==========
Engineering & Technology in India www.engineeringandtechnologyinindia.com
Vol. 1:3 April 2016
==========

MULTIPLE ERROR DETECTION AND CORRECTION OF PARALLEL FIR FILTER BASED ON REDUCED PRECISION LOGIC

Jeswin Baby.T., M.E.
Chitra.E.N., M.E. Student

==========

Abstract

In this paper, we propose efficient methods for error correction which provides better performance for parallel finite impulse response filters. Reduced-precision redundancy (RPR) has been shown to be a viable alternative to triple modular redundancy (TMR) for digital circuits. This paper builds on previous research by offering a detailed analysis of the implementation of RPR on FPGAs to improve reliability in soft error environments. Example implementations and fault injection experiments demonstrate the cost and benefits of RPR, showing how RPR can be used to improve the failure rate by up to 200 times over an unmitigated system at costs less than half that of TMR. A novel method is also presented for improving the error-masking ability of RPR by up to 5 times at no additional hardware cost under certain conditions. This research shows RPR to be a very flexible soft error mitigation technique and offers insight into its application on FPGAs.

Key words: Reduced Precision redundancy, Triple Modular redundant, FPGA (Field Programmable Gate Array.)

I. INTRODUCTION

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. To add redundancy, a general technique known as 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 triples the area and power of the circuit, something that may not be acceptable in some applications. When the circuit to be

protected has algorithmic or structural properties, a better option can be to exploit those properties to implement fault tolerance.

Digital filters are one of the most commonly used signal processing circuits and several techniques have been proposed to protect them from errors. Most of them have focused on finite-impulse response(FIR) filters. To operate reliably in space, a hardware mitigation strategy, such as triple modular redundancy (TMR), must be applied. TMR, however, is very expensive and requires three times more hardware resources than an unmitigated circuit. Motivated by the observation that an FPGA-based radio comprises mostly arithmetic operations, this paper explores the application of reduced precision replica (RPR) to the problem. The metric used to evaluate the effectiveness of RPR is the bit error rate (BER) achieved by the FPGA-based radio. To fully evaluate the benefits of RPR on a communications system, the impact of ionizing radiation on BER must be well understood

RPR is a relatively new technique and is more difficult to implement than TMR. There are a number of important design decisions that must be made for each circuit protected by RPR. These choices include selecting the precision of the reduced-precision circuits and determining the threshold for detecting low-magnitude errors. This paper expands on previous work by clarifying the design space of these design choices and defining the trade-offs associated with these parameters. By understanding the impact of these design choices, more efficient SEU mitigation can be achieved. Using this insight, this paper introduces a new method to increase the effectiveness of RPR by up to 5 times for some systems with no additional hardware cost.

A. Single Event Upsets

SRAM-based (static random access memory) FPGAs consist of a large array of memory cells. These memory cells hold both user data and configuration data that define the operation of the circuit. Charged particles affect these cells by occasionally inverting the contents of a particular cell. To protect an FPGA design from SEUs, several fault tolerance techniques are typically used. First, the upsets themselves are periodically repaired to prevent upset accumulation. Hardware redundancy techniques involve the use of additional, redundant hardware to mask the effects of SEUs. TMR uses three copies of the circuit and

voting to choose the correct output. Figure 1 shows a simplified block diagram of a digital filter protected with TMR. As long as two of the three modules are operating correctly, the final output is correct. TMR is popular because it is straightforward to implement and provides very effective protection for any type of design.

II. TRIPLE MODULAR REDUNDANCY

Triple Modular Redundancy has three identical logic circuits (logic gates) are used to compute the specified Boolean function. The set of data at the input of the first circuit are identical to the input of the second and third gates. In computing, triple modular redundancy, sometimes called triple-mode redundancy.TMR is a fault-tolerant form of

N-modular redundancy, in which three systems perform a process and that result is processed by a majority-voting system to produce a single output. If any one of the three systems fails, the other two systems can correct and mask the fault.

The TMR concept can be applied to many forms of redundancy, such as software redundancy in the form of N-version programming, and is commonly found in fault-tolerant computer systems. Some ECC memory uses triple modular redundancy hardware (rather than the more common Hamming code), because triple modular redundancy hardware is faster than Hamming error correction software.

