ISSN: 2278 – 909X
International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 6, Issue 5, May 2017

Implementation of fault tolerant and reconfigurable filters based on error correction codes

Sreelakshmi S , Felix M Philip

***Abstract*— Filters are broadly used in signal processing and communication systems. The filters so used for the purpose are digital filters. The complexity of communication and signal processing systems are increasing day by day, so is the number of blocks or elements that they have. In many cases, these elements operate in parallel, and perform the same processing but on different signals. A typical example of those elements are digital filters. A scheme based on error correction coding(ECC) has been proposed to protect parallel filters. In this scheme, each filter is treated as a bit, and redundant filters that act as parity check bits are introduced to detect and correct errors, and it is shown that filter inputs and outputs are not bits but numbers which enables a more efficient protection. This reduces the protection overhead and makes the number of redundant filters independent of the number of parallel filters. This new scheme allows more efficient protection when the number of parallel filters is large. The proposed scheme is first described and then illustrated with two case studies for both FIR and IIR filters. Finally, the effectiveness in protecting against errors is evaluated for a field-programmable gate array implementation. Along with these the reconfigurability of the filters are also analyzed based on distributed arithmetic (DA)-based approaches for high-throughput reconfigurable implementation of finite impulse response (FIR) filters whose filter coefficients change during runtime. The effectiveness of the design is also analyzed by comparing the various hardware requirements of the system**

***Index Terms*—Error correction codes(ECC),Parallel filters,Distributed arithmetic(DA).**

I. INTRODUCTION

Digital filters are one of the most commonly used signal processing circuits and several techniques have been used to protect them from errors. Most of them are focused on finite-impulse response (FIR) filters. Various techniques are used for the same. the error correction codes used for the same vary with time. Earlier used techniques apply only for the error detection in single filter and thus the application area of the filters is less. The use of various techniques to reduce the cost of implementing filters were also designed in earlier days. Here a design based on the hamming code technique is used for the error correction. It is analyzed for both FIR and IIR filters.

Reconfigurable finite impulse response (FIR) filter whose filter coefficients change dynamically during runtime plays an important role in the software defined radio (SDR) systems multi-channel filters and digital up/down converters . However, the well-known multiple constant multiplication (MCM)-based technique which is widely used for the implementation of FIR filters cannot be used when the filter coefficients change dynamically. On the other hand, general multiplier-based structure requires a large chip area, and consequently enforces limitation on the maximum possible order of the filter that can be realized for high-throughput applications.

In this brief, a general scheme to protect parallel filters is presented. The 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 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. It can mainly be used in design of antennas and RADAR where large numbers of filtering circuits are used.

First it gives an introduction about the various filters and application areas of them followed by various techniques used. Many papers are reviewed on the various error correction codes and a study is done on how the error correction codes can be used for the design of filters. Current system is overviewed and limitations are found and a new system based on hamming codes is proposed which is used for error correction. Later the chapter discusses about the implementation of the system and the results obtained

II. RELATED WORK

Digital filters are widely used in signal processing and communication n systems. In some cases, the reliability of those systems is critical and fault tolerant implementations are needed. So that, the idea is generalized to show that

***Sreelakshmi.S**, Department of ECE, Jawaharlal College of Engineering and Technology ,APJ Abdul Kalam Technological University, Palakkad, India,*

***Felix M Philip** Department of ECE, Jawaharlal College of Engineering and Technology, APJ Abdul Kalam Technological University Palakkad, India*

parallel FIR filters can be protected using error correction codes.

Basic approach used for error detection is the use of redundancy, where additional bits are added to facilitate detection and correction of errors. Popular techniques are:
• Simple Parity check
• Two-dimensional Parity check
• Checksum
• Cyclic redundancy check
The most common and least expensive mechanism for error-detection is the simple parity check. In this technique, a redundant bit called parity bit, is added to every data unit so that the number of 1s in the unit (including the parity becomes even). Blocks of data from the source are subjected to a check bit or Parity bit generator form, where a parity of 1 is added to the block if it contains an odd number of 1's (ON bits) and 0 is added if it contains an even number of 1's. At the receiving end the parity bit is computed from the received data bits and compared with the received parity bit. This scheme makes the total number of 1's even, that is why it is called even parity checking.

