www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 5 Issue 09 September 2016 Page No.18163-18168

Implementation of Fault Tolerant Filter Based On Error Correction Codes for Digital Applications

Chilagani Sai Bhagath (M.Tech.) [1] *Palaka Suresh Associate Professor M.Tech, Ph.D* [2]
Holymary Institute Of Technology, Ranga Reddy District, Hyderabad, Ts 501301-India
Csbhagath@Gmail.Com[1] Palaka.Suresh@Gmail.Com[2]

Abstract : In today's world there is a great need for the design of low power and area efficient high performance DSP system.FIR filter is considered to be the fundamental device in the broad application of wireless as well as the video and image processing system. With the aim of getting the reliable operation, these filters are protected using the Error correction Code. Filters are broadly used in dealing out with signal processing and communication systems. 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. 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.

Keywords: FIR filter, Error Correction Code, Parallel Processing

1. 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,

DOI: 10.18535/ijecs/v5i9.63

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.

Digital filters are very important part of DSP. In fact, their extraordinary performance is one of the key reasons that DSP has become so popular. As the applications of digital circuits in signal processing reached to its peak, possibilities of faults and its detection and correction within digital circuitry may also need to be advanced. However, filters do not exclusively act in the frequency domain; especially in the field of signal processing many other targets for filtering exist. Correlations can be removed for certain frequency components and not for others without having to act in the frequency domain. It is common in DSP to say that a filter's input and output signals are in the time domain. This is because signals are usually created by sampling at regular intervals of time.

2. 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.

3. LITERATURE SURVEY

[1]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 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 this way using hamming codes error can be detected and corrected within the circuit.

[2] 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.

[3] 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

[4] In this paper they have proposed an architecture for the implementation of fault-tolerant computation within a high throughput multi-rate 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.

4. PROPOSED METHODOLOGY

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:

$$p_1 = d_1 \oplus d_2 \oplus d_3 \qquad p_2 = d_1 \oplus d_2 \oplus d_4 \; p_3 = d_1 \oplus d_3 \oplus d_4 \qquad (1)$$

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 recomputing 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, those are

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \qquad (2)$$

$$H = \begin{bmatrix} 1 & 1 & 10 & 1 & 00 \\ 1 & 1 & 01 & 0 & 10 \\ 1 & 0 & 11 & 0 & 01 \end{bmatrix} \qquad (3)$$

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. The correspondence of values of s to error position is captured in Table I.

Table 1: 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

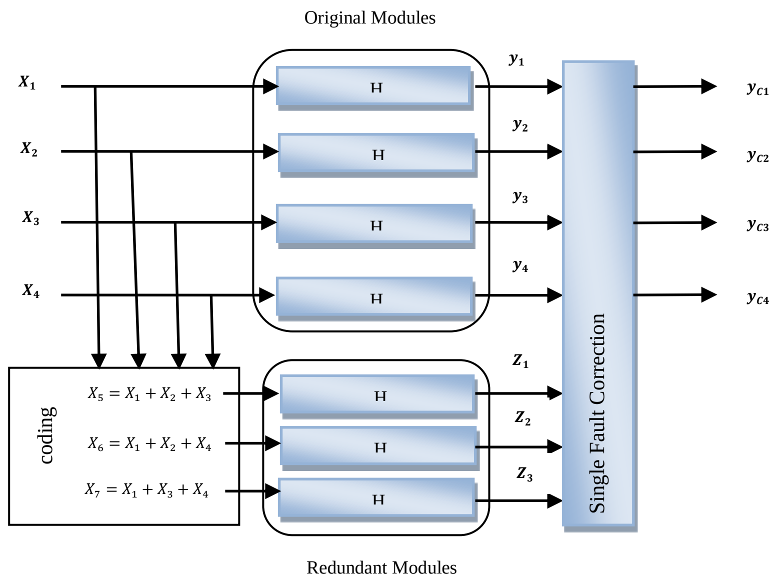


Fig.4.1: Proposed block diagram

Once the erroneous bit is identified, it is corrected by simply inverting the bit. This ECC scheme can be applied to the parallel filters considered by defining a set of check filters zj . For the case of four filters y1, y2, y3, y4 and the Hamming code, the check filters would be