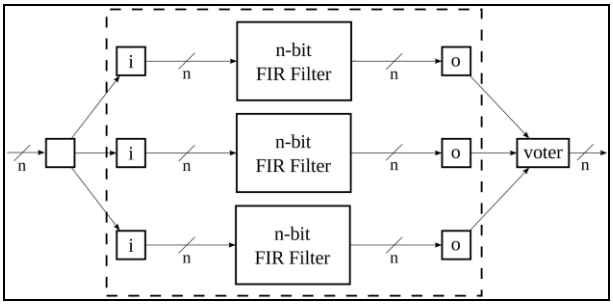

Fig. 1. Simplified block diagram of n-bit FIR filter protected with TMR

A. TMR with Hamming Code

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 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 ⊕d2 ⊕d3

p2 = d1 ⊕d2 ⊕d4

p3 = d1 ⊕d3 ⊕d4

For the case of four filters y1, y2, y3, y4 and the Hamming code, the check filters would be

$$z1[n] = \sum_{l=0}^{\infty} (x1[n-l] + x2[n-l] + x3[n-l]) \cdot h[l]$$
$$z2[n] = \sum_{l=0}^{\infty} (x1[n-l] + x2[n-l] + x4[n-l]) \cdot h[l]$$
$$z3[n] = \sum_{l=0}^{\infty} (x1[n-l] + x3[n-l] + x4[n-l]) \cdot h[l]$$

B. Error Detection and Correction

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

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

$z3\,[n] = y1\,[n] + y3[n] + y4[n].$

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.. For example, when an error on y1 is detected, it can be corrected by making

$Y\,c1[n] = z1[n] - y2[n] - y3[n].$

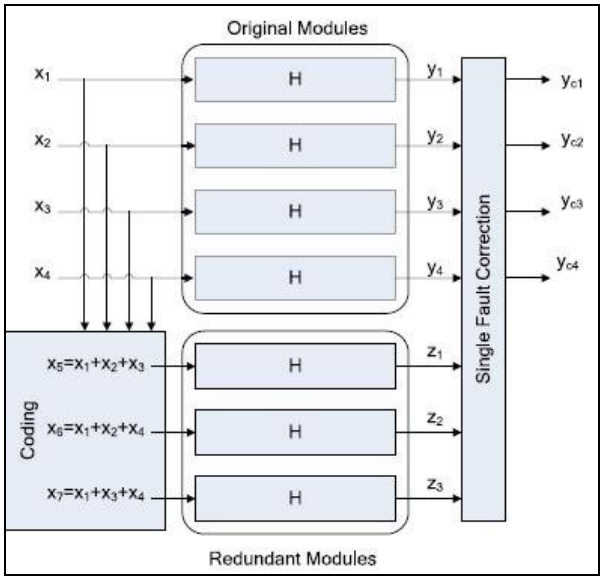

Fig 2: Four Filters and Hamming Codes.

Similar equations can be used to correct errors on the rest of the data outputs and calculate s = yHT to detect errors. For the filters, correction is achieved by reconstructing the erroneous outputs using the rest of the data and check outputs 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

Table1: Error Location in the Hamming Code

S1 S2 S3	ERROR BIT POSITION	ACTION
000	NO ERROR	NONE
111	d 1	CORRECT d1
110	d 2	CORRECT d2

101	d 3	CORRECT d3
011	d 4	CORRECT d4
100	P 1	CORRECT p1
010	P 2	CORRECT p2
001	P 3	CORRECT p3

III. THE PROPOSED WORK

A. REDUCED PRECISION REDUNDANCY

RPR is a redundancy technique similar to TMR that requires less hardware overhead by using reduced-precision (RP) arithmetic in two of its three replicas. It takes advantage of the fact that RF arithmetic can be a good estimate of computations that use higher precision. When TMR protects the entire circuit and provides an error-free output, RPR simply limits the error at the output of a module.

