

FPGA Implementation of Reduced Precision Redundancy Protected Parallel Fir Filters

Sneha P¹ Mrs. Jamuna S²

¹(MTECH student, Dept. Electronics & Communication, Dayananda Sagar College of Engineering, India)

²(Associate professor, Dept. Electronics & Communication, Dayananda Sagar College of Engineering, India)

Abstract: In Signal processing and Communication systems, Digital Filters are widely used for computational purposes. These filters must be protected from induced faults. For these systems, filters must be implemented in such a way as to achieve fault tolerance so that reliability is improved. Many techniques have been proposed that exploit filter structures and properties. As technology scales, complex systems use many filters. In those complex systems, some of the filters operate in parallel. Parallel Filters are those filters whose impulse response is same for different inputs that produce different outputs. In this paper Finite Impulse Response (FIR) filters are used. Different techniques like Triple Modular Redundancy (TMR) technique which add redundancy to the logic, Error Correction Codes (ECCs) using Hamming Code, Decimal Matrix Code (DMC), etc have been proposed. In this paper, a new method called Reduced Precision Redundancy (RPR) which is an extension of TMR technique have been proposed. This paper illustrates the RPR protection method applied for Parallel FIR filters and compares with the TMR and ECC methods in terms of implementation cost, delay, power and ease of correcting the faults. This design is simulated in Modelsim and Xilinx Isim and implemented in FPGA Virtex-5 kit. It is shown that RPR is preferred when multi-bit error occurs.

Key words: Error Correction Codes (ECCs), Finite Impulse Response (FIR) Filters, Hamming Code, Induced Faults, Parallel Filters, Reduced Precision Redundancy, Triple Modular Redundancy (TMR).

I. Introduction

Digital Filters plays a major role in communication systems. The main purpose of using filters is to remove the undesired signal components and passing only desired signal at the output. As the CMOS technology advances, number of techniques can be used to protect filters from errors. Those range from modifications in the manufacturing process of the circuits to reduce the number of errors by exploiting their arithmetic or structural properties or by adding redundancy at the logic or system level to ensure that errors do not affect the system functionality. When the circuit is to be protected has arithmetic or structural properties, we can exploit these properties to implement fault tolerance. One such method is application of Error Correction Codes (ECCs) [1] using each of the filter outputs as the equivalent of a bit. When the number of parallel filters is more, then this scheme enables more efficient implementation. But the disadvantage is it can only correct single bit error. A general technique is used to add redundancy known as Triple modular Redundancy (TMR) [2]. The TMR triples the design and produces output by choosing from the decision logic and voter circuit which eliminates the error. But the disadvantage of TMR is that the area and power consumption is more as the number of circuits is more. Other techniques have also been proposed to protect the Digital filters from injected faults. In [3], a relationship between input sequence of FIR filter and memory elements (used for shift registers) are used to detect errors. In [4], to achieve fault tolerance properties of FIR filters are used at a word level. FIR filters are protected from induced faults by making use of Residue Number Systems [5] and arithmetic codes [6]. Finally, by using different implementation structures of the FIR filters with only one redundant module has been proposed to correct errors. In all these papers mentioned so far, the protection of a single filter is considered.

Parallel Filters are used in Filter banks and protections of these systems are addressed at a higher level by considering parallel filters as a block. This idea was explored in [7] where two parallel filters with the same response that processed different input signals were considered. It was shown that with only one redundant copy, single error correction can be implemented.

In this paper, an extension of TMR method called Reduced Precision Redundancy (RPR) [8] technique is explained. This technique is same as TMR method except that it uses reduced precision replicas. This will reduce the cost of implementation thereby reducing power, delay and resources used. Comparison of TMR, ECC and RPR protected Parallel Filter techniques is done. It is shown that RPR technique is suitable for multi-bit errors in arithmetic operations.

In Section II, FIR Filter is briefly explained. Then in Section III, Parallel Filters is explained. Section IV explains about existing methods TMR and ECC. In Section V Proposed Scheme (RPR) is presented. Section

VI presents a case study to illustrate the effectiveness of the approach. Finally, the Conclusions are summarized in Section VII.

II. FIR Filters

A discrete time filter has the following equation

$$y[n] = \sum_{l=0}^{\infty} x[n-l] \cdot h[l] \tag{1}$$

where $x[n]$ is the input signal, $y[n]$ is the output and $h[l]$ is the impulse response of the filter.

