# Structural DMR: a Technique for Implementation of Soft Error Tolerant FIR Filters

Pedro Reviriego, Chris J. Bleakley and Juan Antonio Maestro

**Abstract— In this paper, an efficient technique for implementation of soft error tolerant Finite Impulse Response (FIR) filters is presented. The proposed technique uses two implementations of the basic filter with different structures operating in parallel. A soft error occurring in either filter causes the outputs of the filters to differ, or mismatch, for at least one sample. The filters are specifically designed so that, when a soft error occurs, they produce distinct error patterns at the filter output. An error detection circuit monitors the basic filter outputs and identifies any mismatches. An error correction circuit determines which filter is in error based on the mismatch pattern and selects the error-free filter result as the output of the overall, error protected system. This technique is referred to as Structural Dual Modular Redundancy (DMR) since it enhances traditional DMR to provide error correction, as well as error detection, by means of filter modules with different structures. The proposed technique has been implemented and evaluated. The system achieves a soft error correction rate of close to 100% for isolated, single soft errors and has a logic complexity significantly less than that of conventional TMR.**

***Index Terms*—Filter, FIR, Soft Errors, DMR.**

## I. Introduction

RELIABITY is a major concern for advanced electronic systems [1]. Designing systems that are tolerant to soft errors is becoming increasingly important due to reductions in circuit feature size and voltage level. Soft errors are transient errors in circuit nodes that can affect both sequential and combinational elements. A wide range of techniques have been used to protect circuits against soft errors, including specialised manufacturing processes, circuit level design techniques and system level redundancy [2]. One commonly used technique is modular redundancy whereby the circuit to be protected is replicated *N* times and extra logic is added to detect and correct errors. In the case that *N* equals two, the technique is known as Dual Modular Redundancy (DMR). In DMR, the outputs of two identical modules are compared and an error is detected if the outputs differ. Conventional DMR does not provide error correction. In the case that *N* equals 3, the technique is known as Triple Modular Redundancy (TMR). TMR provides error detection and correction by means of voting.

Signal processing circuits are used in many applications including communications, data storage, audio processing and video processing [3]. Many of these circuits exhibit a regular structure and have properties that can be exploited to provide effective protection against errors [4]. Finite Impulse Response (FIR) filters are one of the most commonly used signal processing circuits. Error protection for FIR filter circuits has been widely studied in the past. Protection of the registers in FIR filters using Hamming codes and parity bits were investigated in [5] and [6], respectively. In [7], a parity-based checksum was proposed for detecting errors in FIR filters. In [8], reduced precision modular redundancy was proposed for decreasing the cost of protection. In addition, several techniques have been proposed specifically for protection of adaptive FIR filters (see [9] and references therein).

The system proposed herein employs design diversity for error protection in that different module implementations are used. Previous work has used design diversity to protect against design errors [10], multiple module faults [11] and common mode failures (a single fault affecting multiple modules) [11]. In contrast, the work described herein uses design diversity to detect and correct a single soft error occurring in a single module. To the authors' knowledge, this is the first proposal that achieves fault detection and correction using two FIR filter modules rather than three.

As already mentioned, in this paper, a novel technique for protecting FIR filters from single, isolated soft errors is proposed. The approach uses two implementations of the basic filter. The implementations differ in their structure. When a soft error occurs in one of the filters, the outputs of the filters differ (mismatch) for one or more samples. The filters are designed so that they produce different error patterns at the output. An error detection circuit compares the filter outputs and flags any mismatches. An error correction circuit uses the mismatch pattern to determine which filter is in error. It selects the output of the error-free filter as the final, error protected system output.

The rest of the paper is organized as follows. Section II provides background on FIR filter implementation in Integrated Circuits. The proposed approach is presented in

Manuscript received August 24, 2010. This work was supported by the Spanish Ministry of Science and Education under Grant AYA2009-13300-C03 and by a University College Dublin Seed Funding grant.

P. Reviriego and J.A. Maestro are with Universidad Antonio de Nebrija, C/ Pirineos, 55 E-28040 Madrid, Spain (phone: +34-914521100; fax: +34-914521110; email: {previrie, jmaestro}@nebrija.es).

C. J. Bleakley is with University College Dublin, Belfield, Dublin 4, Ireland (phone: +353-17165353; fax: +353-12695396; email: chris.bleakley@ucd.ie).