In [2] a RFFF (Reduced Faultless FIR Filter) is a technique combined with TMR used to detect and correct multiple errors is explained. TMR increases the parameters like area, power and delay. 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's functionality. To add redundancy, a general technique known as triple modular redundancy (TMR) can be used. The TMR,is a technique which triplicates the design and adds voting logic to correct errors

.

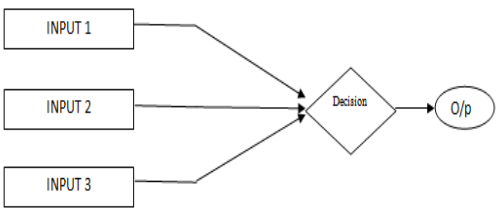

Fig 1 Triple Modular Redundancy

When the circuit to be protected has algorithmic or structural properties, a better option can be to exploit those properties to implement fault tolerance. To operate reliably in space, a hardware mitigation strategy, such as triple modular redundancy (TMR), must be applied. TMR, however, is very expensive and requires three times more hardware resources than an unmitigated circuit. Motivated by the observation that an FPGA-based radio comprises mostly arithmetic operations, this paper explores the application of reduced faultless FIR filter (RFFF) to the problem. The metric used to evaluate the effectiveness of RFFF is the bit error rate (BER) achieved by the FPGA-based radio. To fully evaluate the benefits of RFFF on a communications system, the impact of ionizing radiation on BER must be well understood.

In [5],a study on reconfigurable filters is carried out.Reconfigurable finite impulse response (FIR) filter whose filter coefficients change dynamically during runtime plays an important role in the software defined radio (SDR) systems multi-channel filters , and digital up/down converters[5]. However, the well-known multiple constant multiplication (MCM)-based technique which is widely used for the implementation of FIR filters cannot be used when the filter coefficients change dynamically. On the other hand, general multiplier-based structure requires a large chip area, and consequently enforces limitation on the maximum possible order of the filter that can be realized for high-throughput applications. Distributed arithmetic (DA)-based technique has gained substantial popularity, in recent years, for their high throughput processing capability and increased regularity which result in cost-effective and area-time efficient computing structures. The main operations required for DA-based computation are a sequence of lookup-table (LUT) accesses followed by shift-accumulation operations of the LUT output. The conventional DA implementation used for the implementation of FIR filter assumes that impulse response coefficients are fixed and this behavior makes it possible to use ROM-based LUTs.

III. PROPOSED WORK

The various types of techniques are verified, tested and implemented in digital signal processing circuits having parallel filters as the block to be protected. 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. The proposed scheme 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 more efficient implementations when the number of parallel filters is large. As the proposed scheme is more efficient only when the number of filters is large, it limits the area of application to the higher order when implemented using parallel FIR filters. It makes considerable difference in cost when used for lower order application of DSP. Further, the scheme cannot be used to provide more powerful protection using advanced ECCs that can correct failures in multiples modules. Many properties such as symmetric filters satisfying the perfect reconstruction condition can only be obtained by IIR filters.

