Efficient Parallel Filter Design of Coding Scheme with Lower Delay and Area Profiles

Kavita Goswami[1], Prof. Dr. A.K.Shrivastava [2]

[1]M.Tech. Scholar, [2]Guide

Department of Electronics and Communication Engineering, LNCT, Bhopal

Abstract - The way of communication is getting very rich these days and more fascinating in terms of different approach ample applications and even systems. These systems are having backbone system of signal processing and lots of encodings. All the systems having complex structures are having complex process and complex circuitry. These digital circuitries are having various elements like digital filters, modulators, error correction blocks, etc. This work focusing on designing of a efficient coding scheme for eleven parallel filter architecture which will be fault tolerant. Each of them treated as a bit, and redundant filters are as parity check bits. Which will helps us to detect and correct errors. The extended design is having better area and speed profiles than previous less number of parallel filters. The comparison is shown in the synthesis results.

Keywords - Parallel Filters, Fault Tolerant, Coding Schemes, Error Correction, FPGA.

I. INTRODUCTION

High reliability is needed in many signal processing applications to ensure continuous operation and to check the integrity of results. High reliability is needed in life critical applications, such as aircraft guidance systems or in medical equipment, where failures can jeopardize human lives, or in remote applications, such as satellites or underwater acoustic monitors, where repair is impossible or prohibitively expensive. Robustness is also needed in systems that must operate in hazardous environments, such as military equipment, or in spacecraft that must be protected against radiation. In all of these applications there is a high cost of failure, and reliability is of great importance.

The complexity of signal processing algorithms has been steadily increasing due to the availability of special purpose, high-speed signal processors. Many algorithms that were once too computationally intensive, are now implemented in real time by multiprocessor systems. In these systems, the large amount of hardware increases the likelihood of a failure occurring, and makes reliable operation difficult.

It is impossible to guarantee that components of a system will never fail. Instead, failures should be anticipated, and systems designed to tolerate failures gracefully. This design methodology is known as fault-tolerant computing and it received considerable attention by early computer designers because of the unreliability of existing components. After the development of integrated circuits, which were several orders of magnitude more reliable, fault-tolerance became a secondary issue. Attention was focused on developing faster, more complex circuits, and as a result, circuit densities grew exponentially. In many areas, however, semiconductor reliability has not kept pace with the level of integration, and fault-tolerance is becoming a major issue again.

A component is said to have failed when it does not compute the correct output, given its input. A failure is caused by a physical fault, and the manifestation of a failure is errors within the system [1]. In a fault-tolerant system, the basic idea is to tolerate internal errors, and keep them from reaching and corrupting the output. This process is known as error masking.

A fault may be permanent or transient. A permanent fault is caused by a static flaw in a component and may result from manufacturing defects or physical damage during operation. A permanent fault need not produce errors for every input. Transient or soft faults are momentary faults that occur infrequently and are randomly distributed throughout the system. These types of faults can be induced by several sources: alpha particles emitted from the substrate and packaging material; cross coupling between closely spaced signal lines; unpredictable timing glitches from rare, absolute worst case delays; and electromigration phenomenon in small conductors [2]. The current trends toward higher clock speeds and denser components aggravate the problem of transient errors.

The highest and most desirable level of fault-tolerance is known as concurrent error masking. In this technique, errors are masked during operation and the system continues to function with no visible degradation in performance. In general, concurrent error masking usually requires the following steps:

- Fault Detection - determine that the output is invalid and that a fault has occurred within the system.
- Fault Location - determine which system component failed.
- Fault Correction - determine the correct output.

Concurrent error masking is difficult and expensive to achieve, and basically requires the entire system to be checked for errors at each time step. Often, lower levels of protection are acceptable and a few erroneous outputs may be tolerated. In these instances, less expensive techniques which check only a part of the system at each time step are adequate. In other applications, there is a high cost for computing erroneous results and a lower cost for delaying the result.

