IJSEAT International Journal of Science Engineering and Advance Technology

Protection of "Fault Tolerant Parallel Filters" by Hamming code with Reversible logic

Sabbi Suryakala[1], M.Nagendra Kumar[2]
M.Tech(student)[1], HOD & Associate professor[2]
[1]VLSI & Embedded systems,[1,2]Dept.of Electronics and Communication Engineering
Kakinada Institute of Engineering and Technology for Women, Korangi,AP, INDIA

***Abstract*—** Advanced channels are generally utilized as a part of flag preparing and correspondence frameworks. Now and again, the dependability of those frameworks is basic, and blame tolerant channel executions are required. Throughout the years, numerous methods that endeavor the channels' structure and properties to accomplish adaptation to internal failure have been proposed. As innovation scales, it empowers more intricate frameworks that join many channels. In those unpredictable frameworks, it is regular that a portion of the channels work in parallel, for instance, by applying a similar channel to various info signals. . The complexity occurs while decoding the received encoded data. More often the transmitted data is subjected to the channel noise which influences the original signal. To overcome this problem many error correction codes (ECC's) are introduced.Recently, a simple technique that exploits the presence of parallel filters to achieve fault tolerance has been presented In this paper we proposed an error detection and correction code called hamming code. The hamming code not only detects the errors as conventional codes but also it is able to correct the data. In addition the process is supported with reversible gate logic. This is the updated design methodology to reduce the power consumption and complexity. Reversible processing will likewise prompt to change in vitality productivity. Vitality effectiveness will on a very basic level influence the speed of circuits, for example, nano-circuits and in this way the speed of most figuring applications. To build the compactness of gadgets again reversible registering is required. This thought is summed up to demonstrate that parallel channels can be secured utilizing mistake revision codes (ECCs) in which every channel is what might as well be called a bit in a customary ECC. This new plan permits more productive security when the quantity of parallel channels is expansive. The system is assessed utilizing a contextual investigation of parallel limited motivation reaction channels demonstrating the viability as far as security and execution cost.

***Index Terms*—** Parallel filters, Error correction codes (ECCs) , Reversible logic gates.

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 adjustments in the assembling procedure of the circuits to decrease the quantity of mistakes to including excess at the rationale or framework level to guarantee that blunders don't influence the framework usefulness [1]. To include excess, a general strategy known as triple secluded repetition (TMR) can be utilized. The TMR, which triplicates the outline and adds voting rationale to right blunders, is regularly utilized. Nonetheless, it dramatically multiplies the territory and force of the circuit, something that may not be adequate in a few applications. At the point when the circuit to be secured has algorithmic or basic properties, a superior alternative can be to misuse those properties to execute adaptation to non-critical failure. One case is flag preparing circuits for which particular procedures have been proposed throughout the years [2].

Advanced channels are a standout amongst the most normally utilized flag preparing circuits and a few systems have been proposed to shield them from blunders. A large portion of them have concentrated on limited drive reaction (FIR) channels. For instance, in [3], the utilization of decreased exactness imitations was proposed to diminish the cost of actualizing particular excess in FIR channels. In [4], a relationship between the memory components of a FIR channel and the info arrangement was utilized to identify mistakes. Different plans have misused the FIR properties at a word level to likewise accomplish adaptation to internal failure [5]. The utilization of buildup number frameworks [6] and math codes [7] has additionally been proposed to secure channels. At long last, the utilization of various usage structures of the FIR channels to right mistakes with just a single repetitive module has additionally been proposed [8].

In every one of the procedures specified in this way, the security of a solitary channel is considered.

In any case, it is progressively regular to discover frameworks in which a few channels work in parallel. This is the situation in channel banks [9] and in numerous advanced correspondence frameworks [10]. For those frameworks, the security of the channels can be tended to at a larger amount by considering the parallel channels as the square to be ensured. This thought was investigated in [11], where two parallel channels with a similar reaction that prepared diverse information signs were considered. It was demonstrated that with just a single repetitive duplicate, single blunder revision can be actualized. In this manner, a noteworthy cost lessening contrasted and TMR was gotten.