In Signal Processing, Finite Impulse Response (FIR) Filters are those filters whose impulse response $h[l]$ is nonzero only for a finite number of samples that is impulse response settles to zero in finite time whereas Infinite Impulse Response (IIR) Filters are those filters whose impulse response $h[l]$ is infinite due to its feedback system which may continue to respond indefinitely.

In this paper we mainly use FIR filters because of following reasons 1) It requires no feedback which makes implementation simpler. 2) FIR filters are inherently stable because the output is the sum of finite number of finite multiples of the input values. 3) These filters can be designed as linear phase by making coefficient sequence symmetric. This property is desired for phase-sensitive applications for example data communications, and crossover filters. 4) Many Digital Signal Processors (DSPs) provide specialized hardware features to make FIR filters efficient for many applications.

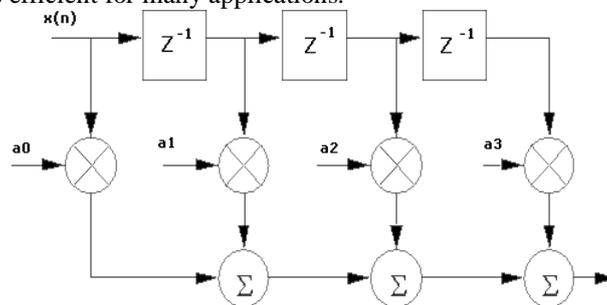


Fig.1 General Form of FIR Filter

For Casual Discrete time FIR filter of order N, the output is the weighted sum of input values.

$$y[n] = \sum_{i=0}^N a_i x[n-i] \tag{2}$$

where $x[n]$ is the input signal, $y[n]$ is the output signal, N is the order of filter and a is the value of impulse response at the i^{th} instant for $0 \leq i \leq N$ and Z^{-1} is the delay element. If the filter is a direct form FIR filter then b_i is also a coefficient of the filter. The $x[n-i]$ is usually referred as taps. Fig. 1 shows the general form of FIR filter.

III. Parallel Filters

Parallel Filters are those filters which have same impulse response with different inputs and produces different outputs. Fig.2 shows a set of k parallel filters with the same impulse response and different inputs and outputs. These Parallel filters are found in communication systems that use several channels in parallel and also in data acquisition and processing applications.

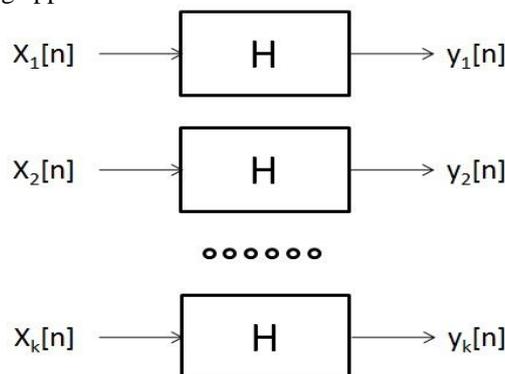


Fig.2 Parallel Filters with the same impulse response

Property of Parallel filters-The sum of any combination of the outputs $y_i[n]$ can also be obtained by adding corresponding inputs $x_i[n]$ and filtering the resulting signal with the same filter $h[l]$. For example

$$y_1[n] + y_2[n] = \sum_{l=0}^{\infty} (x_1[n-l] + x_2[n-l]) \cdot h[l] \tag{3}$$

IV. Existing Methods

4.1 ECC method

This method is based on the use of Error Correction Codes. An ECC block takes a block of k bits and produces a block of n bits by adding $(n-k)$ parity check bits [9]. The Parity check bits are xor combinations of input k data bits. Considering Hamming Code [10] with input $k = 4$ data bits and output $n=7$ bits, three parity check bits p_1, p_2, p_3 are needed which are computed as follows:

$$\begin{aligned} p_1 &= d_1 \oplus d_2 \oplus d_3 \\ p_2 &= d_1 \oplus d_2 \oplus d_4 \\ p_3 &= d_1 \oplus d_3 \oplus d_4 \end{aligned} \tag{4}$$

If there is any error on input data bits we can detect and correct using parity check bits. For example, an error on d_1 will cause errors on the three parity check bits p_1, p_2, p_3 ; an error on d_2 will affect only p_1 and p_2 ; an error on d_3 will affect only on p_1 and p_3 and finally an error on d_4 will affect p_2 and p_3 . Once the erroneous bit is identified, it is corrected by simply inverting that bit.

