Chandragupta Thatta* et al.
(IJITR) INTERNATIONAL JOURNAL OF INNOVATIVE TECHNOLOGY AND RESEARCH
Volume No.5, Issue No.2, February – March 2017, 5621-5625.

Low Area And Delay Implementation Of Error Correcting And Error Detecting Code Using Reversible Gate

CHANDRAGUPTA THATTA
Student
Sri Vaishnavi College of Engineering
(Approved by AICTE & Affiliated to JNTU, Kakinada), Singupuram (Vill & post), Srikakulam (Rural) Mandal.

K.V.R.L.PRASAD
Asst. Professor
Sri Vaishnavi College of Engineering
(Approved by AICTE & Affiliated to JNTU, Kakinada), Singupuram (Vill & post), Srikakulam (Rural) Mandal.

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. Over the years, 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 technique is evaluated using a case study of parallel finite impulse response filters showing the effectiveness in terms of protection and implementation cost.**

Index Terms— **Parallel Filters, Error Correction Codes (ECCS), Reversible Logic Gates, VERILOG.**

I. INTRODUCTION

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

To add redundancy, a general technique known as a triple modular redundancy (TMR) can be used. The TMR, which triplicates the design and adds voting logic to correct errors, is commonly used. However, it more than the triples area and power of the circuit, something that may not be acceptable in some of the applications. When the circuit to be protected has algorithmic or structural of the properties, a better option can be to exploit those properties to implement of the fault tolerance.

One example is signal processing circuits for which the specific some techniques have been proposed over the years. Digital filters are one of the most commonly used in the signal processing circuits and several techniques have been proposed to protect them from errors. For example, the use of reduced precision replicas was proposed to reduce the cost of implementing modular redundancy in FIR filters. Other schemes have exploited the FIR properties at a word level to also achieve fault tolerance. A relationship between the memory elements of an FIR filter and the sequence of input was used to detect errors.

We consider where two parallel filters with the same response that processed different input signals. However, it is increasingly common to find systems in which the several filters operate in parallel. This is the case in filter banks and in many of the modern communication systems. For those systems, the protection of filters can be addressed at a higher level by considering the parallel filters as the block to be protected. This idea was explored. It was shown that with only copy of the one redundant, single error correction can implemented. Therefore, a significant reduction of cost compared with TMR was obtained.

The use of number systems of residue and arithmetic codes has also been proposed to protect filters. Finally, the FIR filters are used to different implementation structures to correct errors with only one of the redundant module has also been proposed. In all the techniques mentioned so far, the protection of a single filter is considered.

In this brief, a general scheme to protect parallel filters is presented. Parallel filters with the same response that process of different input signals are considered. The new approach is based on the

application of error correction codes (ECCs) using each outputs of the filter as the equivalent of bit in and ECC codeword. This is a generalization of the scheme is presented and enables the more efficient implementations when the number of parallel filters is large not only small. The scheme can also be used to provide powerful protection is more using advanced ECCs that can correct failures in the multiples modules.

II. PARALLEL FILTERS WITH THE SAME RESPONSE

A discrete time filter implements the following equation:

$$y[n] = x[n-l] \cdot h[l] \quad \text{for } l = 0 \text{ to } \infty$$

Where $x[n]$ is the input signal, the output is $y[n]$, and $h[l]$ is the impulse response of the filter. When the response of $h[l]$ is nonzero, only a finite number of samples, the filter is known as a FIR filter, otherwise the filter is an infinite impulse response (IIR) filter. There are many types of several structures to the implement both IIR and FIR filters. In data acquisition and processing applications is also used to filter several signals with the same response In the following, the k parallel filters with the same response and different input signals are considered. These parallel filters are illustrated in Fig.1. This some of filter is found in some communication systems so that use several channels in parallel.

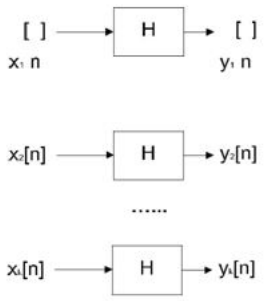


Fig.1. Parallel filters with the same response.