$$z_1[n] = \sum_{l=0}(x_1[n-l] + x_2[n-l] + x_3[n-l]) \cdot h[l] \quad z_2[n] = \sum_{l=0}^{\infty}(x_1[n-l] + x_2[n-l] + x_4[n-l]) \cdot h[l]$$

$$z_3[n] = \sum_{l=0}^{\infty}(x_1[n-l] + x_3[n-l] + x_4[n-l]) \cdot h[l] \quad (4)$$

and the checking is done by testing if

$$z_1[n] = \sum_{l=0}(x_1[n-l] + x_2[n-l] + x_3[n-l]) \cdot h[l] \quad z_2[n] = \sum_{l=0}^{\infty}(x_1[n-l] + x_2[n-l] + x_4[n-l]) \cdot h[l]$$

$$z_3[n] = \sum_{l=0}^{\infty}(x_1[n-l] + x_3[n-l] + x_4[n-l]) \cdot h[l] \quad (5)$$

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

$$y_{c1}[n] = z_1[n] - y_2[n] - y_3[n] \quad (8)$$

Similar equations can be used to correct errors on the rest of the data outputs. In our case, we can define the check matrix as

$$H = \begin{bmatrix} 1 & 1 & 10 & -1 & 0 & 0 \\ 1 & 1 & 01 & 0 & -1 & 0 \\ 1 & 0 & 11 & 0 & 0 & -1 \end{bmatrix} \qquad (9)$$

And calculate s = y HT to detect errors. Then, 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. It is important to note that due to different finite precision effects in the original and check filter implementations, the comparisons in can show small differences. Those differences will depend on the quantization effects in the filter implementations that have been widely studied for different filter structures. The interested reader is referred to for further details. Therefore, a threshold must be used in the comparisons so that values smaller than the threshold are classified as 0.

This means that small errors may not be corrected. This will not be an issue in most cases as small errors are acceptable. The detailed study of the effect of these small errors on the signal to noise ratio at the output of the filter is left for future work. The reader can get more details on this type of analysis. 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 . The additional operations required for encoding and decoding are simple additions, subtractions, and comparisons and should have little effect on the overall complexity of the circuit. This is illustrated in Section IV in which a case study is presented.

In the discussion, so far the effect of errors affecting the encoding and decoding logic has not been considered. 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 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 cases, errors will only affect one of the zi outputs and according to Table I, the data outputs y j will not be affected. Similarly, by avoiding logic sharing, single errors in the computation of the s vector will only affect one of its bits. The final correction elements such as that need to be tripled to ensure that they do not propagate errors to the outputs. However, as their complexity is small compared with that of the filters, the impact on the overall circuit cost will be low.

5. APPLICATIONS AND ADVANTAGES

Applications

1) Error correction.
2) Digital signal processing.
3) Automotive.
4) Medical.
5) Space applications.