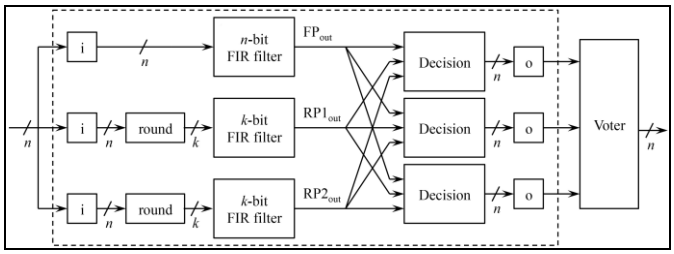

Fig. 3.Block diagram of an n-bit FIR filter protected with RPR using k-bit RF filters (k<n)

. In this paper, a set of four parallel filters with 16 coefficients are used. The input data x[n] and coefficients bi are quantized to 8 bits. The filter output is quantized with 16 bits. Error is introduced at the output of fourth filter y4 [n].Using RPR technique, single bit error

is detected and corrected. Verilog Code is written and simulated in ModelSim and Xilinx ISim and implemented in Virtex 5 xc5vlx110t-2ff1136.

RPR is able to reduce the negative impact of SEUs on numerical computations; none of this work has investigated the benefits of RPR on the performance of a communication system. This paper measures the effects of soft errors on the BER of a communications receiver. RPR is applied to two different styles of communication receivers, and the BER of the receivers protected by RPR are compared against a receiver without RPR. In addition to RPR critical clock and reset signals are included in the mitigation approach to maximize the benefits of RPR.

B. Implementation of RPR with TMR

It shows a block diagram of an n-bit finite impulse response (FIR) filter (a filter with n-bit registers and coefficients) protected with RPR. Note that the decision blocks and outputs can be triplicates as well to avoid single points of failure in those modules. The outputs of the three identical decision blocks are voted on, as in the TMR system.

To determine the presence of an error, the decision block compares the outputs of the full-precision (FP) filter (FPout) with the outputs of the two RP filters (RP1out and RP2out) as follows:

if ((FP out -RP1out>Th) AND (RP1out =RP2out)

Output= RP2out

else

Output = FP out.

In other words the FP output is used when no error is found or when the two RP modules disagree. Otherwise, the RP output is used, which provides an estimate of the correct FP output. RPR, in the form presented here, has two main parameters that can be adjusted. The impact of these parameter settings can be understood in terms of the arithmetic error.

€= FP true RP true

Where FP true and RP true are the outputs of the FP and RP filters, respectively, when no SEU is present. First, the size of the RP modules can be modified. The size of the RP filters is measured by the bit-width of the filter input signal, k. A larger RP filter gives a better estimate of the FP filter. This results in a better detection of errors in the FP filter and a lower €. A smaller RP filter is desirable because a smaller RP filter reduces the cost of mitigation. The second parameter for RPR is the threshold value Th. A threshold that is too small will cause the RP output to be chosen even when there are no errors in the FP module. To prevent this, Th ¸ max € is required. On the other hand, if Th is too large, the FP output is used even when there are significant errors in that module. In fact any error that is larger than max € must be due to an upset in the system. Consequently, Th should be no greater than this value.

IV RESULTS AND DISCUSSION

The idea of this paper is to compare the resources used by the RPR, ECC and TMR method. From synthesis report of all the three methods it is observed that area utilization is less in RPR method compared to other methods saving 30% of all resource types (slices, flip-flops and LUTs). Delay in proposed scheme is less compared to TMR and more compared to ECC and power consumption is less in RPR compared to other methods. Here the modules are taken as FIR filter simulation results shows the case when two modules are faulty. If module2 is faulty then the module2 output is recovered. The fault free output is copied to the faulty module. However, if module2 is permanent fault, then module2 is discarded if module3 is also faulty, and then the system will be halted. RPR has an advantage over TMR when it is able to sufficiently limit the magnitude of the SEU induced noise at a lower hardware cost. RPR is not suited to protect any type of circuitry as TMR is the decision hardware required.. Operations that can be approximated with less hardware than the standard module are candidates for RPR. RPR has been used to protect arithmetic operations. In addition the approximation and the decision hardware required to choose the final output must not exceed the cost of TMR, otherwise, any advantage of RPR is lost.