The same concept is applied for Parallel filters. In Fig 3, $k=4$ parallel filters H are used which is marked as Original modules and 3 redundant modules are added to produce $n=7$ filter outputs. x_1, x_2, x_3, x_4 are inputs to the filter and y_1, y_2, y_3, y_4 are corresponding filter outputs. Inputs to redundant modules are x_5, x_6 and x_7 which are the xor combination of inputs and z_1, z_2 and z_3 are corresponding check filter outputs using the property of Parallel filters

$$\begin{aligned} z_1[n] &= \sum_{l=0}^{\infty} (x_1[n-l] + x_2[n-l] + x_3[n-l]) \cdot h[l] \\ z_2[n] &= \sum_{l=0}^{\infty} (x_1[n-l] + x_2[n-l] + x_4[n-l]) \cdot h[l] \\ z_3[n] &= \sum_{l=0}^{\infty} (x_1[n-l] + x_3[n-l] + x_4[n-l]) \cdot h[l] \end{aligned} \tag{5}$$

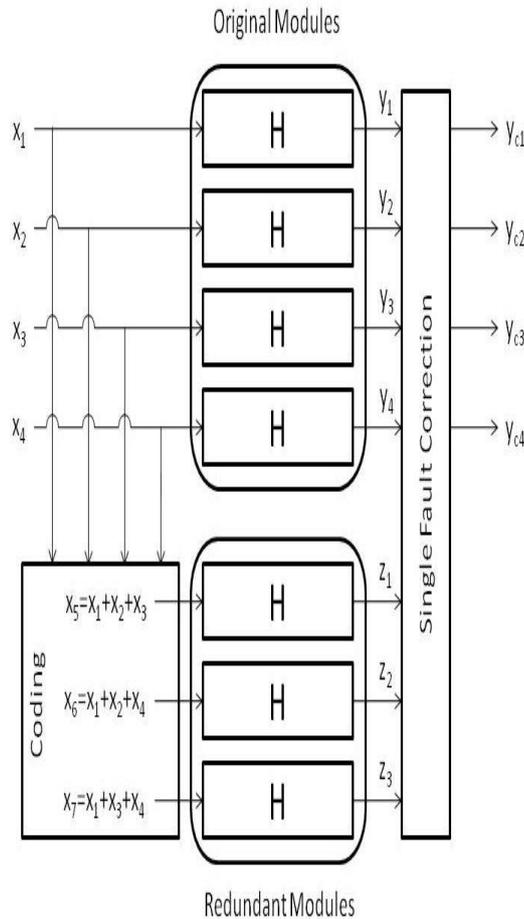


Fig.3 ECC scheme for four filters and a Hamming code

If the filter is affected by induced fault then testing is done by checking

$$\begin{aligned} z_1 [n] &= y_1 [n] + y_2 [n] + y_3 [n] \\ z_2 [n] &= y_1 [n] + y_2 [n] + y_4 [n] \\ z_3 [n] &= y_1 [n] + y_3 [n] + y_4 [n] \end{aligned} \tag{6}$$

For example, an error on filter y_1 will cause errors on the checks of z_1, z_2, z_3 . Similarly, errors on the other filters will cause errors on the different group of z_i . Therefore error can be located using ECC scheme. The correction can be done using only three redundant filters. For the filters, correction is achieved by reconstructing the erroneous output using the rest of the data and check outputs. For example, when an error on y_1 is detected, it can be corrected by making

$$y_{c1} [n] = z_1 [n] - y_2 [n] - y_3 [n] \tag{7}$$

Similarly for other erroneous filter outputs different correction equations are used.

4.2 TMR

Digital systems can be also be improved by using redundant components. For meeting future reliability requirements for digital computers in space and certain military applications a new scheme called Triple Modular Redundancy was developed. There can be two types of failures, system failures caused by permanent component failures and transient failures caused by noise. The same technique can be used for combating both types of failures. The application of redundancy together with the improvement of component reliability and reduction of system complexity will be required to solve the problem.