Section III. In Section IV, the technique is evaluated in terms of error protection performance and circuit area, and is compared to TMR. Finally, in Section V, the conclusions from this work are summarized.

## II. Background

A Finite Impulse Response (FIR) filter implements the following equation

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

where x[n] is the input signal, y[n] is the output and h[i] is the impulse response of the filter. The non-zero values of the impulse response are all in the range 0 to N-1.

A number of different structures can be used to implement FIR filters [12]. Figure 1 shows the transpose of the direct form. This is a straight-forward implementation of Eq. (1). Figure 2 shows a mixed transpose-cascade structure in which the filter is decomposed into two sub-filters that are connected in cascade. The functionality of the transpose-cascade filter can be described as follows

$$y''[n] = \sum_{j=0}^{2} \left( \sum_{i=0}^{L-2} x[n-j-1-i] \cdot b_1[i] \right) \cdot b_2[j] \tag{2}$$

These two filter structures can be made equivalent, in terms of functionality, by appropriate selection of $b_1[i]$ and $b_2[j]$. This can be achieved by decomposing the original filter in terms of its z-domain zeros [3].

It should be noted that, in Figure 2 a register has been placed between the two sub-filters to avoid a long critical path involving two multiplications ($b_1[0]$ and $b_2[0]$) and two additions. A critical path such as this would reduce the maximum frequency that the circuit could operate at. This register introduces a one sample delay between the outputs of the structures shown in Figure 1 and Figure 2.

Let us assume that there is no logic sharing between the multipliers. This is the case when the filters are either programmable or adaptive. Both programmable and adaptive filters require that the coefficients can be independently modified [12]. This prevents logic sharing in a fully parallel design. In contrast, multiplier logic may be shared between taps to reduce area in fixed coefficient filters.

Let us also assume that TMR is used to protect the registers that store the filter coefficients, such that soft errors in these registers do not have any affect on the filter output.

Consider a soft error occurring in the transpose form (Figure 1). Since the filter is purely feed-forward the error will only affect one output sample of y[n]. Similarly, consider a soft error occurring in the cascade filter (Figure 2). Provided that the error does not occur in the final stage ($b_2[j]$), more than one output sample will be affected. For example, soft errors in the first block of the filter will cause three consecutive outputs samples to be corrupted.

Since soft errors are rare [1][2], it can be assumed that only one error can occur in the filter system at any one time. Similarly, it can be assumed that soft errors are isolated. That is, a soft error cannot occur until after correction has been completed for the previous soft error. For the system described herein, there are assumed to be at least L+3 clock cycles between soft error events.

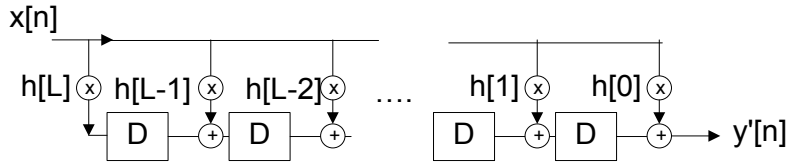


Figure 1. FIR implementation using the transpose of the direct form.

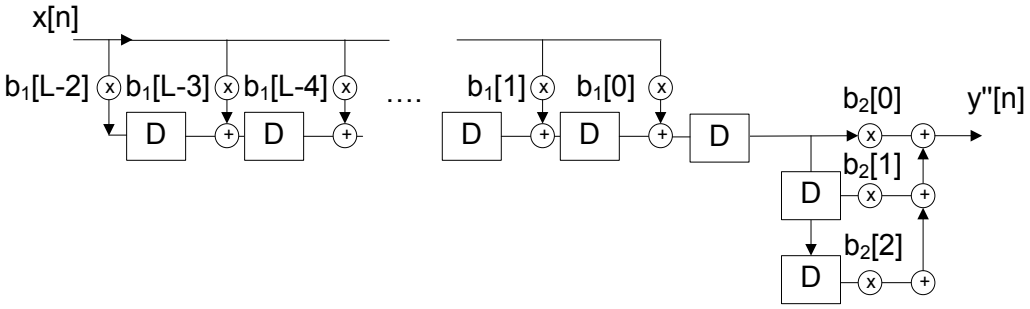


Figure 2. FIR implementation using a mixed transpose-cascade structure.