II. FIR FILTER

"FIR" means "Finite Impulse Response". If theres put in an impulse, that is, a single "1" sample followed by many "0" samples, zeroes will come out after the "1" sample has made its way through the delay line of the filter In the common case, the impulse response is finite because there is no feedback in the FIR. A lack of feedback guarantees that the impulse response will be finite. Therefore, the term "finite impulse response" is nearly synonymous with "no feedback".

However, if feedback is employed yet the impulse response is finite, the filter still is a FIR. An example is the moving average filter, in which the nth prior sample is subtracted (fed back) each time a new sample comes in. This filter has a finite impulse response even though it uses feedback: after N samples of an impulse, the output will always be zero.

The Lth-order LTI FIR filter is graphically interpreted in Fig . It can be seen to consist of a collection of a “tapped delay line,” adders, and multipliers. One of the operands presented to each multiplier is an FIR coefficient, often referred to as a “tap weight” for obvious reasons. Historically, the FIR filter is also known by the name “transversal filter,” suggesting its “tapped delay line” structure.

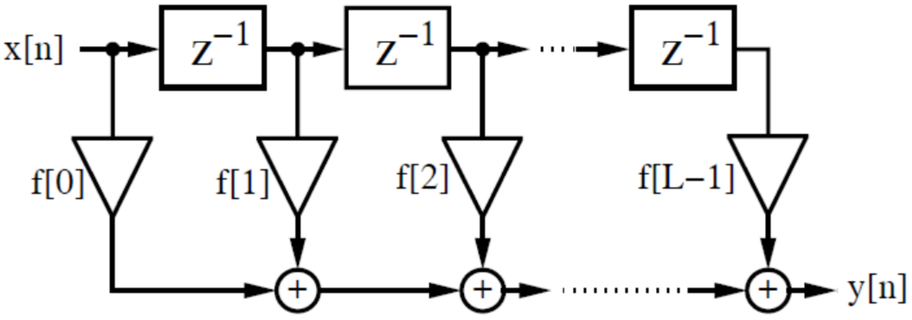


Figure 2.1 Direct form of FIR filter.

A variation of the direct FIR model is called the transposed FIR filter. It can be constructed from the FIR filter in Fig.

A transposed FIR filter is shown in Fig. 2 and is, in general, the preferred implementation of an FIR filter. The benefit of this filter is that there is no need for an extra shift register for x[n], and there is no need for an extra pipeline stage for the adder (tree) of the products to achieve high throughput.

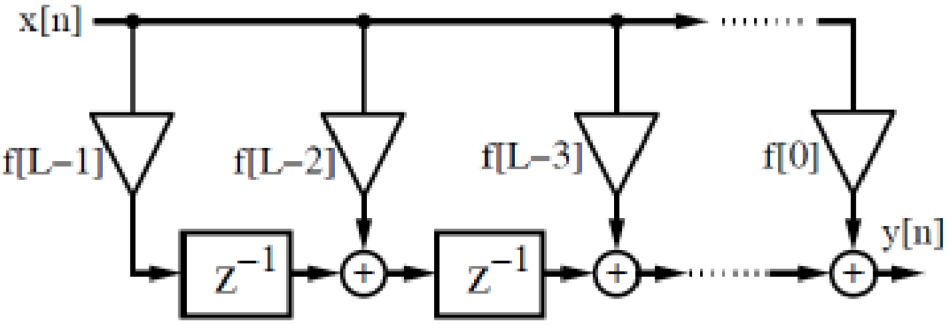


Figure 2.2 Transposed form of FIR filter.

III. PROPOSED DESIGN

An efficient coding scheme for eleven parallel filter has been proposed in this work illustrated in figure 3.1 General Form of Proposed Coding Scheme. There are 4X6 coding matrix having input vector X_1, X_2, X_3, X_4, and output vectors y_1, y_2, y_3, y_4, Two modules are there original modules and redundant modules are designed to work parallel. A general form of coding scheme of proposed work has given in figure 3.2. A redundant module generate a redundant signal Z_5 and Z_6 for input signal Z_1, Z_2,Z_3,Z_4.