Here filters having same response is used ,that is the filters are placed parallel to each other depending on the number of filters required. the modules are divided such that the original modules and the redundant modules process the same input. The only change is that the redundant modules will contain the operation of coding modules attached to

them.the coding module is nothing but where the operation to understand which filter contains error is programmed

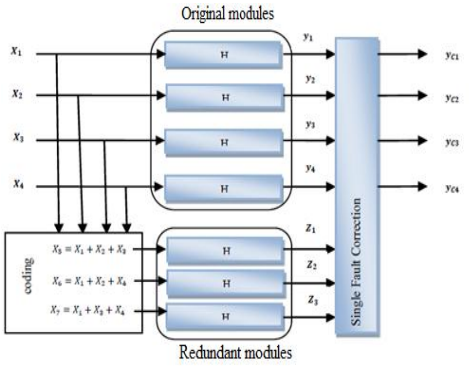

Fig 2 Architecture of proposed system

In the proposed work, we are going to extend scheme of IIR parallel filters by using more powerful multibit ECCs i.e. Hamming codes to correct errors on multiple filters.

A. 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:

$p1 = d1 \copyright d2 \copyright d3$
$p2 = d1 \copyright d2 \copyright d4$
$p3 = d1 \copyright d3 \copyright d4.$
©- convolution

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

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

$$H = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}$$

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

Table 1 Error Position Of Hamming Codes

s1 s2 s3	Error Bit Position	Action
000	no error	none
111	d_1	correct d_1
110	d_2	correct d_2
101	d_3	correct d_3
011	d_4	correct d_4
100	p_1	correct p_1
010	p_2	correct p_2
001	p_3	correct p_3

B. Reconfigurable fir filters

FPGA technology has grown tremendously from a dedicated hardware to a heterogeneous system which is considered as a popular choice in communication base stations instead of being just a prototype platform. The proposed reconfigurable FIR filter may also be implemented as part for the complete system on FPGA. Therefore, in this section, we propose reconfigurable DA-based FIR filter for FPGA implementation. The architecture suggested in Section III for high-throughput implementation of DA-based FIR filter is not suitable for FPGA implementation. The structure of Fig involves $N(2^M - 1)/M$ number of registers for the implementation of LUTs for FIR filter of length N. But, registers are scarce resource in FPGA since each LUT in many FPGA devices contains only two bits of registers. Therefore, the LUTs are required to be implemented by distributed RAM (DRAM) for FPGA implementation. But, unlike the case of RPPG , the multiple number of partial inner products $S_{l,p}$ cannot be retrieved from the DRAM simultaneously since only one LUT value can be read from the DRAM per cycle. Moreover, if L is the bit width of input, the duration of sample period of the design is L times the operating clock period, which may not be suitable for the application requiring high throughput. Using a DRAM to implement LUT for each bit slice will lead to very high resource consumption. Thus, we decompose the partial inner-product generator into Q parallel sections, and each section has R time-multiplexed operations corresponding to R bit slices. When L is a composite number given by L = RQ (R and Q are two positive integers), the index l in (8a) can be mapped into (r + qR) for r = 0; 1; :::;R - 1 and q = 0; 1; :::;Q - 1

$$y=\sum_{q=0}^{Q-1} 2^{-Rq}\left[\sum_{r=0}^{R-1} 2^{-r}\left(\sum_{p=0}^{P-1} S_{r+qR,p}\right)\right]$$

To implement the above equation , the proposed structure has Q sections, and each section consists of P DRAM-based reconfigurable partial product generators (DRPPG) and the PAT to calculate the rightmost summation followed by shift

accumulator that performs over R cycles according to the second summation.