TABLE I
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

Original Modules

An interesting of property for these parallel filters is that the sum of any combination of the outputs yi $[n]$ can also be obtained by adding the corresponding inputs xi $[n]$ and filtering the resulting signal with the same filter $h[l]$. For example

$$y_1[n] + y_2[n] = \sum_{l=0}^{\infty} (x_1[n-l] + x_2[n-l]) \cdot h[l]. \qquad (2)$$

This observation will be used in the following steps to develop the proposed fault tolerant implementation

III. HAMMING CODE

The new technique system is based on the use of the ECCs (Hamming code) with reversible logic. A simple ECC takes a block of k bits and produces a block of n bits by adding $n - k$ parity check bits [13]. The parity check bits are XOR gate combinations of the k data bits. By most properly designing those combinations it is possible to detect errors and correct errors. For an example, let us consider a simple Hamming code [14] with $k = 4$ and $n = 7$. In this case, the three parity check bits are p_1, p_2, p_3 is computed as a function of the data bits are d_1, d_2, d_3, d_4 as follows:

$$p_1 = d_1 \oplus d_2 \oplus d_3$$

$$p_2 = d_1 \oplus d_2 \oplus d_4$$

$$p_3 = d_1 \oplus d_3 \oplus d_4.$$

The data of bits 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 recompiling the parity check bits and comparing the results with the values stored. An error on d_1 will cause errors on the three parity bit checks; an error on d_2 only in p_1 and p_2; an error on d_3 in p_1 and p_3; and finally an error on d_4 in p_2 and p_3. Therefore, the information of data bit in error can be located and the error can be corrected. This is formulated in terms of the generating is denoted by G and parity check H matrixes. For the Hamming code considered in the example, those are

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

$$H = \begin{matrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{matrix}.$$

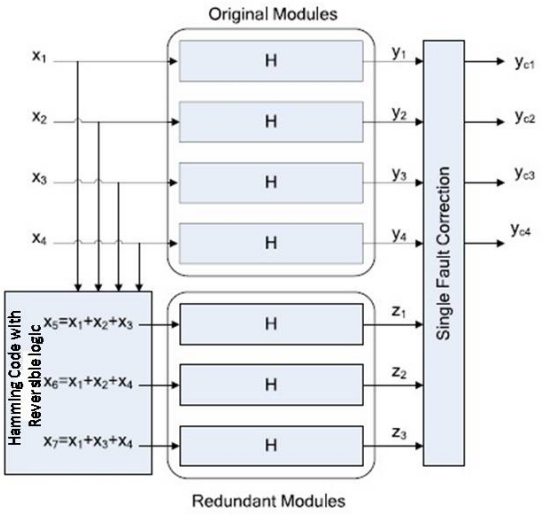


Fig. 2. Proposed scheme for four filters using Hamming code and Reversible logic

Encoding is done by computing $y = x \bullet G$ and the error detection is done by computing $s = y \bullet H^T$, where the operator • is based on the module of two addition (XOR) gates and multiplication. Correction is done using the vector *s*, known as syndrome, to identify the data bit in error. The correspondence of values of *s* to error position is captured in Table I. Once the erroneous data bit is identified, it is corrected by the simply inverting the bit. 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 y_1, y_2, y_3, y_4 and the Hamming code, the check filters would be

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

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

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

And the checking is done by testing if

$$z_1[n] = y_1[n] + y_2[n] + y_3[n]$$

$$z_2[n] = y_1[n] + y_2[n] + y_4[n]$$

$$z_3[n] = y_1[n] + y_3[n] + y_4[n].$$

For example, an error on the filter y_1 will cause errors on the checks of z_1, z_2, and z_3. Similarly, errors on the other filters will cause errors on a different group of z_i. Therefore, as with the traditional ECCs, the error can be detected and corrected.

The overall scheme is illustrated on Fig. 2. It can be observed that correction is achieved with only three redundant data of filters. For the filters, correction of the data is achieved by reconstructing the erroneous outputs using the remaining of the data and check outputs. For example, when an error on y_1 is detected, it can be corrected by making

$$y_{c1}[n] = z_1[n] - y_2[n] - y_3[n].$$

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

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & -1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & -1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & -1 \end{bmatrix}$$

and calculate the value of $s = y\,H^T$ to detect errors. Then, the vector *s* is also used to particularly identify the filter in error. In our case, a nonzero value in vector *s* is equivalent to one (1) in the traditional Hamming code. A zero value in the check corresponds to a zero in the traditional Hamming code.

It is an important to note that due to different finite precision effects in the original and check filter implementations, the comparisons in the (7) can show the small differences. Those differences will depend on the quantization effects in the filter implementations that have been studied in widely for different filter structures. [12]. Therefore, a threshold value must be used in the comparisons so that values smaller than the threshold values are classified as 0. This means that small number of errors may not be corrected. 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 in [3].

