Enhanced Implementations of Hamming Codes to Protect FIR Filters

Shih-Fu Liu, Pedro Reviriego, *Member, IEEE*, and Juan Antonio Maestro, *Member, IEEE*

***Abstract*—Soft errors, especially those produced in harsh environments (as for example radiation), are a major concern for digital circuits. As technology shrinks, the failure probability of systems working in these environments is steadily growing. In order to tackle this issue, several design techniques are being used, which consider reliability as a key design parameter. Those based on redundancy, as Triple Modular Redundancy (TMR) and Hamming Codes, are especially popular, since they are quite straightforward to implement. However, this usually comes at a high cost, which many times is unfeasible for certain kinds of applications. Therefore, it is convenient to understand the structure and functionality of circuits in order take advantage of their inherent protection capabilities and therefore, minimize the extra redundancy. In this paper, this approach is used in order to optimize the protection of finite impulse response filters using Hamming codes.**

***Index Terms*—Digital filters, hamming codes, reliability, Single Event Upset (SEU).**

I. Introduction

SINCE the early stages of microelectronics, radiation has been identified as a source of alteration of functionality of integrated circuits. As the technology is steadily evolving and shrinking from generation to generation, the effects of a charged particle striking the silicon surface of an integrated circuit are getting more and more significant [1]. Radiation effects can be divided into two categories: Single Event Effects (SEE) and Total Ionizing Dose (TID). Permanent damage of the transistor or device structure results from TID and from the Single Event Latchup (SEL), Single Event Gate Rupture (SEGR) and Single Event Burnout (SEB) subclasses of SEE. Single Event Transient (SET) and Single Event Upset (SEU), which cause temporary or transient errors, are the other subclasses of SEE. Techniques for mitigation of these effects are addressed in [2], [3].

When a particle hits the silicon, it loses its energy and transmits it to the silicon, causing a current burst. In the case of SEUs, these can randomly change the content of storage cells. To protect storage cells of integrated circuits from this phenomenon, several approaches may be followed. One is by technology hardening of memory cells [4] and another one is by designing circuits able to detect an SEU event and act accordingly to prevent error propagation and guarantee full reliability in the system. Triple Modular Redundancy (TMR) and Error Detection and Correction Code (EDAC), like Hamming Code, are examples of such methods.

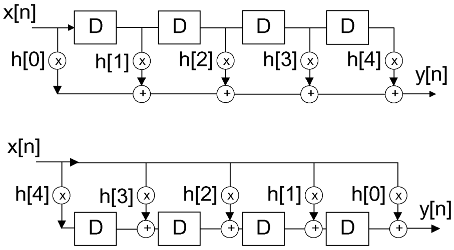

Fig. 1. Direct form FIR filter structure and transposed FIR filter structure.

Filters are commonly used in digital communication systems for equalization, signal separation, noise reduction, etc. As communications are fundamental to space borne applications, such as satellites, unmanned missions, etc., digital filters play an important role in space systems [5].

There are two main types of digital filters: the recursive and the non-recursive filters. They are referred as infinite impulse response (IIR) filters and finite impulse response (FIR) filters, respectively. The FIR filters are widely used, as they have good stability and can be easily designed to match a given response.

This paper introduces optimizations for the use of Hamming Codes to protect generic FIR filter, described by (1) and illustrated in Fig. 1, to show that by knowing attributes of the design to protect, it is possible to reduce resource consumption and achieve an optimal design

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

Fig. 1 shows the direct form and the transposed form of a 4-tap FIR filter structure, where the input vector $x[n]$ is stored into the first tap D of the delay line, and then shifted through all the taps. In this process, the content of the taps are multiplied with their corresponding coefficient $h[i]$ and the sum of these products yields the filter output $y[n]$.

As mentioned, TMR and Hamming EDAC are successful methods to protect a design from SEUs. These techniques were studied by Hentschke *et al.* in [6] for FIR filters.