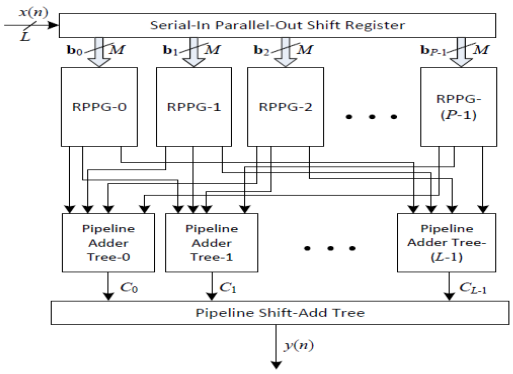

Fig 3 : RPPG structure of distributed architecture

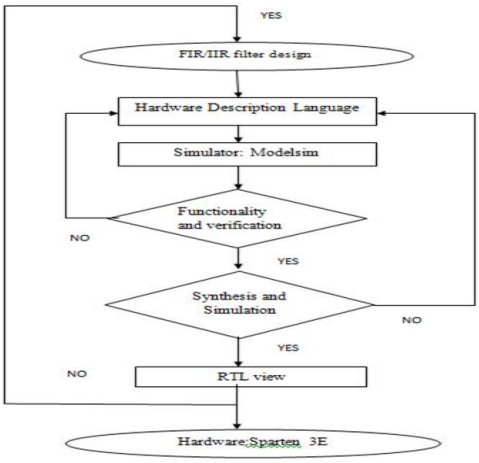

Fig 4: Flowchart Of Proposed Work

However, we can use dual-port DRAM to reduce the total size of LUTs by half since that two DRPPGs from two different sections can share the single DRAM. The proposed structure can produce QP partial inner products in a single cycle .. In the rth cycle, P DRPPGs in qth section generate P partial inner products $Sr_{+qR;p}$ for p = 0; 1; :::; P - 1 to be added by the PAT. The output of the PAT are accumulated by shift-accumulator. The accumulated value is reset every R cycles by the control signal to keep the accumulator register ready to be used for calculation of the next filter output. If the maximum operating clock period is fclk, the proposed structure can support the input sample rate of fclk=R. The reconfigurable filters are designed such that there is no much change in the program, ie once the particular FIR filter is designed based on a certain window method, the values obtained from the filter response is fed to the shift register, where these values are divided among the various reconfigurable product generator. The process of changing the values into binary values and storing the MSB and LSB of the values are done in the product generator.(these operation works on the equation stated above).the values are then stored separately in the pipeline adder and finally fed to the shift add tree.

C. Proposed Work Flow

The work flow of the proposed system can be defined using the below points and the flowchart

1) Study of Digital filters and its parallel design over error detection and correction.
2) Designing of FIR filter with same response.
3) VHDL implementation and verification of FIR filters using hamming codes.
4) To study and verify efficiency of previous work using FIR filter response.
5) Designing of IIR filter with same response and different applied input signal.
6) VHDL implementation and verification of parallel IIR filters using Hamming codes.
7) Computing error detection and correction performance.
8) Simulation and performance evaluation

IV. IMPLEMENTATION

The design and implementation of the proposed system can be divided into various steps.first step is to design a particular filter both FIR and IIR and calculate the coefficients.the calculation is done based on various design steps available for the filters. After obtaining the coefficients, these are loaded into the program and and is verified for the accuracy. After the above step is done the proposed system is implemented in the filters for the parallel filters and then verified for the error correction capability. these process are verified using Modelsim simulator. The above process is verified for both the filter types. After the verification of the error correction capability, the reconfigurability is tested. It includes first finding out a filter characteristics and loading them in the LUT .the distributed arithmetic concept is used after that to calculate the coefficients of the filter. These values are automatically loaded in the program at the run time. Error correction and detection is also verified for the filters Implementing image processing applications on a general purpose computer can be easier, but not very time efficient due to additional constraints on memory and other peripheral devices. Application specific hardware implementation offers much greater speed than a software implementation. With advances in the VLSI (Very Large Scale Integrated) technology hardware implementation has become an attractive alternative. In proposed method FPGA (Field Programmable Gate Array) implementation of fault

tolerant filters is carried out. The entire algorithm is designed using VHDL in modelsim and synthesized using Xilinx ISE.It is implemented in FPGA using Sparten 6 kit and verified for the efficiency.

V. RESULTS

The result of the work can be analyzed in 3 steps

A.FIR Filters

For the FIR filters the proposed methodology was implemented for both lower and higher order filters and it can thus be analyzed that this methodology works up to 8 order filters.