The expression for redundant signal can be expressed as

$$X_5 = a_{51}X_1 + a_{52}X_2 + a_{53}X_3 + a_{54}X_4 \dots\dots\dots\dots(1)$$

$$X_6 = a_{61}X_1 + a_{62}X_2 + a_{63}X_3 + a_{64}X_4 \dots\dots\dots(2)$$

Figure 3.3 has given the RTL schematic of proposed Efficient Parallel Filter Design Coding Scheme with Lower Delay and Area Profiles

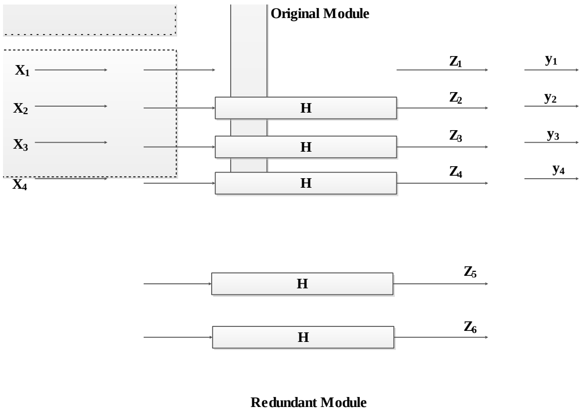


Fig.3.1 General Form of Proposed Coding Scheme.

It can be examined that from figure 3.1 that the number increases with the logarithm in construct two in light of the quantity of filters. Along these lines, the cost is considerably smaller.

The coding of the redundant filters proposed in this work are basically depends simple additions operation that replace the XOR binary operations in traditional ECCs. However, since both the data sources and yields of the filters are arrangements of numbers, a more general coding can be used. This type of coding has been explored for linear time-invariant systems refer to Eq. no (1) and Eq. no(2),

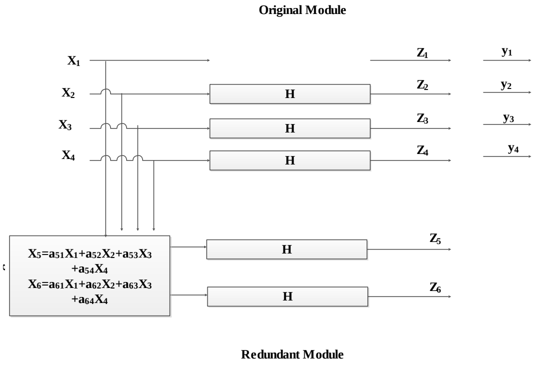


Fig.3.2 Practical coding scheme to protect four parallel filters.

The proposed plot is outlined in Fig. 3.2 Practical coding plan to ensure four parallel filters. The input signals are encoded utilizing a matrix with arbitrary coefficients to generate the signals that enter the four original and two redundant filters the redundant signal is X_5 and X_6.

The error correction and detection logic can be simplified assuming that there is only a single error.

The RTL schematic of proposed work has been illustrated n figure 3.3. Obtained from the synthesis of proposed design on Xilinx 13.1 ISE design suite in device family Vertex 4 FPGA.