In this brief, a general plan to secure parallel channels is displayed. As in [11], parallel channels with a similar reaction that procedure diverse information signs are considered. The new approach depends on the utilization of blunder rectification codes (ECCs) utilizing each of the channel yields as what might as well be called a bit in and ECC codeword. This is a speculation of the plan exhibited in [11] and empowers more effective usage when the quantity of parallel channels is substantial. The plan can likewise be utilized to give all the more capable security utilizing propelled ECCs that can redress disappointments in products modules.

This plan is actualized utilizing the idea of reversible entryways. Reversible rationale requires non-devastation of data. Along these lines the quantity of information sources must be equivalent to the quantity of yields. It is conceivable to fabricate circuits from many-port doors that don't obliterate the ability to store data and Because of every state-change is reversible. Rationale circuits worked from reversible rationale doors, you can diminish the power utilization self-assertively while running at full speed.

Whatever is left of this brief presents the new plan by first condensing the parallel channels considered in Section II. At that point, in Section III, the proposed plan is introduced. Segment IV displays a contextual analysis to show the viability of the approach. At long last, the conclusions are compressed in Section V.

II. PARALLEL FILTERS WITH THE SAME RESPONSE

A discrete time filter implements the following equation:

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

where $x[n]$ is the input signal, $y[n]$ is the output, and $h[l]$ is the impulse response of the filter [12]. 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. There
are several structures to implement both FIR and IIR filters.

In the following, a set of k parallel filters with the same response and different input signals are considered. These parallel filters are illustrated in Fig. 1. This kind of filter is found in some communication systems that use several channels in parallel. In data acquisition and processing applications is also common to filter several signals with the same response.

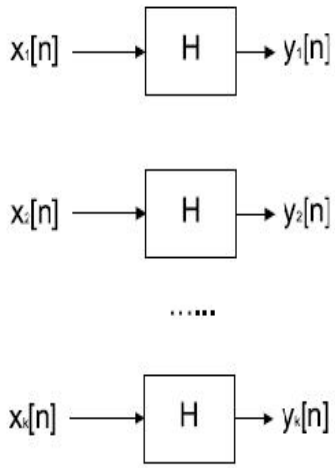

Fig. 1. Parallel filters with the same response.

An interesting 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

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

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

III. PROPOSED SCHEME

The new method depends on the utilization of the ECCs. A basic ECC takes a piece of k bits and

produces a square of n bits by including n−k equality check bits [13]. The equality check bits are XOR blends of the k information bits. By legitimately planning those mixes it is conceivable to recognize and redress blunders. For instance, let us consider a basic Hamming code [14] with k = 4 and n = 7. For this situation, the three equality check bits p1, p2, p3 are processed as an element of the information bits d1, d2, d3, d4 as takes after:

$$\begin{aligned} p_1 &= d_1 \oplus d_2 \oplus d_3 \\ p_2 &= d_1 \oplus d_2 \oplus d_4 \\ p_3 &= d_1 \oplus d_3 \oplus d_4. \end{aligned} \tag{3}$$

The information and equality check bits are put away and can be recouped later regardless of the possibility that there is a blunder in one of the bits. This is finished by recomputing the equality check bits and contrasting the outcomes and the qualities put away. In the illustration considered, a blunder on d1 will bring about mistakes on the three equality checks; a mistake on d2 just in p1 and p2; a blunder on d3 in p1 and p3; lastly a blunder on d4 in p2 and p3. Hence, the information bit in blunder can be found and the mistake can be adjusted. This is regularly figured as far as the producing G and equality check H frameworks. For the Hamming code considered in the case, those are

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \tag{4}$$

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}. \tag{5}$$

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

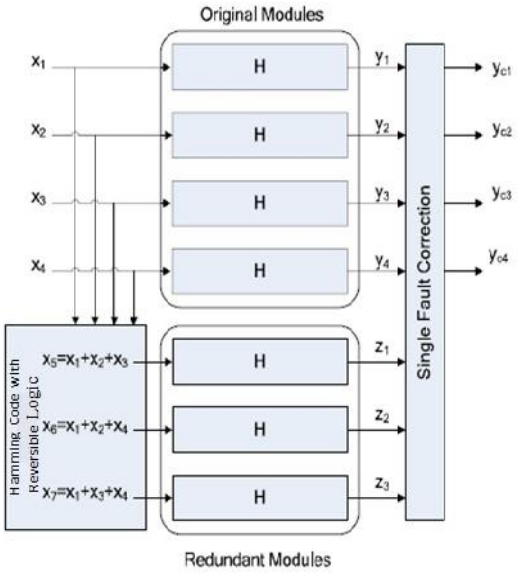

Fig. 2. Proposed scheme for four filters and a Hamming code using Reversible Logic