First the system is verified for a system without error in the system, thus obtaining the result a 1bit parity error is added and then we can see that at the time the error goes high ,the input in which the error has been added toggles, thus the first output before the correction can be verified and found that it is not equal to the input. As the clock goes on the error detection and correction circuit corrects the parity error and thus from the next clock cycle the output goes in synchrous with the input.

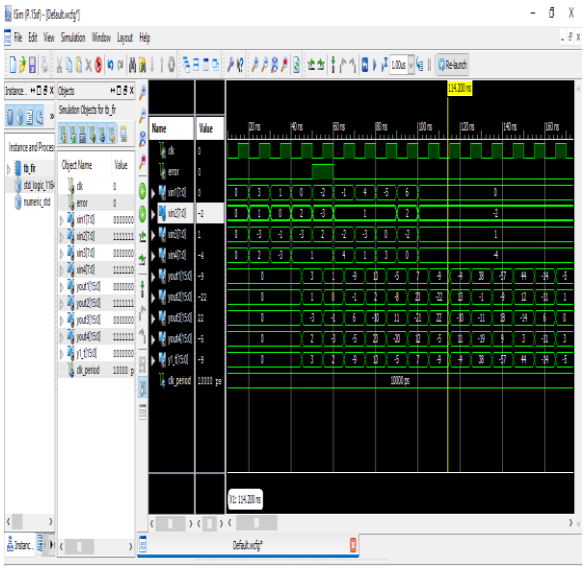

Fig 5 :Screenshot of output waveform of fault tolerant FIR filter

The waveform is analyzed in mainly two ways, first for the normal filter operation where the coefficients are kept on adding depending on the delay and the input clock signal. As the filter operation is found to be correct. Next is the verification of the parallel filter operation along with the error detection and correction capability. The error detection and correction can be verified by analyzing which filter value gets changed with the clock cycle. Once the filter is obtained, the next step is to analyze in the coding whether it is being corrected automatically. It can be verified in the next clock cycle such that the corrected and the original filter output gets synchronized. The above mentioned proposed work is analyzed from the lower order to the higher order. it is verified from the third order to eight order filter. The waveform here is of the eighth order filter because the aim was to expand the system for the higher order filters. Also it is noted that single bit error is corrected with higher accuracy. The hamming code based error correction technique works well for the system. This system can also be done for multibit errors but there should be a slight change in the error correction technique.

B.IIR filters

The same proposed methodology can be analyzed for IIR filters. For the IIR filters as the impulse response is infinite the analysis was a bit difficult and was verified up to the 3rd order filters.upto that these filters showed the same functionality as the FIR filters, here also the output is verified at all the stages and was understood that when an error occurs in the system these filters automatically correct them. As the error signal goes high input and output varies and as the clock signal goes the next cycle the input and output synchronizes. The waveform here also can be verified for the IIR operation of the filters that is showing the addition of the coefficients and change of coefficients along with the clock signal. Here it is seen that unlike the finite impulse response filters the operation of the filters keeps on continuing after even the clock is stopped.IIR filters also thus works properly for our system.