Advantages

Error correction is more compared with other codes

6. RESULTS

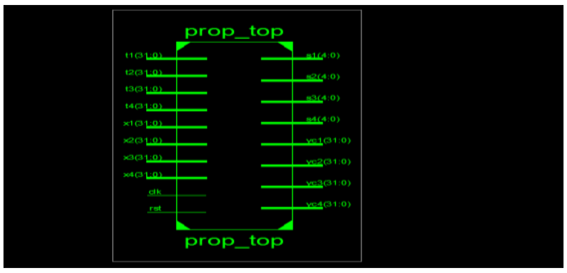


Fig. 6.1: Top module

DOI: 10.18535/ijecs/v5i9.63

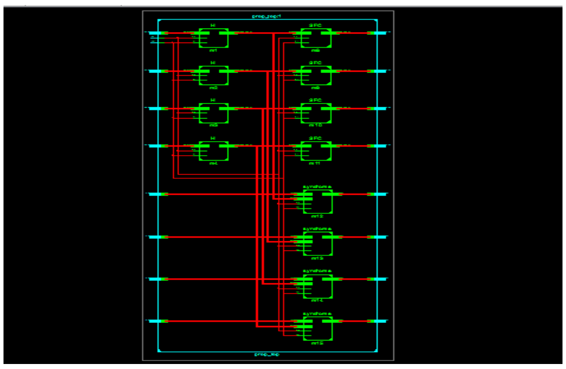


Fig. 6.2: RTL Schematic

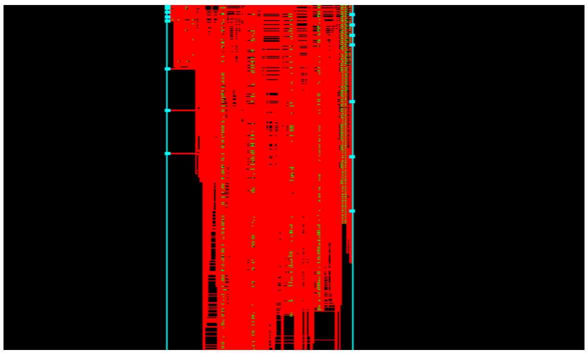
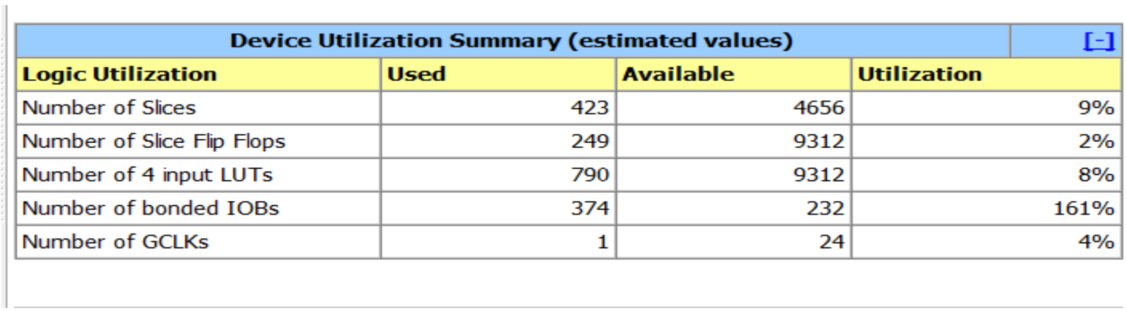
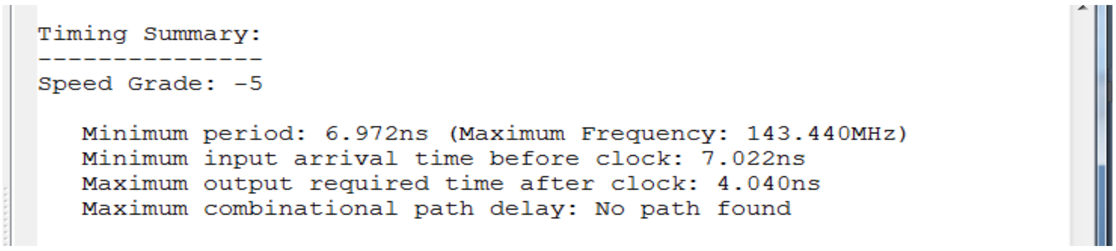


Fig. 6.3: Technology schematic



Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slices	423	4656	9%
Number of Slice Flip Flops	249	9312	2%
Number of 4 input LUTs	790	9312	8%
Number of bonded IOBs	374	232	161%
Number of GCLKs	1	24	4%