TMR is the simplest and effective way for protecting a design. The area consumption of this method is obviously up to three times higher, depending on which implementation is chosen. The optimal design of the TMR logic is discussed in [7], where Lima *et al.* researched TMR logic implementations of a digital FIR filter for FPGAs.

Manuscript received September 07, 2009; revised February 11, 2010; accepted February 11, 2010. Date of current version August 18, 2010. This work was supported in part by the Spanish Ministry of Science and Innovation under Grant AYA2009-13300-C03-01, by the Regional Government of Madrid, and by the European Union FEDER programme.

The authors are with Universidad Antonio de Nebrija, C/Pirineos, 55 E-28040 Madrid, Spain (e-mail: {sliu,previrie,jmaestro}@nebrija.es).

Digital Object Identifier 10.1109/TNS.2010.2051162

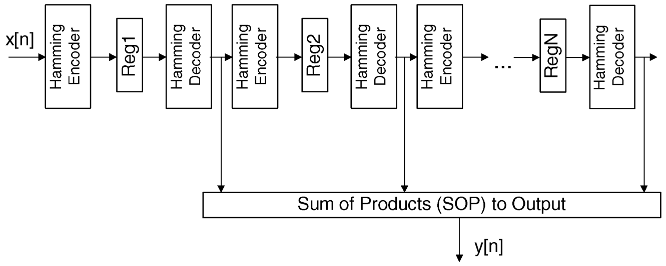

Fig. 2. Existing Hamming EDAC protection approach.

Hamming codes are a simple class of block codes using the Hamming rule to determine the parity bits based on the number of information bits. This rule is articulated by the inequality given in

$$d + p + 1 \leq 2^p \tag{2}$$

where d is the number of data bits and p is the number of parity bits. These codes have a minimum distance of three, which describes the number of different bits between two valid codewords, and thus they are capable of correcting all single errors within a block. This capability is defined as Single Error Correction (SEC). Syndrome decoding is especially suited for Hamming codes. In fact, the syndrome can be formed to act as a binary pointer to identify the error location.

Hamming encoders and decoders perform specific combinational operations on the data in order to generate parity bits (in the case of encoders) and to correct errors (in the case of decoders).

A previous approach to protect FIR filters using Hamming codes consists in adding one encoder and one decoder before and after each register that is going to be protected [6], as illustrated in Fig. 2.

This way of using EDAC codes to protect the circuit incurs a bigger delay in the critical path with respect to the use of TMR, which will be discussed further in this paper.

The area is obviously larger than in the case of the unprotected FIR, as p additional registers are needed in each tap plus the combinational logic for the encoders and decoders, but for some implementations it is lower than in the case of TMR, as shown in the following sections.

In the rest of the paper, alternative approaches to protect FIR filters with Hamming codes are presented.

II. Proposed Techiques

In this section, several enhancements are proposed to reduce the number of encoders. These are based on the specific system knowledge of the FIR implementation, an approach that has been previously used to protect other circuits [8], [9].

A. Hamming Single Encoder

One of these enhancements would be to remove the encoders from the delay line, as shown in Fig. 3, as they are only used if there are errors in the circuit and even in that situation, if only a single error is present in the register, it can still be corrected with the decoder at each stage. In summary, these additional encoders are useful only if we assume that a tap value will be hit by more than one SEU at different time instants, as it propagates through the delay line.

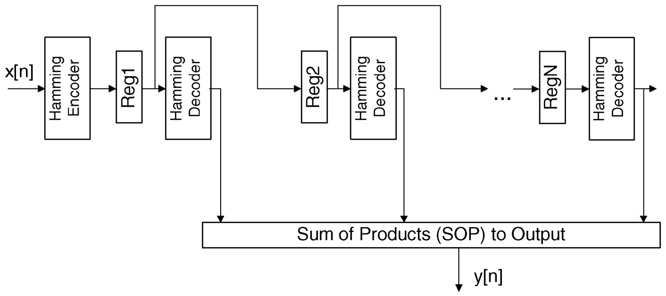

Fig. 3. FIR filter protection with a single encoder.