## III. Structural DMR

The proposed technique is referred to herein as Structural DMR since it uses two implementations of the basic filter with different structures. The overall architecture of the system is shown in Figure 3. The system consists of two implementations of the filter, logic for error detection and

correction, and a duplicated cascade final stage. One of the filter implementations uses the transpose of the direct form (shown in Figure 1) and the other uses the cascade form (shown in Figure 2).

The error detection and correction logic is shown in Figure 4. The error detection logic compares the transpose and cascade filter outputs. As discussed previously, a soft error in the transpose filter causes only one error in the filter output y'[n], while a soft error in all but the final stage of the cascade filter causes multiple consecutive errors in y''[n]. Hence, the error correction logic checks for consecutive mismatches between the filter outputs. If there is more than one mismatch then the error must have occurred in the cascade filter. If there is only a single mismatch then the error may have occurred in the transpose filter or in the final stage of cascade filter. Hence, the outputs of the cascade filter and the duplicated cascade final stage are compared. If they differ then the error must be in the cascade filter. Otherwise the error must be in the transpose filter. The error correction logic selects the error-free filter output as the final output from the overall error-protected system.

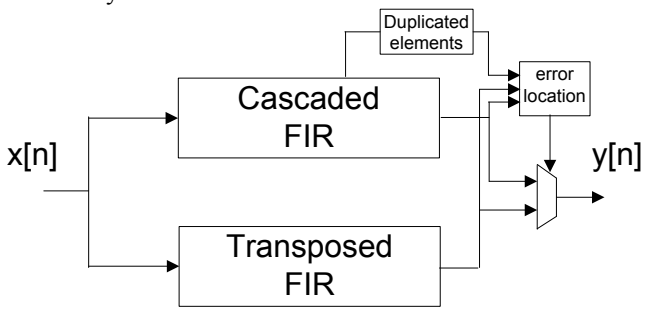


Figure 3. Proposed protection technique.

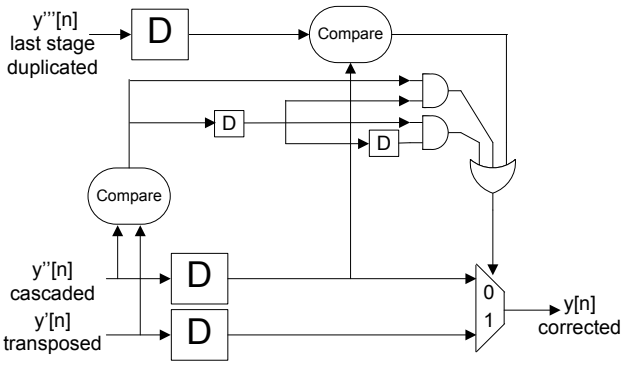


Figure 4. Error location and correction logic.

Clearly soft errors can occur in the error detection and correction logic itself. It may seem that additional protection would be needed for this logic. However a simple analysis shows that single errors in the error correction logic do not cause errors in the system output. Soft errors in the registers, comparators or selection logic only change which filter is selected as error-free. Since a single error cannot occur in the filters and in the correction logic, both of the filter outputs must be correct. Therefore, it does not matter which filter is selected for output.

In a practical implementation, a finite precision data-path must be used. Hence, in real systems, the outputs of the two modules may differ slightly even in the error-free case due to differences in quantization effects between the two filters. This problem can be addressed by employing a threshold in the comparison used for error detection. A mismatch is only flagged when the difference between the filter outputs is greater than the pre-determined threshold.

The use of a threshold introduces some limitations on the error correction effectiveness of the proposed technique. To analyze this effect, let us consider that a soft error occurs in the transpose part of the mixed transpose-cascade filter and it produces an error $e_{in}$ in the output value of this block. This error will then enter the cascade part of the filter and the following errors will occur at the output on consecutive samples: $\{b_2[0]*e_{in}, b_2[1]*e_{in}, b_2[2]*e_{in}\}$. These errors will be detected and corrected by the system unless one or more of the output errors is smaller in magnitude than the threshold $\tau$ used for comparison with transpose filter output. Therefore the following upper bound can be placed on the probability of failure, $p_f$:

$$p_f = p\left(e_{in} \min\left(|b_2[0]|,|b_2[2]|,|b_2[2]|\right) < \tau\right) \quad (3)$$