Encoding is done by computing $y = x \bullet G$ and error detection is done by computing $s = y \bullet HT$,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. The correspondence of values of s to error position is captured in Table I. Once the erroneous bit is identified, it is corrected by 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 $y1$, $y2$, $y3$, $y4$ and the Hamming code, the check filters would be

$$\begin{aligned} z_1[n] &= \sum_{l=0}^{\infty} \left(x_1[n-l] + x_2[n-l] + x_3[n-l]\right) \cdot h[l] \\ z_2[n] &= \sum_{l=0}^{\infty} \left(x_1[n-l] + x_2[n-l] + x_4[n-l]\right) \cdot h[l] \\ z_3[n] &= \sum_{l=0}^{\infty} \left(x_1[n-l] + x_3[n-l] + x_4[n-l]\right) \cdot h[l] \end{aligned} \tag{6}$$

and the checking is done by testing if

$$\begin{aligned} z_1[n] &= y_1[n] + y_2[n] + y_3[n] \\ z_2[n] &= y_1[n] + y_2[n] + y_4[n] \\ z_3[n] &= y_1[n] + y_3[n] + y_4[n]. \end{aligned} \tag{7}$$

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.

The overall scheme is illustrated on Fig. 2. It can be observed that correction is achieved with only three redundant filters.

For the filters, correction is achieved by reconstructing the erroneous outputs using the rest of the data and check outputs. For example, when an error on $y1$ is detected, it can be corrected by making

$$y_{c1}[n] = z_1[n] - y_2[n] - y_3[n]. \quad (8)$$

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

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & -1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & -1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & -1 \end{bmatrix} \quad (9)$$

and calculate $s = yHT$ to detect errors. Then, the vector s is also used to identify the filter in error. In our case, a nonzero value in vector s is equivalent to 1 in the traditional Hamming code. A zero value in the check corresponds to a 0 in the traditional Hamming code.

It is important to note that due to different finite precision effects in the original and check filter implementations, the comparisons in (7) can show small differences. Those differences will depend on the quantization effects in the filter implementations that have been widely studied for different filter structures. The interested reader is referred to [12] for further details. Therefore, a threshold must be used in the comparisons so that values smaller than the threshold are classified as 0. This means that small errors may not be corrected. This will not be an issue in most cases as small errors are acceptable. 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 alternative formulation, it is clear that the scheme can be used for any number of parallel filters and any linear block 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. This is the same with respect to customary ECCs for which the overhead abatements as the square size increments [13].

The extra operations required for encoding and deciphering are basic increases, subtractions, and examinations and ought to have little impact on the general unpredictability of the circuit. This is represented in Section IV in which a contextual analysis is exhibited.

In the discourse, so far the impact of blunders influencing the encoding and translating rationale has not been considered. The encoder and decoder incorporate a few increases and subtractions and thusly the likelihood of blunders influencing them can't be ignored. Concentrating on the encoders, it can be seen that a portion of the figurings of the zi share adders. For instance, taking a gander at (6), z1 and z2 share the term y1 + y2. Hence, a mistake in that snake could influence both z1 and z2 bringing about a miscorrection on y2. To guarantee that solitary mistakes in the encoding rationale won't influence the information yields, one alternative is to keep away from rationale sharing by figuring each of the zi autonomously. All things considered, mistakes will just influence one of the zi yields and as indicated by Table I, the information yields y j won't be influenced. So also, by keeping away from rationale sharing, single blunders in the calculation of the s vector will just influence one of its bits. The last remedy components, for example, that in (8) should be tripled to guarantee that they don't engender mistakes to the yields. Gadget Utilization rundown appeared in the Table IV legitimize that the proposed outline procedure using hamming code with reversible rationale decreases the power utilization and many-sided quality furthermore prompt to change in vitality productivity. As their unpredictability is little contrasted and that of the channels, the effect on the general circuit cost will be low. This is affirmed by the outcomes exhibited in Section IV for a contextual investigation.