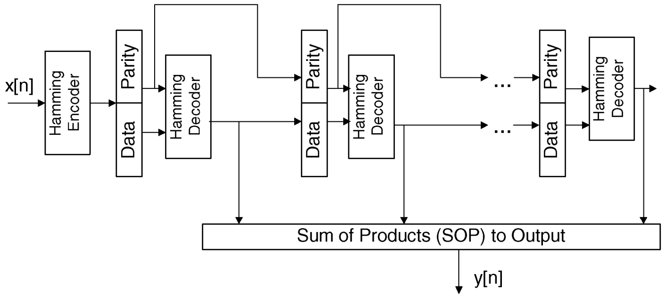

Fig. 4. FIR filter protection with additional data protection.

B. Additional Hamming Data Protection

A further improvement would be to use the output of each decoder to feed the data bits of the next register while the parity bits are taken directly from the previous stage, as shown in Fig. 4. This would allow recovering from multiple errors that occur in the data bits as long as they happen in different clock cycles. This is achieved without additional encoders.

C. Shared Hamming Decoder

A more sophisticated approach to reduce the complexity is shown in Fig. 5, where the Hamming decoder is broken apart yielding a syndrome calculator, an error locator and an error corrector. The syndrome is calculated through XOR operations of the data and parity bits as shown in Fig. 5(a). This should contain only zeros when there are no SEUs. When the data bits contain an SEU, then there will be 1's in the syndrome for identifying the SEU position in the locator, using the syndrome information.

The locator sends out an error vector with the exact position of the bit-flip to the corrector. This uses the OR-combined syndrome as an enable to initiate with the received error vector the correction of the faulty bit [Fig. 5(b)].

In this way, the locator logic is shared among all taps reducing overall complexity under the assumption that only one SEU per cycle will occur. Moreover, it allows recovering from multiple errors in the data bits in different clock cycles, as in the design of Fig. 4. These improvements to the previous technique proposed in [6] are further researched in this paper to prove that they reduce the area cost.

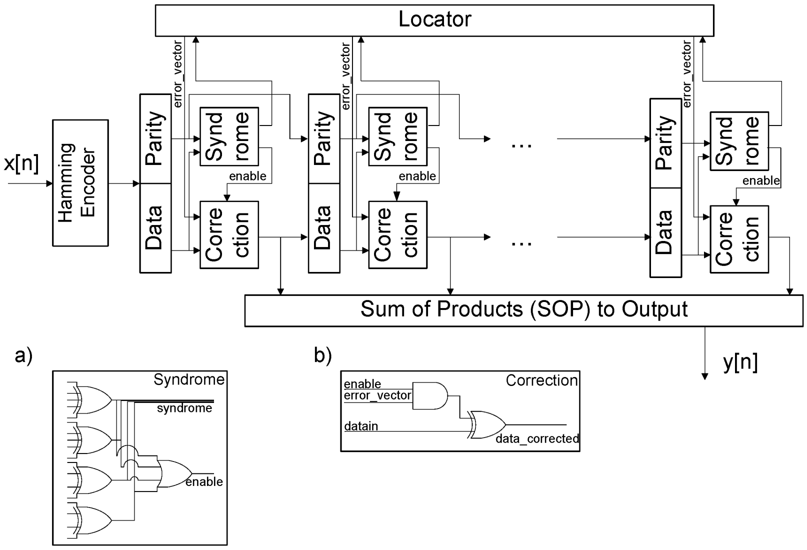

Fig. 5. FIR with Hamming using one encoder and a common decoder; (a) syndrome circuit, (b) correction circuit.

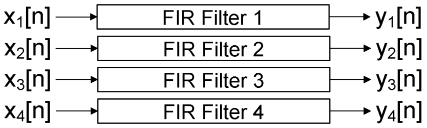

Fig. 6. Four parallel filters.

D. *Exploiting Hamming Through Parallelism*

Nowadays, it is common to find systems where parallel FIR filter structures are used, as for instance in digital communication such as wireless LAN [10] or Ethernet [11], where there are 2 or 4 parallel FIR filters, respectively (see Fig. 6).