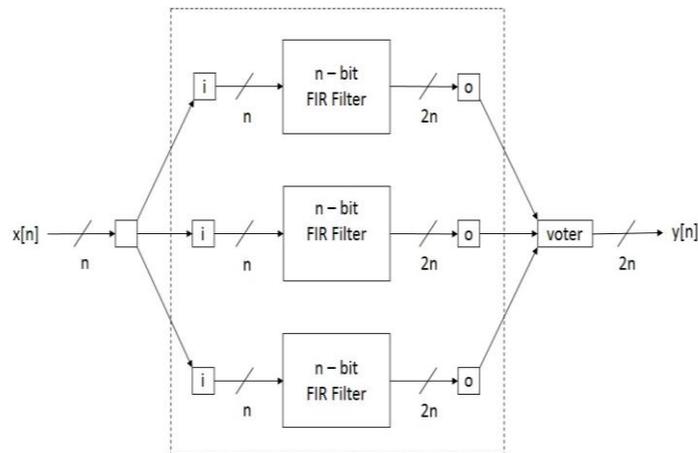


Fig.4 TMR protected n-bit FIR filter

Redundancy technique is proposed by using 1) the most reliable components and 2) the least possible complexity. This technique uses three copies of circuit and two-out-of-three voting concept at the low level. Fig. 4 shows TMR protected single n bit filter. The input $x[n]$ as well as FIR filter is triplicated and triplicated output is sent to voter circuit which acts as majority module. Output of any two filters matches then voter circuit selects that as final output.

For four Parallel Filters the same figure is implemented four times as shown in fig 5.

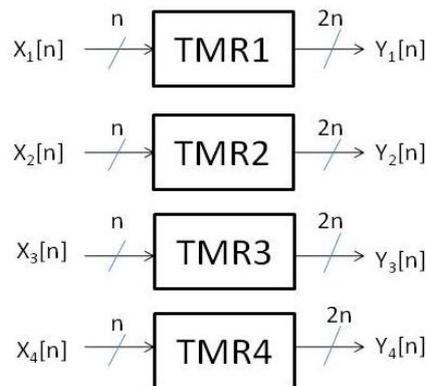


Fig.5 TMR protected four Parallel Filters

The advantage of TMR method is that it can correct multi bit errors in a filter. But there are several disadvantages 1) It triplicates the area. 2) Delay is more. 3) Use of Unequal reliabilities of modules and associated voting circuit will lower the TMR reliability. 4) This method gives correct output as long as two of the three modules are operating correctly. For large number of parallel filters, this method may decrease the TMR Reliability. Hence we go for protecting the parallel filters by using Reduced Precision Replicas.

V. Proposed Scheme – Reduced Precision Redundancy

As TMR involves large overhead in terms of circuit area, power consumption and delay, a novel algorithmic noise-tolerance (ANT) technique known as Reduced Precision Redundancy (RPR) technique is proposed. RPR is used as a part of power reduction technique for ASIC based digital signal processing systems [11, 12]. In RPR method, low (reduced) precision replicas are combined with the TMR circuit while achieving significant energy savings. In this case, when there is a difference between the full precision circuit and the replicas that exceeds the maximum expected error due to the reduced precision it is assumed that a fault is induced in the full precision block and the output from the replicas is used to correct the error. RPR applies redundancy only to the most significant numerical bits of a circuit and in this way significantly decreases the needed chip area and power consumption over that required for TMR. This approach is shown in figure 6.