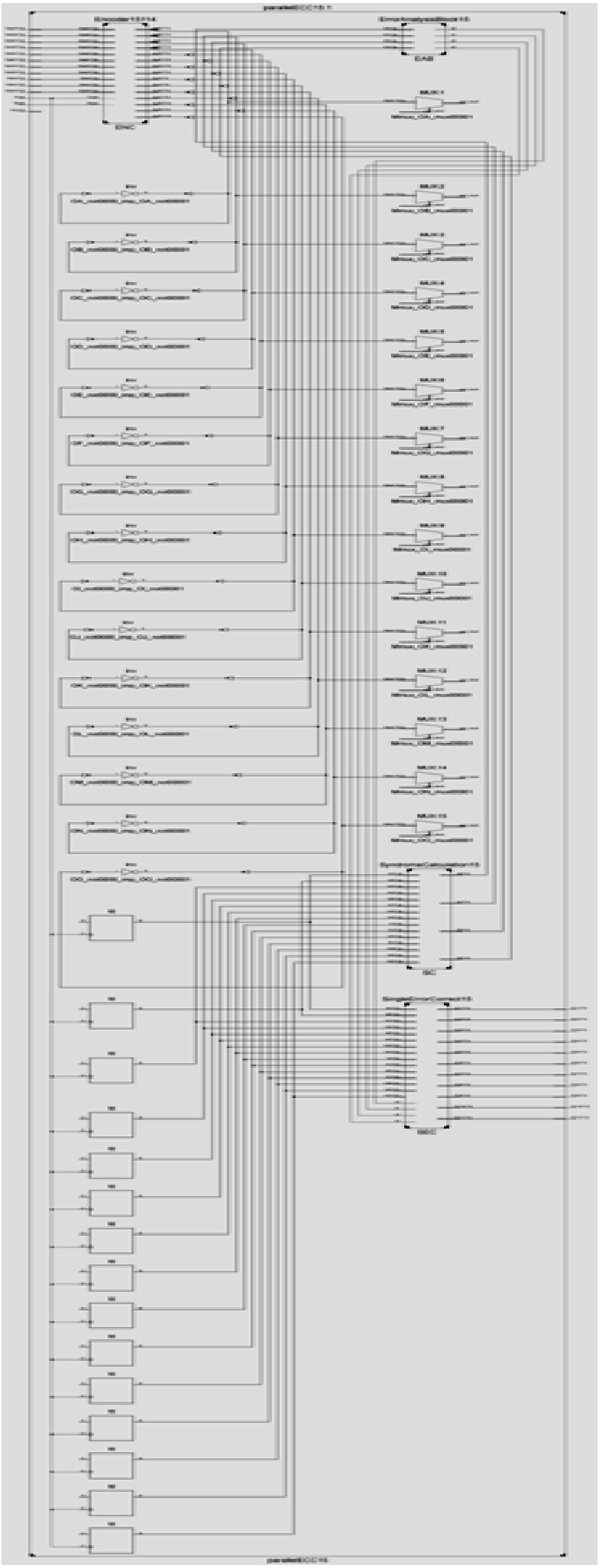
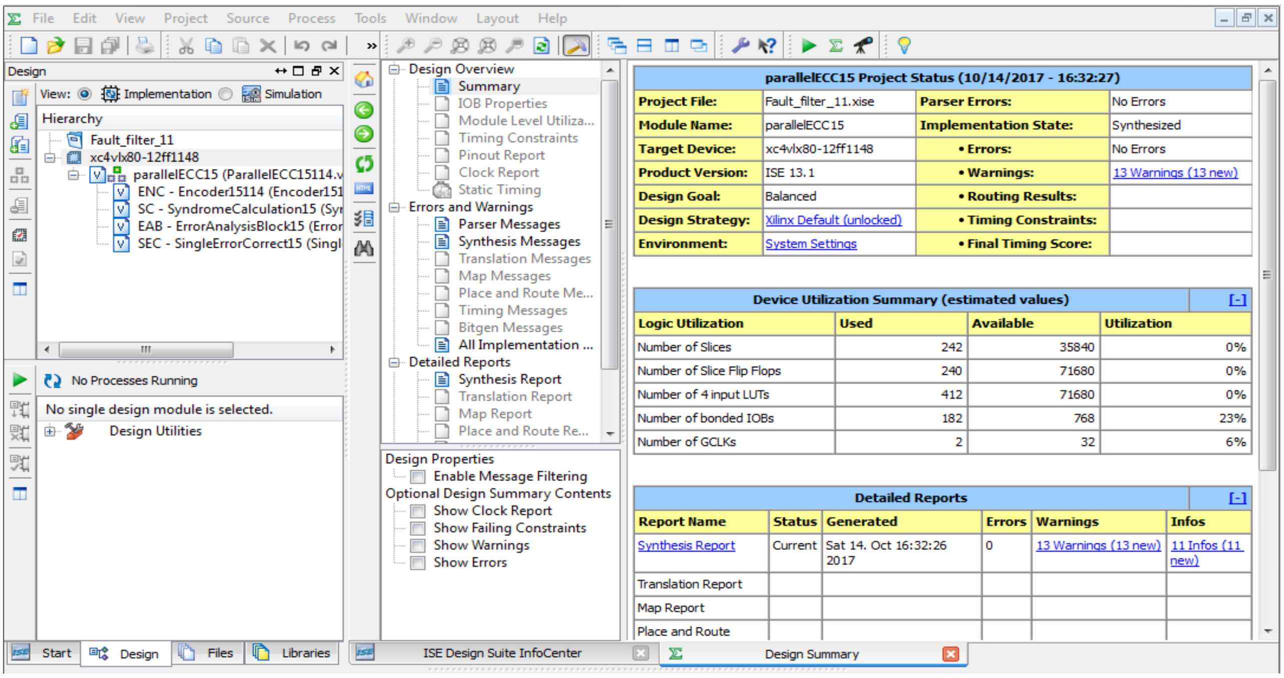