By unifying the data streams from all the channels and coding them as one, as depicted in Fig. 7, reduction is achieved in the parity bits since when doubling the data bits d, the parity bits p only increase by one. According to (2), when increasing the data bits from 8 to 16 or 32, the parity bits increase by 1 or 2, respectively. In such a case, it is possible to use the error vector from Fig. 5 instead of doubling or quadrupling it for 2 or 4 parallel FIR filters, respectively. This is done by using different enables for each data block.

All the techniques proposed in this section optimize and highlight the benefits of Hamming protection. However, Hamming codes have some drawbacks when they are compared with TMR. The main one is that for all proposed techniques, the decoder unit or the syndrome, locator and corrector are added to the critical path of the output, and therefore, it decreases the maximum operating frequency of the circuit.

III. EXPERIMENTAL RESULTS

In the previous sections, some Hamming EDAC protection techniques have been discussed that can be used for generic FIR filters. In this section, their effectiveness for actual implementations will be evaluated. To that end, the proposed techniques have been implemented in VHDL and then the circuits have been synthesized for a commercial ASIC library [12] and for the Xilinx VIRTEX-5 [13]. This will also allow assessing the efficiency of the proposed techniques in terms of circuit complexity and compare it with the traditional approaches (like TMR). Then, an experimental setup based on a single event upset simulation tool was used to insert SEUs in the circuit and evaluate their effects showing the effectiveness of the proposed techniques.

This environment is based on the Single Event Upset Simulation Tool (SST) developed at the European Space Agency [14], which has been extended in functionality for the purpose of these experiments. A block diagram of the simulation platform is depicted in Fig. 8, and a more detailed description can be found in [15].

The generic FIR filters have been implemented with 5 and 11 taps using the same structure as in [6] and [7], respectively. Equations (3) and (4) show the FIR filter coefficients, as mentioned in the cited references

$$h[n] = [-1\ 24\ 50\ 50\ 24\ -1] \tag{3}$$

$$h[n] = [1\ -1\ -9\ 6\ 73\ 120\ 120\ 73\ 6\ -9\ -1\ 1]. \tag{4}$$

For SRAM-based FPGA, implementation errors could also occur in the configuration bits, what would cause changes in the circuit functionality [7]. The rest of the paper focuses on errors on the design flip-flops only and assumes that the configuration memory is protected by other techniques such as scrubbing [7].

A. *Effectiveness*

The results show that the effectiveness of all the compared techniques is similar, since they protect against isolated single events. However, when dealing with SEUs that occur a few clock cycles apart, the different techniques offer somewhat different behaviors. In the traditional Hamming protection, as long as there is only one single event per cycle, there will be no errors in the output. However, for the protection technique depicted in Fig. 3, only one SEU could occur in each coded word, as it traverses the delay line. For example, if one bit-flip happens in the coded word of the second tap, and two cycles later (when the erroneous word reaches the fourth tap) another SEU occurs in