With this formulation of alternative, it is clear that the scheme can be used for any number of parallel filters and any linear block of the data 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 codes. This is the same as for the traditional of ECCs for which the overhead decreases as the block size increases [13].

IV. CASE STUDY

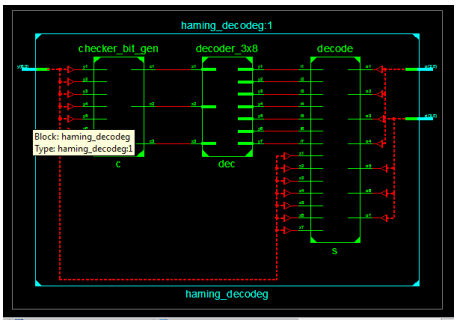
To evaluate the effectiveness of the proposed scheme, a case study is used. A set of parallel FIR filters with 16 coefficients is considered. The input data and coefficients are quantized with 8 bits. The filter output is quantized with 18 bits. For the check filters z_i, since the input is the sum of several inputs x_j, the input bit-width is extended to 10 bits. A small threshold is used in the comparisons such that errors smaller than the threshold are not considered errors. As explained in Section III, no logic sharing was used in the computations in the

encoder and decoder logic to avoid errors on them from propagating to the output.

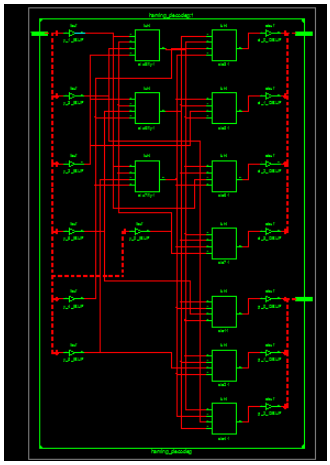
Two configurations are considered. The first one is a block of four parallel filters for which a Hamming code with $k = 4$ and $n = 7$ is used. The second is a block of eleven parallel filters for which a Hamming code with $k = 11$ and $n = 15$ is used.

V. RESULTS

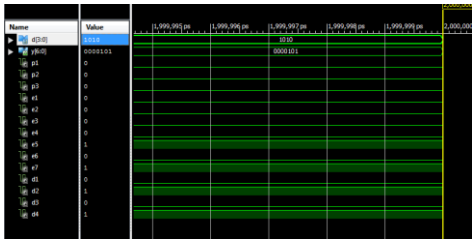
RTL schematic



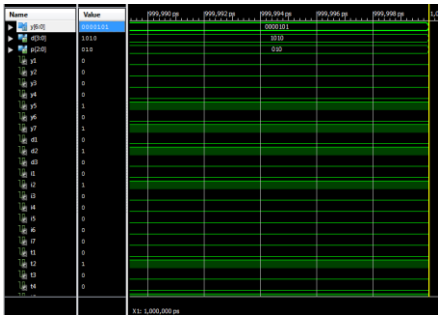
Technological schematic



Encoding wave form



Decoding wave form



Comparison table

	Number of LUT's	Power(w)	Delay(ns)
Proposed encoding	2	0.152	5.870
Proposed decoding	10	0.76	7.213
Existing encoding	2	0.152	5.870
Existing decoding	18	1.36	4.044

VI. CONCLUSION

This brief has presented a new type of scheme to protect the parallel filters that are commonly found in modern signal processing circuits. The scheme can be used for parallel filters that have the same response and process input signals are different. A case study has also been discussed to show the effectiveness of the scheme in terms of error correction and circuit overheads. The technique provides larger benefits when the number of parallel filters is large. The proposed scheme can also be applied to the IIR filters. The proposed scheme can also be combined with the reduced precision replica approach presented in [3] to reduce the required for protection. Future work will consider the evaluation of the benefits of the proposed technique for IIR filters. The extension of the scheme to parallel filters that have the same input and different impulse responses is also a topic for future work. This will be of interest when the number of parallel filters is less as the cost of the proposed scheme is larger in that case. The functionality is verified using ISE simulator and the synthesis is carried out using XILINX synthesizer by developing the RTL using VERILOG HDL.

VII. REFERENCES

[1]. M. Nicolaidis, "Design for the soft error mitigation," *IEEE Trans. Device Mater. Rel.*, vol. 5, no. 3, pp. 405–418, Sep. 2005.