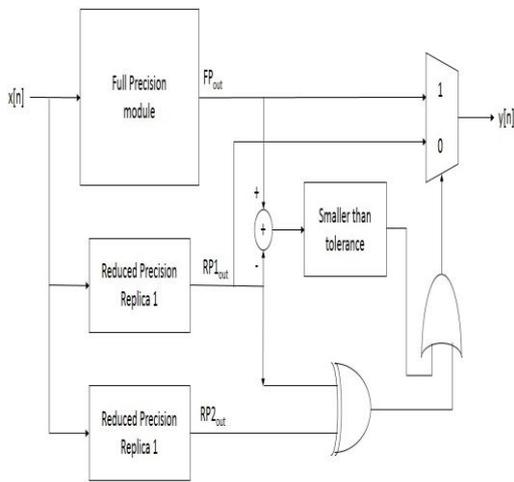


Fig.6 Reduced Precision Redundancy technique

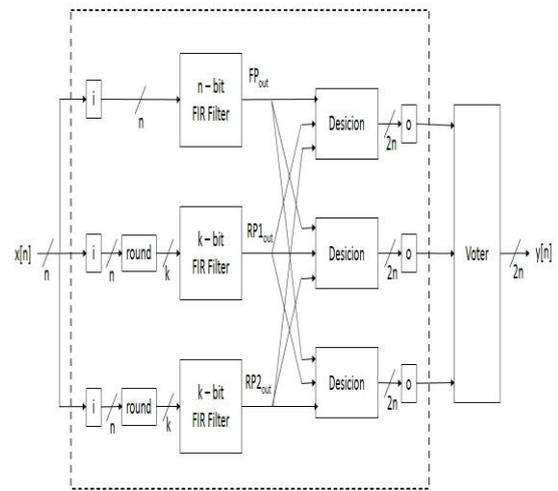


Fig.7 RPR protected n-bit FIR filter

The RP output provides an estimate of the correct FP output. Fig. 7 shows RPR technique applied for n-bit FIR filter.

Inputs to the filter are triplicated same as TMR but the second and third replicas are implemented with reduced precisions RP rounded with k bit where $k < n$. Also the decision and outputs are triplicated to avoid single points of failure in those modules. To determine the presence of error, the decision logic compares the outputs of full-precision (FP) filters with the outputs of two Reduced-Precision (RP) filters. When both RP filters are equal and error tolerance is more, then RP filter output is selected else FP output is selected which means no fault is induced. This is formulated as follows:

$$\text{If } (|FP_{out} - RP1_{out}| > T_h) \text{ AND } (RP1_{out} = RP2_{out})$$

$$\text{Output} \leq RP2_{out}$$

$$\text{Else}$$

$$\text{Output} \leq FP_{out}$$

RPR has two parameters that must be taken into consideration while employing the technique. The impact of these parameter settings is determined by arithmetic error

$$\epsilon = |FP_{true} - RP_{true}| \tag{8}$$

where FP_{True} and RP_{True} are the outputs of the FP and RP filters respectively when no fault is induced. First parameter is the size of the RP modules. In this paper the size of the RP filters is measured by the bit-width of the filter input signal, k. A larger RP filter gives a better estimate of the FP filter. This results in a better detection of errors in the FP filter and a lower ϵ . A smaller RP filter is desirable because a smaller RP filter reduces the cost of mitigation.

The second parameter for RPR is the threshold value T_h . A threshold that is too small will cause the RP output to be chosen even when there are no errors in the FP module. To prevent this, $T_h > \max\{\epsilon\}$ is required. On the other hand, if T_h is too large, the FP output is used even when there are significant errors in

that module. Hence to inculcate both parameters to a suitable level, 8-bit inputs and 8-bit 16 tap FIR filters are considered in this paper. The RP modules are bypassed if one of them is disabled, and the RP output is used whenever the output of a disabled FP module falls outside the error-limiting threshold value. The same concept is applied for four parallel filters taking four inputs and producing four outputs which utilises FIR filters of same impulse response as shown in figure 8.

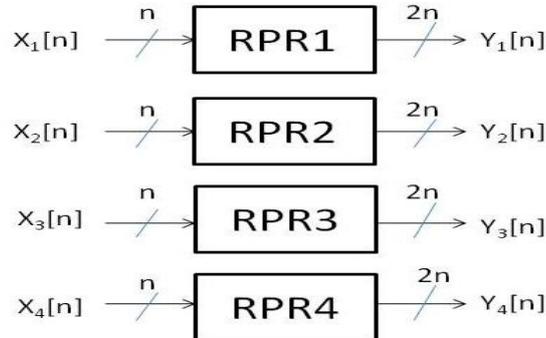


Fig.8 RPR protected four Parallel Filters

VI. Case Study

In this paper, a set of four parallel filters with 16 coefficients are used. The input data $x[n]$ and coefficients b_i are quantized to 8 bits. The filter output is quantized with 16 bits. Error is introduced at the output of fourth filter $y_4[n]$. Using RPR technique, single bit error is detected and corrected. Verilog Code is written and simulated in ModelSim and Xilinx ISim and implemented in Virtex 5 xc5vlx110t-2ff1136. Fig. 9 shows RTL schematic and Fig. 10 shows simulation results. Inputs given are $x_1=26$, $x_2=35$, $x_3=26$ and $x_4= 36$. Threshold used is 4096. Fault is induced in fourth filter, that is, f_{43} which is marked in red is erroneous output and Corrected output is marked in blue, which is y_4 . This method is preferred when multi-bit error occurs.