The error $e_{in}$ is uniformly distributed in the range $-2^{B-1}$ to $+2^{B-1}$ where $B$ is the number of bits used to represent $e_{in}$. In contrast, $\tau$ is set to be greater than the maximum error $e_{out}$ between the filter outputs. This error is dominated by the quantization noise in the filter structures. Conventional filter design requires that $B$ is selected so that the quantization noise at the output is low. Therefore, provided that the coefficients $b_2[0]$, $b_2[1]$ and $b_2[2]$ are selected to have similar magnitude, $p_f$ will be close to zero.

It should be noted that the use of different filter structures in the system means that quantization noise and dynamic range must be analyzed for both structures to ensure that the overall application requirements are met.

## IV. Evaluation

A case study was used to evaluate the proposed technique in terms of protection effectiveness and circuit area.

### A. Case Study

The transpose filter used for the analysis was an eleventh order low pass filter whose impulse response and frequency response are shown in Figures 5 and 6, respectively.

The cascade module was implemented as a tenth order transpose form filter, cascaded with a first order filter. The use of a first order section in the last stage minimizes the cost of duplication of the final stage. This approach can be used for odd filter orders.

The filters were implemented in Verilog with data-path quantization of 16 bits for both coefficients and data. The correction logic was implemented as described in Section III with an error detection threshold $\tau$ of 0.01.

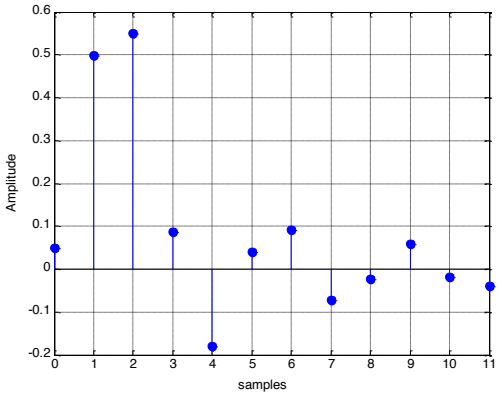


Figure 5. Case study filter impulse response.

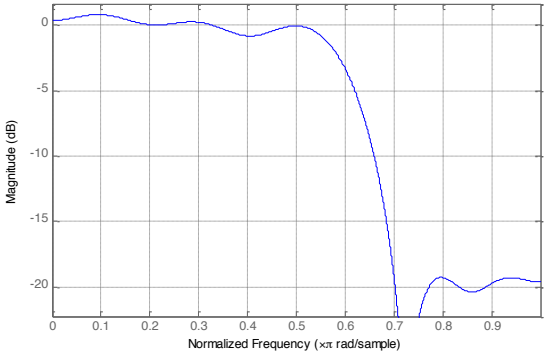


Figure 6. Case study filter frequency response.

## B. *Protection Effectiveness*

To evaluate the effectiveness of the proposed technique, the system was simulated using ModelSim and errors where inserted using the Single Event Upset Simulation Tool (SST) tool [13],[14]. A random input signal in the range -0.5 to 0.5 was used. Errors were inserted at a sufficiently low rate so to ensure that the circuit was in an error free state before each error was inserted. Each error affected either a single register bit or a single combinational node.

Fault campaigns targeting individual components of the circuit were run. Errors were inserted in the transpose filter registers and combinational logic; in the cascade filter registers and combinational logic; and in the error detection and correction logic. Each campaign simulated 1,000 soft errors. The maximum error in the protected system output was below 0.09 in all campaigns. A campaign was run to insert errors randomly in all circuit elements. Again, the campaign simulated 1,000 soft errors. The errors occurring at the individual filter outputs are shown in Figures 7 and 8. The overall system output error, after correction, is shown in Figure 9. In this case, the maximum error in the protected system output was 0.066. These campaign results validate the effectiveness of the proposed technique in protecting against isolated, single soft errors.