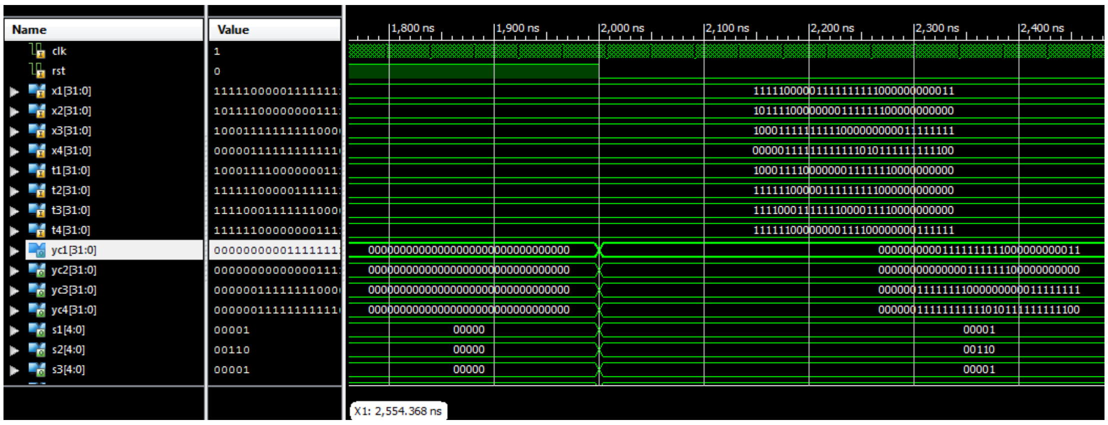
Fig. 6.4: Design summary



```
Timing Summary:
---------------
Speed Grade: -5

   Minimum period: 6.972ns (Maximum Frequency: 143.440MHz)
   Minimum input arrival time before clock: 7.022ns
   Maximum output required time after clock: 4.040ns
   Maximum combinational path delay: No path found
```

Fig. 6.5: Timing summary



Name	Value
clk	1
rst	0
x1[31:0]	1111100000111111
x2[31:0]	1011110000000011
x3[31:0]	1000111111111100
x4[31:0]	0000011111111111
t1[31:0]	1000111100000001
t2[31:0]	1111110000011111
t3[31:0]	1111000111111100
t4[31:0]	1111110000000011
yc1[31:0]	0000000000111111
yc2[31:0]	0000000000000011
yc3[31:0]	0000001111111100
yc4[31:0]	0000001111111111
s1[4:0]	00001
s2[4:0]	00110
s3[4:0]	00001

Fig. 6.6: Simulation results

7. CONCLUSION

In this project, we have presented a 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 proposed scheme can also be applied to the FIR filters. The technique is evaluated using a only two redundant filter to achieve the high error correction in ECC which also reduces the area, delay and power than previous. This will be of interest when the number of parallel filters is small as the cost of the proposed scheme is larger in that case.

REFERENCES

[1] “Fault Tolerant Parallel Filters Based on Error Correction Codes”, Zhen Gao, Pedro Reviriego, Wen Pan, Zhan Xu, Ming Zhao,Jing Wang, and Juan Antonio Maestro.

[2] “Area Efficient and Fault Tolerant FIR Filter”, Brajesh Kumar Gupta (nit jamshedpur), Associate prof. R. Sinha (nit jamshedpur)

[3] “Fault-Tolerant Computation Within Complex FIR Filters.”, P. Chan, G.A. Jullien, L. Imbert, V.S. Dimitrov, G.H. McGibney

[4] “Optimization of Self Checking FIR filters by means of Fault Injection Analysis”, S. Pontarelli, L. Sterpone, G.C. Cardarilli, M. Re, M. Sonza Reorda, A. Salsano, M. Violante.

[5] 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.

[6] R. W. Hamming, “Error correcting and error detecting codes,” Bell Syst. Tech. J., vol. 29, pp. 147–160, Apr. 1950

[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.

[10]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.

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

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