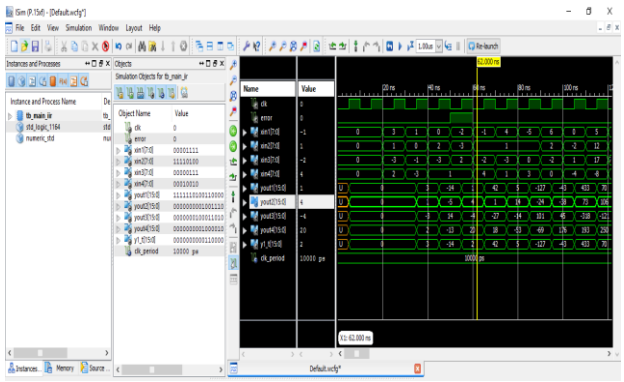

Fig 6 Screenshot of output waveform of fault tolerant IIR filters

C.Reconfigurable filters

Reconfigurable filters are those whose coefficients keep on changing during the run time. Here same as the FIR and IIR filters ,these filters are also analysed for error correction .here the coefficients are obtained from the rectangular window and are stored in the LUT of the FPGA ,these stored values are retrieved each time the program is run and then the distributed arithmetic technigue is applied before these values are added to the program. Here each time the values loaded to the program keeps on changing and hence are reconfigurable in nature.here also when the error signal is given to a particular input and the output toggles at the time of operation and this is corrected automatically to synchronise the output and input.thus the proposed methodology works in all the three cases.

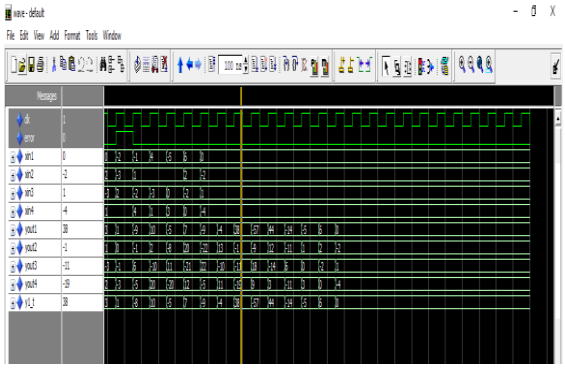

Fig 7 Screenshot of output waveform for reconfigurable filters

The filter thus obtained contain all the characteristics same as the designed FIR filter , the advantage is that here the filter coefficients which keep on changing as the program is run.this makes the proposed work to be used in various filters without much changing the coding and methodology.the analysis is done such that the coefficients are monitored throughout the running of the code .the frequency response of the filter can also be analysed for the effective operation of the code.

D.Result Analysis

The efficiency and advancement of the system can be verified using the values of the LUT(look up tables) used in the system. these can be verified by comparing the values of the flip-flops used . These can be tabulated as

Table 2 Resource Comparison For Four Parallel Filters

	Unptotected	TMR Technique	Arithmetic Residue Codes	Proposed System
Slices	2944	6611	7740	6409
FlipFlops	1328	3196	3980	2408
LUTs	5692	12401	13640	12032

From the above table it can be seen that the number of slices,flipflops and LUTs are less used in the proposed sytem.thus stating the fact that the sytem is efficient and less power consuming.the saving can also be calculated.here we can understand that when the following design is used in hardware implementation,these values of the look up tables can be used to understand or estimate the area required for the storage of the filter values and the program.it can also be seen that from the above table that the proposed system is much more efficient than the earlier technologies used for error detection

Table 3 Saving Analysis Of The Resources.

	Saving compared to TMR	Saving compared to arithmetic codes
Slices	30.5	17.1
Flipflops	24.6	39.4
LUTs	29.7	11.7

From the above table it can be noted that the proposed system is more effective than the existing systems.compared to the triple modular redundancy technique which is most commonly used, the hamming code technique gives 30 percent saving in the slices used. LUT s in the proposed system is also less used than TMR technique that makes the system more efficient.the percentage of saving also helps in analyzing the power requirement of the modules. In case the number of parallel filters are increased,the saving is more in the proposed system

VI.ADVANTAGES AND APPLICATION

A.Advantages:

- Low power design: the proposed work offers a lower power design that is it is very conservative.
- Less area required: as analyzed above ,since the number of look up tables and slices are less in the system, it offers low area based design.

B.Applications

- Filtering operations: as the filters proposed is for the automatic error detection and correction, it can be effectively used in area where large filtering is required without much error effecting the circuit.
- IF stages of the receivers.: in the IF stages of the receivers also where the filters are used this work can be applied.
- RADAR applications: in the receiver sections of the RADAR these are very applicable , as these are mainly used in very secret caes where a small error is to be detected.