TABLE II
RESOURCE COMPARISON FOR FOUR PARALLEL FIR FILTERS

	Unprotected	TMR	Method in [7]	Proposed
Slices	2944	9020	7740	6409
Flip-flops	1224	3984	3980	2941
LUTs	5692	17256	13640	12032

TABLE III
RESOURCE COMPARISON FOR ELEVEN PARALLEL FIR FILTERS

	Unprotected	TMR	Method in [7]	Proposed
Slices	8096	24805	21285	14422
Flip-flops	3366	10956	10945	6478
LUTs	15653	47454	37510	28331

TABLE IV

DEVICE UTILIZATION SUMMARY (ESTIMATED VALUES)			
Logic Utilization	Used	Available	Utilization
Number of Slices	1	4656	0%
Number of 4 input LUTs	2	9312	0%
Number of Bonded IOBs	11	232	4%

III. CASE STUDY

To evaluate the effectiveness of the proposed scheme of Protecting Fault Tolerant Parallel Filters by Hamming code with Reversible logic 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 zi , 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 = 7is used. The second is a block of eleven parallel filters for which a Hamming code with k = 11 and n = 15 is used. Both configurations have been implemented in HDL and mapped to a Xilinx Virtex 4 XC4VLX80 device.

The first evaluation is to compare the resources used by the proposed scheme with those used by TMR, the protection method proposed in [7] (with m = 7) and by an unprotected filter implementation. Those results are presented in Tables II III and IV. It can be observed that the proposed technique provides significant savings (from 26% to 41%) for all the resource types (slices, flip-flops, and LUTs) compared with the TMR. The benefits are larger for the second configuration as expected with values exceeding 40% for all resource types. In that case, the relative number of added check filters $(n-k)/n$ is smaller. When compared with the arithmetic code technique proposed in [7], the savings are smaller but still significant ranging from 11% to 40%. Again, larger savings are obtained for the second configuration.

In summary, the results of this case study confirm that the proposed scheme can reduce the implementation cost significantly compared with the TMR and provides also reductions when compared with other methods such as that in [7]. As discussed before, the reductions are larger when the number of filters is large.

The second evaluation is to assess the effectiveness of the scheme to correct errors. To that end, fault injection experiments have been conducted. In particular, errors have been randomly inserted in the coefficients and inputs of the filters. In all cases, single errors were detected and corrected. In total, 8000 errors for inputs and 8000 errors for filter coefficients were inserted in the different simulation runs. This confirms the effectiveness of the scheme to correct single errors.

V. CONCLUSION

This brief 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 with reversible logic to the parallel filters outputs to detect and correct errors.In addition Reversible computing will also lead to improvement in energy efficiency. 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 can also be applied to the IIR filters. 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. The proposed scheme can also be combined with the reduced precision replica approach presented in [3] to reduce the overhead required for protection. 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. 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] M. Nicolaidis, “Design for soft error mitigation,” *IEEE Trans. Device Mater. Rel.*, vol. 5, no. 3, pp. 405–418, Sep. 2005.

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

[3] 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.

[4] T. Hitana and A. K. Deb, "Bridging concurrent 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-detection processing," *IEEE Trans. Very Large Scale Integr. (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 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 the arithmetic residue codes based fault-tolerant FIR filter design," in *Proc. IEEE IOLTS*, Jun. 2012, pp. 130–133.

[8] P. Reviriego, C. J. Bleakley, and J. A. Maestro, "Strutural DMR: A technique for implementation of soft-error-tolerant FIR filters," *IEEE Trans. Circuits Syst., Exp. Briefs*, vol. 58, no. 8, pp. 512–516, Aug. 2011.

[9] P. P. Vaidyanathan. *Multirate Systems and Filter Banks*. Upper Saddle River, NJ, USA: Prentice-Hall, 1993.

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

[11] 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.

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

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

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