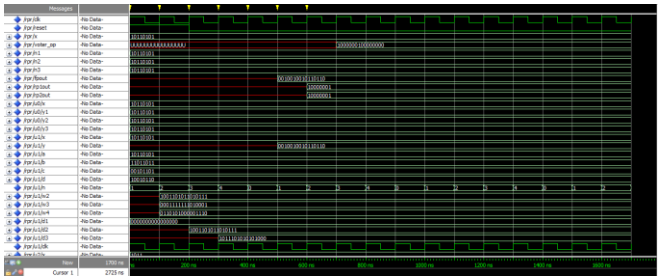

Fig 4: Simulation result of multiple error recovery using RPR

Fig.4 shows the simulation result of multiple error recovery in TMR using RPR technique used in FIR filter All the modes of operation is same as Scan chain multiple error recovery in TMR. The result shows that the proposed technique has greatly reduced in terms of area and power. The main advantage of this technique is that it can detect multiple faults with great minimization in terms of area and delay. The results from these experiments provide concrete evidence that RPR is an effective way of protecting communication circuits from the effects of SEUs.

Table II: Implementation Result

PARAMETER	EXIXTING	PROPOSED
AREA	59.656	37.680
POWER	1364mw	1108mw
DELAY	14.873ns	11.485ns

V CONCLUSION

In this paper, comparison of RPR, TMR and ECC protected Parallel Filters in terms of implementation cost and effectiveness to correct the errors is done. Parallel filters that have same impulse response and with different input and outputs sequences are considered. RPR method is more suitable for multi-bit errors which will also use less area and power compared to other techniques. TMR method can also detect multi-bit errors in any filters but consumes more area and power whereas ECC scheme can detect and correct only single bit error. In Future, utilization of IIR filters instead of FIR filters will be carried out. The extension of the scheme to parallel filters that have the same input and different impulse responses will also be done.

==========

REFERENCES

[1] Cummings, M. and Haruyama, S.FPGA in the software radio.*IEEE Communications Magazine*, **37**, 2 (Feb. 1999),108—112.

[2] Caffrey, M. A space-based reconfigurable radio. In T. P. Plaks and P. M. Athanas (Eds.), *Proceedings of the International Conference on Engineering of Reconfigurable Systems and Algorithms* (ERSA '02), Las Vegas, NV, June 2002, pp. 49—53.

[3] Banu, R. and Vladimirova, T. Fault-tolerant encryption for space applications. *IEEE Transactions on Aerospace and Electronic Systems*, **45**, 1 (Jan. 2009), 266—279

[4] Adell, P., et al Digital control for radiation-hardened switching converters in space. *IEEE Transactions on Aerospace and Electronic Systems*, **46**, 2 (Apr. 2010), 761—770.

[5] Dodd, P. and Massengill, L. Basic mechanisms and modeling of single-event upset in digital microelectronics. *IEEE Transactions on Nuclear Science*, **50**, 3 (June 2003), 583—602

[6] Johnson, E., Wirthlin, M. J., and Caffrey, M. Single-event upset simulation on an FPGA. In T. P. Plaks and P. M. Athanas (Eds.), *Proceedings of the International Conference on Engineering of Reconfigurable Systems and Algorithms* (ERSA), Las Vegas, NV, June 2002, pp. 68—73.

[7] Ostler, P., et al. SRAM FPGA reliability analysis for harsh radiation environments. *IEEE Transactions on Nuclear Science*, **56**, 6 (2009), 3519—3526.

[8] Carmichael, C. Triple module redundancy design techniques for Virtex FPGAs. Xilinx Corporation, Technical Report, Nov. 1, 2001, xAPP197 (v1.0)

[9] Morgan, K., et al. A comparison of TMR with alternative fault-tolerant design techniques for FPGAs. *IEEE Transactions on Nuclear Science*, **54**, 6 (2007),2065—2072.

[10] Huang, K-H. and Abraham, J. Algorithm-based fault tolerance for matrix operations. *IEEE Transactions on Computers*, **C-33**, 6 (June 1984), 518—528.

==========

Jeswin Baby. T., M.E.
jeswinbaby@gmail.com

Chitra. E.N., M.E. Student
Kukku.en@gmail.com

Department of Electronics & Communication Engineering
Sri Subramanya College of Engineering & Technology
NH - 209, Sukkamanaickenpatti
Palani 624615
Tamil Nadu
India