VII.CONCLUSION AND FUTURE SCOPE

The work 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. A case study has also been discussed to show the effectiveness of the scheme in terms of error correction and also of circuit overheads. The technique provides larger benefits when the number of parallel filters is large. The proposed scheme is also applied to the IIR filters. It was seen that IIR filters responded to the proposed model for the lower order filters only.the reconfigurability of the FIR filters are also analyzed and obtained that these sytems are reconfigurable and hence provide a improvement in signal processing circuits. The extension of the scheme to parallel filters that have the same input and different impulse responses is also a topic for future work. Another interesting topic to continue this brief is to explore the use of more powerful multibit ECCs, such as Bose–Chaudhuri–Hocquenghem codes, to correct errors on multiple filters.

REFERENCES

1 .Zhen Gao, Pedro Reviriego, Wen Pan, Zhan Xu, Ming Zhao,Jing Wang, and Juan Antonio Maestro (2015), "Fault Tolerant Parallel Filters Based on Error Correction Codes*", Ieee Transactions On Very Large Scale Integration (Vlsi) Systems, Vol. 23*

2. Brajesh Kumar Gupta (nit Jamshedpur), Associate prof. R. Sinha (nit Jamshedpur)(2013),] "Area Efficient and Fault Tolerant FIR Filter", *International Journal of Scientific & Engineering Research, Volume 4,*

3. P. Chan, G.A. Jullien, L. Imbert, V.S. Dimitrov, G.H. McGibney,(2013), "Fault-Tolerant Computation Within Complex FIR Filters,*international jounal of scientific research*

4. S. Pontarelli, L. Sterpone, G.C. Cardarilli, M. Re, M. Sonza Reorda, A. Salsano, M. Violante. (2007),"Optimization of Self Checking FIR filters by means of Fault Injection Analysis", i*nternational symposium on defect and fault tolerant VLSI systems*

5. Sang Yoon Park, Member, IEEE and Pramod Kumar Meher,(2013) Efficient FPGA and ASIC Realizations of DA-Based Reconfigurable FIR Digital Filter, *Ieee Transactions On Circuits And Systems-Ii:*

6. Gurneet Kaur and Amandeep Singh Sappal,(2014),"Design of FIR Filter using Distributed Arithmetic Architecture*", International Journal on Recent and Innovation Trends in Computing and Communication*

7. B. Shim and N. Shanbhag(2006), "Energy-efficient soft error-tolerant digital signal processing," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.,vol. 14,*

8. R. W. Hamming,(2010) "Error correcting and error detecting codes," *Bell Syst. Tech. J., vol. 29*, pp.

9. *T. Hitana and A. K. Deb(*2004), "Bridging concurrent and non-concurrent error detection in FIR filters,". *Norchip Conf.,*

10. Y.-H. Huang,(2010) "High-efficiency soft-error-tolerant digital signal processing using fine-grain subword-detection processing*," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 18,*

11. S. Pontarelli, G. C. Cardarilli, M. Re, and A. Salsano,(2008) "Totally fault tolerant RNS based FIR filters," *in Proc. IEEE IOLTS,*

Sreelakshmi.S obtained her Bachelor's degree in Electronics and Communication Engineering from University of Calicut,Kerala,India. Then she obtained her Master's degree in Electronics Design Technology from APJ Abdul Kalam Technological University,Kerala,India

Felix M. Philip (Felix Mattathil Philip) obtained his Bachelor's degree in Electronics And Communication Engineering from Mahatma Gandhi University, Kerala, India. Then he obtained his Master's degree in VLSI Design from Anna University, Tamil Nadu, India and he is Research Scholar in Electronics and communication Engineering ("Image Processing"), in Hindustan University, Chennai, India. He is currently an Assistant Professor at the Faculty of Electronics and Communication Engineering in, Jawaharlal College Of Engineering And Technology.