[2]. A. Reddy "An Algorithm-based error detection for signal processing applications," *IEEE Trans. Comput.*, vol. 39, no. 10, pp. 1304–1308, Oct. 1990.

[3]. B. Shim and N. Shanbhag, "The Energy-efficient of soft error-tolerant digital signal processing," *IEEE Trans. Very Large Scale Integration. (VLSI) Syst.*, vol. 14, no. 4, pp. 336–348, Apr. 2006.

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

[5]. Y.-H. Huang, "High-efficiency soft-error-tolerant digital signal processing using fine-grain subword- process of error detection,"

IEEE Trans. Very Large Scale Integation. (VLSI) Syst., vol. 18, no. 2, pp. 291–304, Feb. 2010.

[6]. S. Pontarelli, G. C. Cardarilli, M. Re, and A. Salsano, "Totally fault tolerant RNS based on the FIR filters," in *Proc. IEEE IOLTS*, Jul. 2008, pp. 192–194.

[7]. Z. Gao, W. Yang, X. Chen, M. Zhao, and J. Wang, "Fault missing rate analysis of arithmetic residue data codes based fault-tolerant FIR filter design," in *Proc. IEEE IOLTS*, Jun. 2012, pp. 130–133.

[8]. Reviriego, C. J. Bleakley, and J. A. Maestro, "Strutural DMR: Technique for implementation of soft-error-tolerant FIR filters," *IEEE Trans. Circuits Syst., Exp. Briefs*, vol. 58, no. 8, pp. 512–516, Aug. 2011. *Multirate Systems and Filter Banks*. Upper Saddle River, NJ, USA: Prentice-Hall, 1993.

[9]. A. Sibille, C. Oestges, and A. Zanella, *MIMO: From the Theory to Implementation*. San Francisco, CA, USA: Academic Press, 2010.

[10]. A.V.Oppenheim and R. W. Schafer, *Discrete the Time Signal Processing*. Upper Saddle River, NJ, USA: Prentice-Hall 1999.

[11]. S. Lin and D. J. Costello, *The Error Control Coding*, 2nd ed. Englewood Cliffs, NJ, USA: Prentice-Hall. 2004.

[12]. By R. W. Hamming, "Error correcting and error detecting codes," *Bell Syst. Tech. J.*, vol. 29, pp. 147–160, Apr. 1950

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

[14]. Landauer, R., "Irreversibility and heat generation in the computing process", IBM J. Research and Development, 5(3): pp. 183-191, 1961.

[15]. Bennett, C.H., "Logical reversibility of Computation", IBM J.Research and Development, 17: pp. 525-532, 1973.

[16]. Thapliyal H, M. B.Sshrinivas." A New Reversible TSG Gate and Its Application for Designing Efficient Adder Circuits". Centre for VLSI and Embedded System Technologies International Institute of Information Technology, Hyderabad, 500019, India

[17]. Thapliyal H, M. B.Sshrinivas "Novel Reversible Multiplier Architecture Using Reversible TSG Gate" Computer Systems and Applications, 2006.

[18]. H.r.bhagyalakshmi, M.k.venkatesha," an improved design of a multiplier using reversible Logic gates" International Journal of Engineering Science and Technology Vol. 2(8), 2010, 3838-3845

[19]. Rangaraju H G, Venugopal U, Muralidhara K N, Raja K B "Low Power Reversible Parallel Binary Adder/Subtractor"International journal of VLSI design & Communication Systems (VLSICS) Vol.1, No.3, September 2010

[20]. Thapliyal H, Ranganathan N.," Design of Reversible Latches Optimized for Quantum Cost, Delay and Garbage Outputs" Centre for VLSI and Embedded System Technologies International Institute of Information Technology, Hyderabad, 500019, India.

[21]. V.Rajmohan, V.Ranganathan,"Design of counter using reversible logic" 978-1-4244-8679-3/11/$26.00 ©2011 IEEE.

[22]. Abu Sadat md. Sayem, masashi ueda," Optimization of reversible sequential circuits" Journal of computing, Vol. 2, No. 6, June 2010, NY, USA, ISSN 2151-9617.

[23]. Siva Kumar Sastry, Hari Shyam Shroff ,Sk.Noor Mahammad, V. Kamakoti" Efficient Building Blocks for Reversible Sequential Circuit Design" 1-4244-0173-9/06/$20.00©2006IEEE.