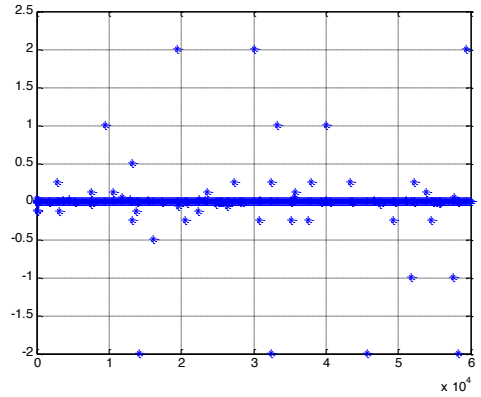


Figure 7. Error at the transposed filter output.

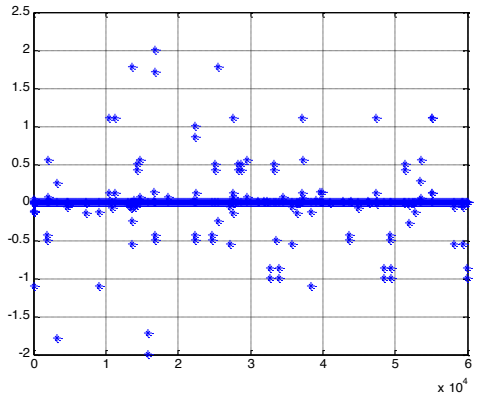


Figure 8. Error at the cascaded filter output.

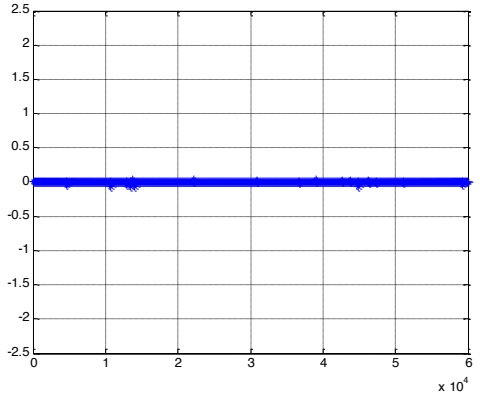


Figure 9 Error at the system output.

Figure 10 shows a zoom of the results presented in Figure 9. A small number of errors exceed the threshold. As discussed previously, these errors arise when a soft error occurs in the transpose stage of the cascade filter but only one of the erroneous output samples exceeds the threshold. This results in the error correction logic misdiagnosing the module in error and mistakenly outputting the erroneous cascade filter values. The maximum error arising in this way can be calculated from the $b_2[j]$ coefficients and the threshold $\tau$, as discussed in the previous section. In the case study, the coefficients are 1 and 8.859 so that the upper bound in (3) takes a value of 0.0859. This means that an error slightly smaller than the threshold value of 0.01 would not be detected and would lead to an error of approximately 0.0859 in the next output sample. The

magnitude of these errors can be minimized by using $b_2[j]$ coefficients of similar magnitude.

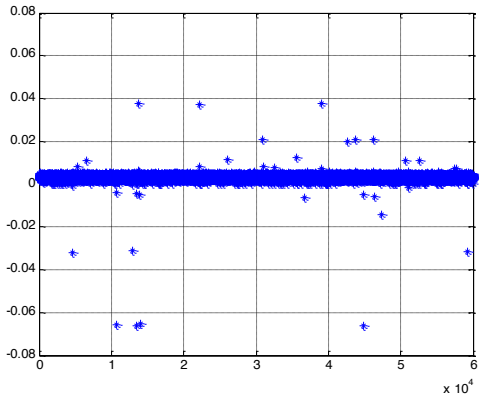


Figure 10. Detail of the error at the system output.

### C. *Complexity*

To evaluate the complexity of the proposed approach, the Verilog implementation of the system was synthesized using a 130nm library. The system was compared with Verilog implementations of the unprotected transpose filter and a TMR protected transpose filter. The results are summarized in Table I in terms of the number of equivalent gates. It can be observed that the proposed technique requires slightly more than two times the number of gates needed for the unprotected implementation. The proposed system provides a significant reduction on the cost of TMR while achieving an effective level of error protection.