The idea of this paper is to compare the resources used by the RPR, ECC and TMR method. From synthesis report of all the three methods it is observed that area utilization is less in RPR method compared to other methods saving 30% of all resource types (slices, flip-flops and LUTs) which is shown in TABLE 1. Delay in proposed scheme is less compared to TMR and more compared to ECC and power consumption is less in RPR compared to other methods shown in TABLE 2.

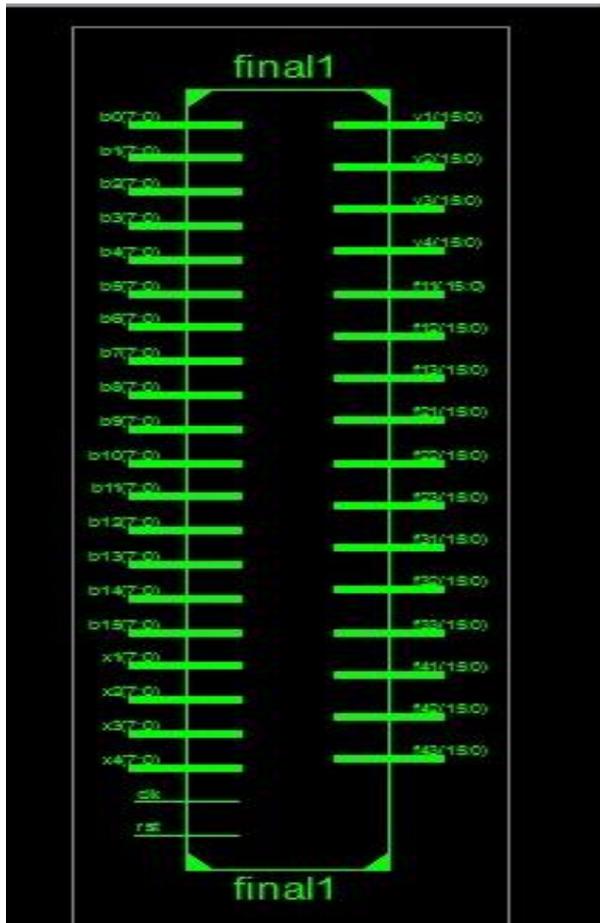


Fig.9 RTL Schematic of Proposed Scheme

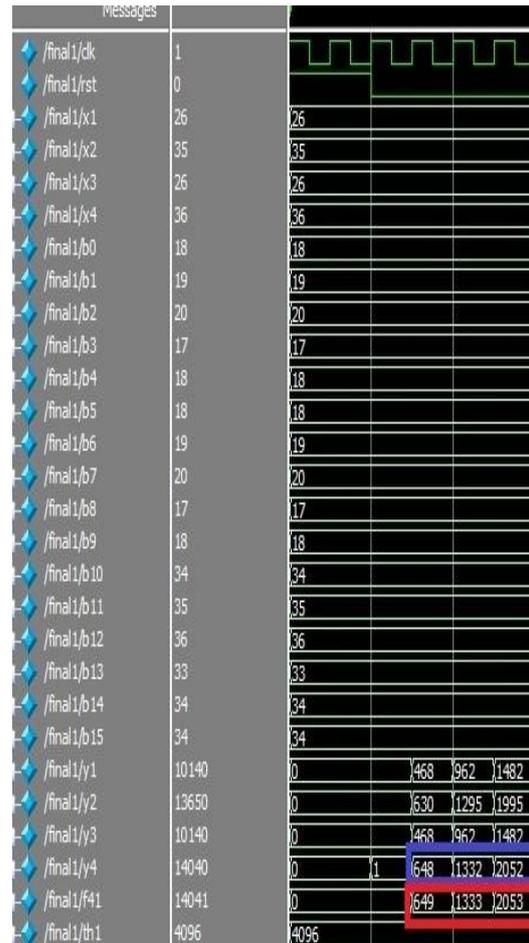


Fig. 10 Simulation Results

Table 1- Area Utilization Summary

Logic Utilization	ECC	TMR	RPR
No. of Slice Registers	714	496	362
No. of Slice LUTs	6165	3226	3163
No. of Fully used LUT-FF pairs	177	121	117

Table 2- Delay And Power Consumption

	ECC	TMR	RPR
DELAY	27.923ns	30.564ns	26.564ns
POWER CONSUMPTION	1.204W	1.240W	1.203W
No. of bonded IOBs	492	418	412

V. Conclusion

In this paper, comparison of RPR, TMR and ECC protected Parallel Filters in terms of implementation cost and effectiveness to correct the errors is done. Parallel filters that have same impulse response and with different input and outputs sequences are considered. RPR method is more suitable for multi-bit errors which will also use less area and power compared to other techniques. TMR method can also detect multi-bit errors in any filters but consumes more area and power whereas ECC scheme can detect and correct only single bit error.