Fig.3.3 RTL View of the Design

IV. SYNTHESIS OUTCOMES

Proposed Design of an efficient coding scheme of parallel filter has been implemented and synthesised in Xilinx ISE design suite 1.3.1 using FPGA vetex 4 XC4VLX80 device family. And Simulated using Isin Simulator. The Synthesis screen of proposed design has been given in figure 4.1. It is clear from device utilization summary under synthesis report that the proposed design has batter performance as compared to existing work with respect to area calculated in number of slices flip-flops and IO buffs. The comparison between proposed work and existing work has given in table 4.1.



parallelECC15 Project Status (10/14/2017 - 16:32:27)			
Project File:	Fault_filter_11.xise	**Parser Errors:**	No Errors
Module Name:	parallelECC15	**Implementation State:**	Synthesized
Target Device:	xc4vlx80-12ff1148	**• Errors:**	No Errors
Product Version:	ISE 13.1	**• Warnings:**	13 Warnings (13 new)
Design Goal:	Balanced	**• Routing Results:**	
Design Strategy:	Xilinx Default (unlocked)	**• Timing Constraints:**	
Environment:	System Settings	**• Final Timing Score:**	

Device Utilization Summary (estimated values) [-]

Logic Utilization	Used	Available	Utilization
Number of Slices	242	35840	0%
Number of Slice Flip Flops	240	71680	0%
Number of 4 input LUTs	412	71680	0%
Number of bonded IOBs	182	768	23%
Number of GCLKs	2	32	6%

Detailed Reports [-]

Report Name	Status	Generated	Errors	Warnings	Infos
Synthesis Report	Current	Sat 14. Oct 16:32:26 2017	0	13 Warnings (13 new)	11 Infos (11 new)
Translation Report					
Map Report					
Place and Route					

Figure 4.1 Synthesis Screen of Proposed Work.

Table:4.1 Comparison of with Previous Work

Parameters	Previous Work with 8-Filters [1]	Proposed Work
Slices	9872	242
Flip Flops	3994	240
LUTs	19136	412

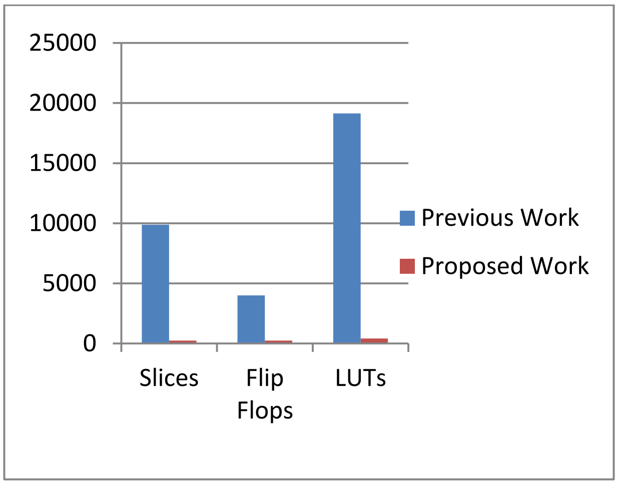


Figure 4.1 Comparison plot of proposed work vs previous work.

