PARALLEL FILTER USING ERROR CORRECTION CODES IN MODERN SIGNAL PROCESSING CIRCUITS

Sulthan Alavudeen.S [1], Samuvel Paul Raj [2], Ajith Kumar [3]

[1,2,3] U.G Student, Department of Electronics and Communication Engineering,
Raja College of Engineering and Technology, Madurai, Tamilnadu, (India)

ABSTRACT

Digital filters are widely used in signal processing and communication systems. In some cases, the reliability of those systems is critical, and fault tolerant filter implementations are needed. Next many techniques that exploit the filters' structure and properties to achieve fault tolerance have been proposed. As technology scales, it enables more complex systems that incorporate many filters. In those complex systems, it is common that some of the filters operate in parallel, for example, by applying the same filter to different input signals. Recently, a simple technique that exploits the presence of parallel filters to achieve fault tolerance has been presented. In this brief, that idea is generalized to show that parallel filters can be protected using error correction codes (ECCs) in which each filter is the equivalent of a bit in a traditional ECC. This new scheme allows more efficient protection when the number of parallel filters is large. The proposed scheme coded in HDL and simulated using xilinx 12.1 e.

Keywords: Error correction codes (ECCs), filters, and soft errors.

I. INTRODUCTION

Electronic circuits are increasingly present in automotive, medical, and space applications where reliability is critical. In those applications, the circuits have to provide some degree of fault tolerance. This need is further increased by the intrinsic reliability challenges of advanced CMOS technologies that include, e.g., manufacturing variations and soft errors. A number of techniques can be used to protect a circuit from errors. Those range from modifications in the manufacturing process of the circuits to reduce the number of errors to adding redundancy at the logic or system level to ensure that errors do not affect the system functionality.

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

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

When the response h[l] is nonzero, only for a finite number of samples, the filter is known as a FIR filter, otherwise the filter is an infinite impulse response (IIR) filter impulse response of the filter

II.PROPOSED SYSTEM

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 [14] with $k = 4$ and $n = 7$. In this case, the three parity check bits $p1$, $p2$, $p3$ are computed as a function of the data bits $d1$, $d2$, $d3$, $d4$ as follows:

$$p1 = d1 \oplus d2 \oplus d3$$

$$p2 = d1 \oplus d2 \oplus d4$$

$$p3 = d1 \oplus d3 \oplus d4.$$

- Based on the use of the ECCs
- Error Correction Codes (ECCs) using each of the filter outputs
- It can correct failures in multiples modules
- *In The Parity, the error is detected and corrected by using,*
    - $yc1[n] = z1[n] - y2[n] - y3[n]$

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.

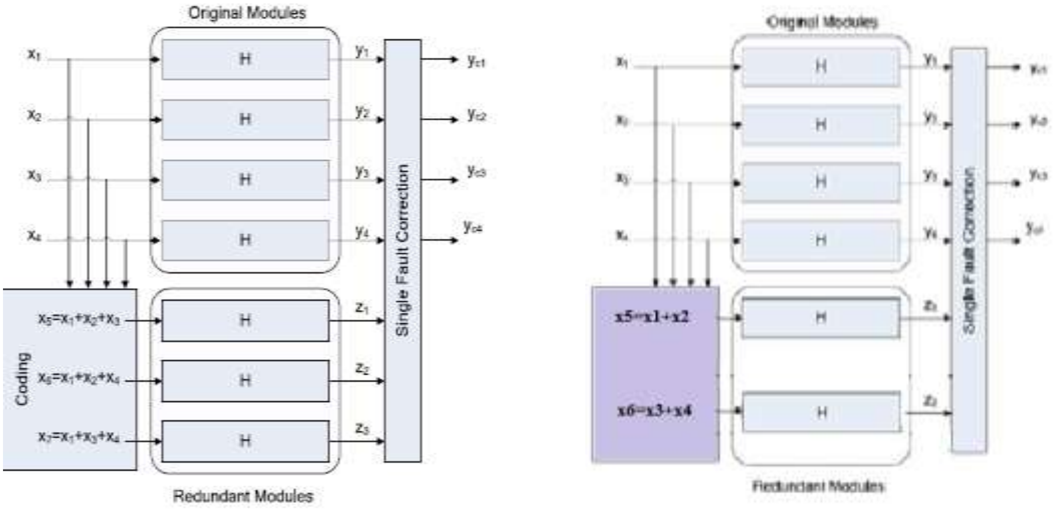


Figure 1: Block Diagram of Proposed Work

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

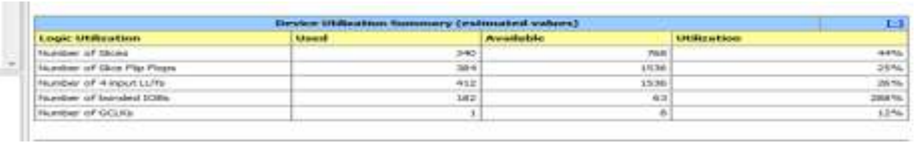
$$z1[n] = y1[n] + y2[n] + y3[n]$$

$$z2[n] = y1[n] + y2[n] + y4[n]$$

$$z3[n] = y1[n] + y3[n] + y4[n]$$

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.

III. RESULTS AND DISCUSSION



Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	340	768	44%
Number of Slice Flip Flops	384	1536	25%
Number of 4 input LUTs	412	1536	26%
Number of bonded IOBs	182	63	288%
Number of GCLKs	1	8	12%

Figure 2: Design Summary

S.NO	PARAMETER	EXISTING METHOD	PROPOSED
1	NUMBER OF SLICES	428	340
2	IOB'S	208	182
3	LUT'S	558	412

Figure 3: Performance Analysis of Proposed Work

Figure 4: Simulation of Proposed Work

IV. 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] 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.

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

[3] 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.*

[4] 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.

[5] 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.*