International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 04 Issue 05
April 2017

VLSI design of Parallel Filters Based on Error Correction Codes for Fault Tolerance

Sudheer Chirivella, PG Scholar , Dept. Of ECE ,
Email-Id: sudheerchase@gmail.com
Gaddam Sekhar Reddy, M.Tech [Asst. Professor Dept. Of ECE]
Email-Id: gaddamsekharreddy@gmail.com
Sannikanti Kishore Babu , M.Tech ,[Ph.D] ,[Associate Professor & HOD Dept. Of ECE] , Email-Id: kish.fr@gmail.com
Vikas Group Of Institutions , Nunna (v) , Vijayawada(M) -521212, Krishna Dt.

Abstract:- *Digital filters are widely used in digital signal processing and communication systems. In most operations, the reliability of those systems is important , and fault tolerant filter implementations are required. Different techniques have been proposed that exploit the filters' structure and properties to reach the fault tolerance. The technology scaling enables more complex systems that operate in many filters. In that type of complex systems, it is common that some of the filters operate in parallel, for example, by applying the same filter to different input signals. Recently, a simple technique that exploits the presence of parallel filters to achieve fault tolerance has been presented. 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. The technique is evaluated using a case study of parallel finite impulse response filters showing the effectiveness in terms of protection and implementation cost.*

Key Terms:- Error correction codes (ECCs), filters, soft errors.

I. INTRODUCTION Electronic circuits are increasingly present in automotive, medical, and space applications where reliability is critical. In those applications, the circuits have to provide some degree of fault tolerance. This need is further increased by the intrinsic reliability challenges of advanced CMOS technologies that include, e.g., manufacturing variations and soft errors. 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 [1]. To add redundancy, a general technique known as triple modular redundancy (TMR) can be used. The TMR, which triplicates the design and adds voting logic to correct errors, is commonly used.

Digital filters are one of the most commonly used signal processing circuits and several techniques have been proposed to protect them from errors. Most of them have focused on finite-impulse response (FIR) filters. For example, in [3], the use of reduced precision replicas was proposed to reduce the cost of

implementing modular redundancy in FIR filters. In [4], a relationship between the memory elements of an FIR filter and the input sequence was used to detect errors. Other schemes have exploited the FIR properties at a word level to also achieve fault tolerance [5]. The use of residue number systems [6] and arithmetic codes [7] has also been proposed to protect filters.

it is increasingly common to find systems in which several filters operate in parallel. This is the case in filter banks [9] and in many modern communication systems [10]. For those systems, the protection of the filters can be addressed at a higher level by considering the parallel filters as the block to be protected. This idea was explored in [11], where two parallel filters with the same response that processed different input signals were considered.

As in [11], parallel filters with the same response that process different input signals are considered. The new approach is based on the application of error correction codes (ECCs) using each of the filter outputs as the equivalent of a bit in and ECC codeword. This is a generalization of the scheme presented in [11] and enables more efficient implementations when the number of parallel filters is large. The scheme can also be used to provide more powerful protection using advanced ECCs that can correct failures in multiples modules.

II. PARALLEL FILTERS WITH THE SAME RESPONSE

A discrete time filter implements the following equation:

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

where x[n] is the input signal, y[n] is the output, and h[l] is the impulse response of the filter [12]. 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. An interesting property for these parallel filters is that the sum of any combination of the outputs yi[n] can also be obtained by adding the corresponding inputs xi[n] and filtering the resulting signal with the same filter h[l]. For example

$$y1[n] + y2[n] = \sum_{l=0}^{\infty} (x1[n-l] + x2[n-l]).h[l] \qquad (2)$$

III. PROPOSED SCHEME

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 [13]. 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 [14] 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; (3)

TABLE I

International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 04 Issue 05
April 2017

ERROR LOCATION IN THE HAMMING CODE

$s_1 s_2 s_3$	Error Bit Position	Action
0 0 0	No error	None
1 1 1	d_1	correct d_1
1 1 0	d_2	correct d_2
1 0 1	d_3	correct d_3
0 1 1	d_4	correct d_4
1 0 0	p_1	correct p_1
0 1 0	p_2	correct p_2
0 0 1	p_3	correct p_3

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, those are

$$G = \begin{bmatrix} 1\,0\,0\,0\,1\,1\,1 \\ 0\,1\,0\,0\,1\,1\,0 \\ 0\,0\,1\,0\,1\,0\,1 \\ 0\,0\,0\,1\,0\,1\,1 \end{bmatrix} \quad (4)$$

$$H = \begin{bmatrix} 1\,1\,1\,0\,1\,0\,0 \\ 1\,1\,0\,1\,0\,1\,0 \\ 1\;0\,1\,1\,0\,0\,1 \end{bmatrix} \quad (5)$$

Encoding is done by computing y = x • G and error detection is done by computing $s = y \bullet H^T$, 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. The correspondence of values of s to error position is captured in Table I. Once the erroneous bit is identified, it is corrected by simply inverting the bit.

