i3}\right), 0, \sum_{i=1}^{4} b_{i4}, -\sum_{i=1}^{4} c_{i4}\right] \end{cases}$$

In a practical implementation, the first four rows of the matrix would be an identity matrix so that the inputs to the original filters are the incoming signals. The error correction and detection logic can be simplified assuming that there is only a single error. In more detail, four checks are needed each involving five filters and excluding one. Once the faulty filter is identified, the error can be corrected by reconstructing the outputs using the remaining filters.

4. RESULTS AND DISCUSSION

To illustrate the use of the proposed scheme, two case studies that consider the protection of four and eight parallel filters are presented in this section. In both cases, a coding matrix that preserves the inputs to the original

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ a_{51} & a_{52} & a_{53} & a_{54} \\ a_{61} & a_{62} & a_{63} & a_{64} \end{pmatrix}$$

$$z_1^{corrected} = z_5 - (z_2 + z_3 + z_4)$$

filters is used. The corresponding *A* matrix is the identity matrix on the first four rows and only the last two rows have generic coefficients. The matrix is:

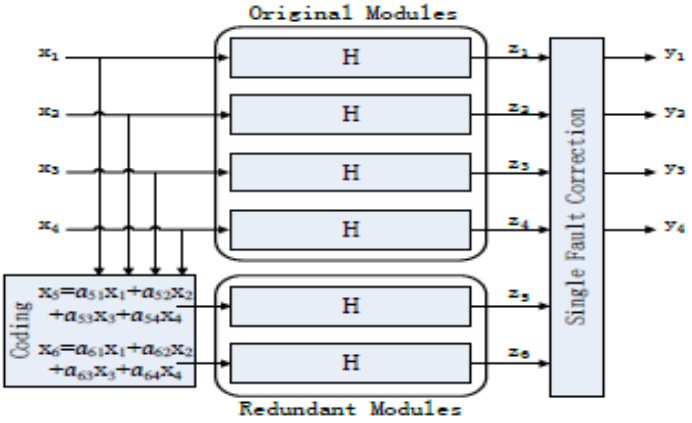

Fig.3. Practical coding scheme to protect four parallel filters

To simplify the implementation, those rows should have values that minimize the complexity of multiplications and the increase of the dynamic range in the redundant filters. This simplified checking can also be derived by noting that p1 and p2 simply check if the output values of the redundant filters (z5 and z6) match the value reconstructed using the outputs of the original filters (z1, z2, z3, z4). Therefore, in the absence of errors both p1 and p2 will be zero. From the coding matrix, for non-zero p1 and p2, it becomes clear that an error on the first filter will make p1 = p2 as both a51 and a61 are one. This provides a simple implementation as only three multiplications are needed and two of them are by powers of two and only require a shift. Another three multiplications are needed to compute p2. So in total, the scheme requires only six multiplications. This shows that the error location logic can be efficiently implemented. Finally, when an error is detected, it can be corrected by re-computing the affected filter output using the remaining original filter outputs. The structure is the same as in the first case study and so is the number of redundant filters as in the proposed scheme it does not depend on the number of filters. This is a clear advantage over the previous ECC scheme on which the number of redundant filters grows as the number of filters to protect increases. It can be seen that the same matrix structure and correction procedure can be applied to protect any given number of parallel filters. The scheme can detect and correct all errors that affect a single filter.

CONCLUSION

A new method to implement fault tolerant parallel filters has been presented in this paper. The proposed scheme exploits the linearity of filters to implement an error correction mechanism. In particular, two redundant filters whose inputs are linear combinations of the original filter inputs are used to detect and locate the errors. The coding of those linear combinations was formulated as a general problem to then show how it can efficiently be implemented. The practical implementation was illustrated with two case studies that were evaluated for an FPGA implementation and compared with a previously proposed technique. That technique relies on the use of Error Correction Codes (ECCs) such that each filter is treated as a bit in the ECC. The results show that the proposed scheme outperforms the ECC technique (lower costs achieving similar fault-tolerant capability). Therefore, the proposed technique can be useful to implement fault tolerant parallel filters. Future work will consider applying the scheme to parallel filters that have the same input signal but different impulse responses.

REFERENCES

[1] P.P Vaidyanathan. “Multirate Systems and Filter Banks”, Prentice Hall, 1993.

[2] A. Sibille, C. Oestges and A. Zanella “MIMO: From Theory to Implementation”, Academic Press, 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”, Springer Verlag, 2010.

[4] M. Nicolaidis, “Design for soft error mitigation”, IEEE Trans. on Device and Materials Reliability, vol. 5, no. 3, pp. 405–418, Sept. 2005.

[5] C. L. Chen and M. Y. Hsiao, “Error-correcting codes for semiconductor memory applications: a state-of-the-art review”, IBM J. of Research and Development, vol. 28, no. 2, pp. 124-134, 1984.

[6] A. Reddy and P. Banarjee “Algorithm-based fault detection for signal processing applications”, IEEE Trans. on Computers, 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. of IEEE International Online Test Symposium (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. of the IEEE International Online Test Symposium (IOLTS), 2012.

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

[10] Y.-H. Huang, "High-Efficiency Soft-Error-Tolerant Digital Signal Processing Using Fine-Grain Subword-Detection Processing," IEEE Trans. on Very Large Scale Integration Systems, vol. 18, no 2, pp. 291-304, Feb. 2010.

[11] P. Reviriego, C. J. Bleakley, and J. A. Maestro, "Strutural DMR: A Technique for Implementation of Soft-Error-Tolerant FIR Filters," IEEE Trans. on Circuits and Systems-II: Express 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 Electronic Letters, vol. 48, no 20, pp. 1258-1260, Sept. 2012.

[13] Z. Gao, P. Reviriego, W. Pan, Z. Xu, M. Zhao and J. Wang, J.A. Maestro, "Fault Tolerant Parallel Filters based on Error Correction Codes", IEEE Trans. on Very Large Scale Integration Systems, vol.23, no.2, pp.384-387, Feb. 2015.