V. CONCLUSION AND FUTURE SCOPES

Proposed filter design presented in this work have been implemented in HDL and mapped standard verification environment of the FPGA in Virtex 4 XC4VLX80 device family. The design of a efficient coding scheme for eleven parallel filter architecture which is fault tolerant is suitable for applications that require low power consumption and a occupied small silicon area. The main advantage of the design is that they can be implemented in any FPGA; meaning that they are not dependent on the platform. The comparison of result evaluated that the proposed design have better device utilization as compared to existing design. The extended design is having better area and speed profiles than previous less number of parallel filters. There's a lot work to be done in the field of design parallel filter for fault tolerant computation with power management and delay management as well as area optimization and filtering level enhancement.

REFERENCES

[1] Z. Gao, P. Reviriego, Z. Xu, X. Su, J. Wang and J. A. Maestro, "Efficient Coding Schemes for Fault-Tolerant Parallel Filters," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 62, no. 7, pp. 666-670, July 2015.

[2] Z. Gao et al., "Fault Tolerant Parallel Filters Based on Error Correction Codes," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 23, no. 2, pp. 384-387, Feb. 2015.

[3] K. B. Muralidharan, G. S. Kumar and M. Bhasi, "Fault tolerant state management for high-volume low-latency data stream workloads," 2014 International Conference on Data Science & Engineering (ICDSE), Kochi, 2014, pp. 24-27.

[4] N. Madhuri, S. R. Doradla and M. Surya kalavathi, "Fault tolerant control of Hybrid power filter," 2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies, Ramanathapuram, 2014, pp. 235-239.

[5] J. Park, J. Park and S. Bhunia, "VL-ECC: Variable Data-Length Error Correction Code for Embedded Memory in DSP Applications," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 61, no. 2, pp. 120-124, Feb. 2014.

[6] A. Asuvaran and S. Senthilkumar, "Low delay error correction codes to correct stuck-at defects and soft errors," 2014 International Conference on Advances in Engineering and Technology (ICAET), Nagapattinam, 2014, pp. 1-7.

[7] P. Reviriego, S. Pontarelli, C. J. Bleakley and J. A. Maestro, "Area efficient concurrent error detection and correction for parallel filters," in Electronics Letters, vol. 48, no. 20, pp. 1258-1260, September 27 2012.

[8] P. P. Vaidyanathan, Multirate Systems and Filter Banks, Englewood Cliffs, N.J., USA: Prentice Hall, 1993.

[9] A. Sibille, C. Oestges and A. Zanella, MIMO: From Theory to Implementation, New York, NY, USA: Academic, 2010.

[10] N. Kanekawa, E. H. Ibe, T. Suga and Y. Uematsu, Dependability in Electronic Systems: Mitigation of Hardware Failures, Soft Errors, and Electro-Magnetic Disturbances, New York, NY, USA: Springer Verlag, 2010.

[11] M. Nicolaidis, “Design for soft error mitigation,” IEEE Trans. Device Mater. Rel., vol. 5, no. 3, pp. 405-418, Sep. 2005.

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

[13] A. Reddy and P. Banarjee “Algorithm-based fault detection for sig¬nal processing applications,” IEEE Trans. Comput., vol. 39, no. 10, pp. 1304-1308, Oct. 1990.

[14] S. Pontarelli, G. C. Cardarilli, M. Re, and A. Salsano, “Totally fault toler¬ant RNS based FIR filters,” in Proc. IEEEIOLTS, 2008, pp. 192-194.

[15] 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. IEEEIOLTS, 2012, pp. 130-133.

[16] B. Shim and N. Shanbhag, “Energy-efficient soft error-tolerant digital signal processing,” IEEE Trans. Very Large Scale Integr. Syst., vol. 14, no. 4, pp. 336-348, Apr. 2006.