TABLE I
AREA ESTIMATES FOR FILTER IMPLEMENTATIONS

| | Gate count | Ratio vs unprotected |
|---|---|---|
| Unprotected | 31,757 | 1 |
| Structural DMR | 71,365 | 2.247 |
| TMR | 95,352 | 3.003 |

## V. Conclusions and Future Work

An efficient technique to protect FIR filters from the effects of isolated, single soft errors has been presented. The technique uses two implementations of the basic filter, with different structures, operating in parallel. Use of design diversity allows error correction since the output error patterns are unique for each structure. By observing the filter output mismatch patterns, the module in error can be identified and the error corrected by selecting the output from the other module. This technique is referred to as Structural DMR as it enhances the traditional DMR approach by using filter modules with different structures.

A simple theoretical analysis has been presented to show the effectiveness of the proposed technique. A case study is used to show its effectiveness in protecting against isolated, single soft errors and also to compare its cost in terms of circuit area with TMR. The results suggest that the proposed technique is able to effectively correct isolated, single soft errors. A more complete theoretical analysis and assessment of the reliability achieved by the technique is an interesting area for future work.

It is worth mentioning that the Structural DMR technique could be applied to other FIR filter structures, such as purely cascaded structures, or combined with the reduced precision redundancy approach [8]. A reduced precision filter could replace one of the filter modules leading to lower overall circuit area. The evaluation of a combined protection technique such as this, and of other FIR filter structures, is left for future work.

Finally, since the proposed technique incorporates design diversity, it could also be used to mitigate the effects of common mode faults that affect both filter implementations at the same time [11]. This would be an additional advantage of the proposed technique, as compared to TMR, and opens a new line of investigation.

## References

[1] R. Baumman “Soft errors in advanced computer systems" IEEE Design and Test of Computers, vol. 22, issue 3, May/June 2005, pp. 258 – 266.

[2] M. Nicolaidis “Design for soft error mitigation” IEEE Transactions on Device and Materials Reliability vol. 5, issue 3, Sept. 2005, pp. 405 – 418.

[3] J.K. Proakis and D.G. Manolakis, “Digital Signal Processing: Principles, Algorithms, and Applications”, Prentice Hall, 3rd Edition, 1996.

[4] A. Reddy and P Banarjee “Algorithm-based fault detection for signal processing applications”, IEEE Transactions on Computers, vol. 39, issue 10, Oct. 1990, pp 1304 – 1308.

[5] R. Hentschke, F. Marques, F. Lima, L. Carro, A. Susin and R. Reis, “Analyzing area and performance penalty of protecting different digital modules with Hamming code and triple modular redundancy”, 15th Symposium on Integrated Circuits and Systems Design, 2002, pp. 95-100.

[6] P. Reyes, P. Reviriego, J.A. Maestro, O. Ruano, "A New Protection Technique for Finite Impulse Response (FIR) Filters in the Presence of Soft Errors", Proc. of IEEE International Symposium on Industrial Electronics, 2007.

[7] B. Zagar, and R. Redinbo, “Watchdog parity channels for digital filter protection” Eighteenth International Symposium on Fault-Tolerant Computing, 1988.

[8] B. Shim, S.R. Sridhara and N.R. Shanbhag, “Reliable low-power digital signal processing via reduced precision redundancy” IEEE Transactions on Very Large Scale Integration (VLSI) Systems vol. 12, issue 5, May 2004, pp. 497 – 510.

[9] C. Radhakrishnan and W.K Jenkins, “Fault Tolerance in Transform-Domain Adaptive Filters Operating With Real-Valued Signals” IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 57 , issue 1, 2010 , pp: 166 – 178.

[10] A. Avizienis, and J.P.J Kelly, “Fault Tolerance by Design Diversity: Concepts and Experiments” IEEE Computer vol.: 17 , issue: 8, August 1984 pp. 67-80.

[11] S. Mitra, N.R. Saxena and E.J. McCluskey, “A design diversity metric and analysis of redundant systems”, IEEE Transactions on Computers vol 51 , issue 5 , May 2002, pp 498-510.

[12] A.V. Oppenheim and R.W. Schafer, “Discrete Time Signal Processing”, Prentice Hall 1999.

[13] D. Gonzalez, “Single Event Upset Simulation Tool Functional Description”, ESA Report TEC-EDM/DCC-SST2, July 2004.

[14] O. Ruano, J.A. Maestro, P. Reyes and P. Reviriego, “A Simulation Platform for the Study of Soft Errors on Signal Processing Circuits through Software Fault Injection”, Proc. of IEEE International Symposium on Industrial Electronics, 2007.