In Future, utilization of IIR filters instead of FIR filters will be carried out. The extension of the scheme to parallel filters that have the same input and different impulse responses will also be done.

Acknowledgments

I would like to express my deepest gratitude to my guide Mrs. Jamuna S, Associate Professor, Dept of ECE, Dayananda Sagar Institutions, for her valuable guidance throughout this work. Without her guidance and persistent help this work would have not been possible. I am very thankful to my parents and my sister for their

immense love and trust on me throughout the journey of my life. I also thank all my M.Tech classmates & beloved friends for their perpetual support through all walks of my life.

References

- [1]. Zhen Gao, Pedro Reviriego, Wen Pan, Zhan Xu, Ming Zhao, Jing Wang, and Juan Antonio Maestro, "Fault Tolerant Parallel Filters Based on Error Correction Codes", IEEE transactions on Very Large Scale Integration (VLSI) Systems, Feb. 2014.
- [2]. R. E. Lyons, W. Vanderkulk, "The use of Triple-Modular Redundancy to improve Computer Reliability", in IBM Journal, April 1962.
- [3]. T. Hitana and A. K. Deb, "Bridging concurrent and non-concurrent error detection in FIR filters," in Proc. Norchip Conf., 2004, pp. 75–78.
- [4]. 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.
- [5]. 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.
- [6]. 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.
- [7]. P. Reviriego, S. Pontarelli, C. Bleakley, and J. A. Maestro, "Area efficient concurrent error detection and correction for parallel filters," IET Electron. Lett., vol. 48, no. 20, pp. 1258–1260, Sep. 2012.
- [8]. Pedro Reviriego, Juan A. Maestro, Isabel López and Juan A. de Agapito, "Soft Error Tolerant Infinite Impulse Response Filters using Reduced Precision Replicas", IEEE Trans. VLSI, pp. 493-496, Feb. 2011.
- [9]. S. Lin and D. J. Costello, Error Control Coding, 2nd ed. Englewood Cliffs, NJ, USA: Prentice-Hall. 2004.
- [10]. R. W. Hamming, "Error correcting and error detecting codes," Bell Syst. Tech. J., vol. 29, pp. 147–160, Apr. 1950.
- [11]. Brian Pratt, Megan Fuller, Michael Rice, Michael Wirthlin, "Reduced-Precision Redundancy for Reliable FPGA Communications Systems in High-Radiation Environments", IEEE Trans. on Aerospace and Electronic Systems vol. 49, no. 1, pp. 369-380, Jan. 2013.
- [12]. Shim, B. and Shanbhag, N., "Reduced precision redundancy for low-power digital filtering", Conference Record of the Thirty-Fifth Asilomar Conference on Signals, Systems and Computers, vol. 1, Pacific Grove, CA, 2001, pp. 148–152.
- [13]. Shim, B., Sridhara, S., and Shanbhag, N., "Reliable low-power digital signal processing via reduced precision redundancy", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 12, 5 (2004), 497–510.

Author Biographies

Mrs. Jamuna S is working as an Associate Professor, Department of E&C, Dayananda Sagar College of Engineering, Bangalore. She received her B.E degree in Instrumentation Technology from SJCE, Mysore in the year 1996 and M.Tech Degree in VLSI Design and Embedded Systems from BMSCE, Bangalore in the year 20 05. She is pursuing her Ph.d in Electronics under JNTU Hyderabad. Her research interests include VLSI design and Fault tolerant computing. Contact: js.jamuna@gmail.com



Sneha P received her Bachelor's of Engineering in the field of Electronics and Communication in the year 2013 from Bangalore Institute Of Technology, Bangalore from Visvesvaraya Technological University. She started pursuing her Masters of Technology in the field of VLSI and Embedded Systems from the year 2013-2015. She has presented a paper in a National conference and she has been qualified in GATE 2015 with a GATE score of 373. Contact: snehabha_t92@gmail.com.