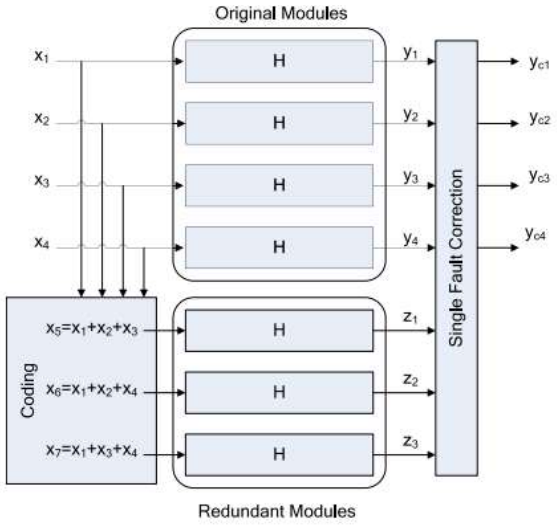

Fig 1: Proposed scheme for four filters and a Hamming code

This ECC scheme can be applied to the parallel filters considered by defining a set of check filters z j . For the case of four filters y1, y2, y3, y4 and the Hamming code, the check filters would be

$$z1(n) = \sum_{l=0}^{\infty}(x1[n-l] + x2[n-l + x3[n-l]).\, h[l]$$

$$z2(n) = \sum_{l=0}^{\infty}(x1[n-l] + x2[n-l + x4[n-l]).\, h[l]$$

$$z3(n) = \sum_{l=0}^{\infty}(x1[n-l] + x3[n-l + x4[n-l]).\, h[l] \quad (6)$$

and the checking is done by testing if

$$z1[n] = x1[n] + x2[n] + x3[n]$$

$$z2[n] = x1[n] + x2[n] + x4[n]$$

$$z3[n] = x1[n] + x3[n] + x4[n] \quad (7)$$

For example, an error on filter y1 will cause errors on the checks of z1, z2, and z3. Similarly, errors on the other filters will cause errors on a different group of zi . Therefore, as

with the traditional ECCs, the error can be located and corrected.

The overall scheme is illustrated on Fig. 2. It can be observed that correction is achieved with only three redundant filters. For the filters, correction is achieved by reconstructing the erroneous outputs using the rest of the data and check outputs. For example, when an error on y1 is detected, it can be corrected by making

$$yc1[n] = z1[n] - y2[n] - y3[n] \quad (8)$$

The vector s is also used to identify the filter in error. In our case, a nonzero value in vector s is equivalent to 1 in the traditional Hamming code. A zero value in the check corresponds to a 0 in the traditional Hamming code.

With this alternative formulation, it is clear that the scheme can be used for any number of parallel filters and any linear block code can be used. The approach is more attractive when the number of filters k is large. For example, when k = 11, only four redundant filters are needed to provide single error correction. This is the same as for traditional ECCs for which the overhead decreases as the block size increases [13].

The encoder and decoder include several additions and subtractions and therefore the possibility of errors affecting them cannot be neglected. Focusing on the encoders, it can be seen that some of the calculations of the zi share adders. For example, looking at (6), z1 and z2 share the term y1 + y2. Therefore, an error in that adder could affect both z1 and z2 causing a miscorrection on y2. To ensure that single errors in the encoding logic will not affect the data outputs, one option is to avoid logic sharing by computing each of the zi independently. In that case, errors will only affect one of the zi outputs and according to Table I, the data outputs y j will not be affected.

TABLE II
Resource comparison for four parallel fir filters

	Unprotected	TMR	Method in [7]	Proposed
Slices	2944	9020	7740	6409
Flip-flops	1224	3984	3980	2941
LUTs	5692	17256	13640	12032

IV. CONCLUSION

This brief has presented a new scheme to protect parallel filters that are commonly found in modern signal processing circuits. 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.

The proposed scheme can also be applied to the IIR filters. Future work will consider the evaluation of the benefits of the proposed technique for IIR filters. The extension of the scheme to parallel filters that have the same input and different impulse responses is also a topic for future work.

Extension Work:-

As an extension we are implementing this parallel filter with 32 bit input vectors.

REFERENCES

[1] M. Nicolaidis, "Design for soft error mitigation," IEEE Trans. Device Mater. Rel., vol. 5, no. 3, pp. 405–418, Sep. 2005.
[2] 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.
[3] B. Shim and N. Shanbhag, "Energy-efficient soft error-tolerant digital signal processing,"

IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 14, no. 4, pp. 336–348, Apr. 2006.

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

[5] 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.

[6] 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.

[7] 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, Jun. 2012, pp. 130–133.

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

[9] P. P. Vaidyanathan. Multirate Systems and Filter Banks. Upper Saddle River, NJ, USA: Prentice-Hall, 1993.

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