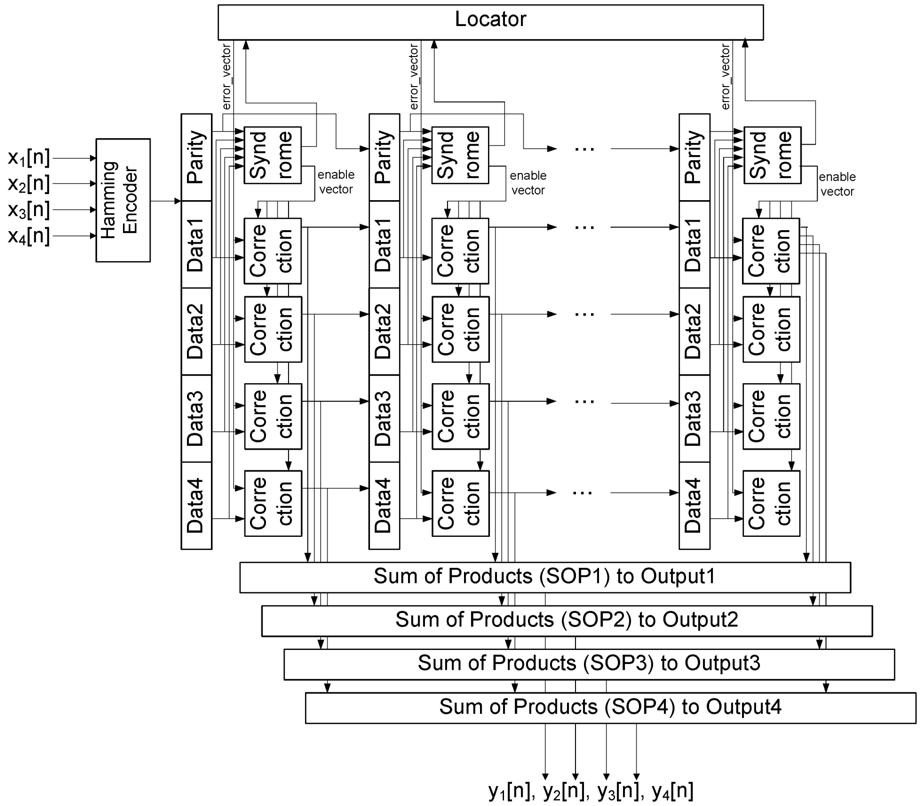

Fig. 7. Four parallel FIR filter with Hamming protection.

one register of that fourth tap, the error will never be successfully corrected. Finally, the remaining proposed techniques will operate without errors, even if single events occur in the data bits at each clock cycle. However, if one SEU hits the parity bits of the coded word stored in the first tap, and during the next cycle another SEU occurs in the data bits of the second stage, the error will not be corrected. Nevertheless, in most cases, the probability of occurrence of single events a few clock cycles apart are negligible and therefore, the protection provided by the proposed alternatives is similar to the traditional Hamming technique.

B. Complexity

The area cost in equivalent gates and in VIRTEX-5 slices of the evaluated protection techniques are provided in Table I and Table II, respectively.

As it can be seen when comparing the Hamming-based techniques, the shared decoder (Fig. 5) offers the most competitive results, considering the combination of protection effectiveness and the total number of gates. The calculation of the area relation in percentage has been done in reference to the standard Hamming implementation. It can be seen that for 5 taps the Hamming shared decoder version uses 8.41% less area in comparison to the standard Hamming version and 18.25% in comparison to the TMR version. This reduction is even larger when increasing the filter taps up to 11 with 10.45% in comparison to standard Hamming but reduced by 17.16% in comparison to TMR.

Table II shows the FPGA synthesis results for all techniques, highlighting that the Hamming shared decoder gives the best performance in terms of slice usage. Comparing the synthesis results to the one of the ordinary Hamming, it achieves overall slice savings of 11.86% for 5 taps and a 14.97% for 11 taps.

However, the operation frequency of the FIR filter protected using TMR is better because of the reduced combinational logic added to the critical path. Results of the synthesis for the parallel FIR filters with Hamming protection and the proposed techniques have been gathered in Table III for ASIC synthesis and Table IV for FPGA synthesis.

In the ASIC area results, a pattern of complexity reduction can be identified when increasing the data bits from 8 to 16 or 32 respectively, between ordinary Hamming and TMR. This increase is even more significant when comparing the results with that of the Hamming shared decoder. In the case of 32 data bits, the difference between the proposed technique and ordinary Hamming is 14.3% and 24.07% comparing against TMR. The synthesis results of the 11-tap filter are similar. The savings of the proposed techniques are 14.0% and 21.7% against the ordinary Hamming and TMR, respectively.

Furthermore, it can be observed that when increasing the data bits the reduction is significant, but when increasing the filter taps the reduction is smaller. This decrease is because the ordinary Hamming complexity is converging to TMR with the increase of the filter taps, as the percentage of the triplicated flip-flops in comparison to the combinational logic in the whole design is changing. This can be seen in the figures of the TMR overhead in comparison to ordinary Hamming of 12.9% for 5

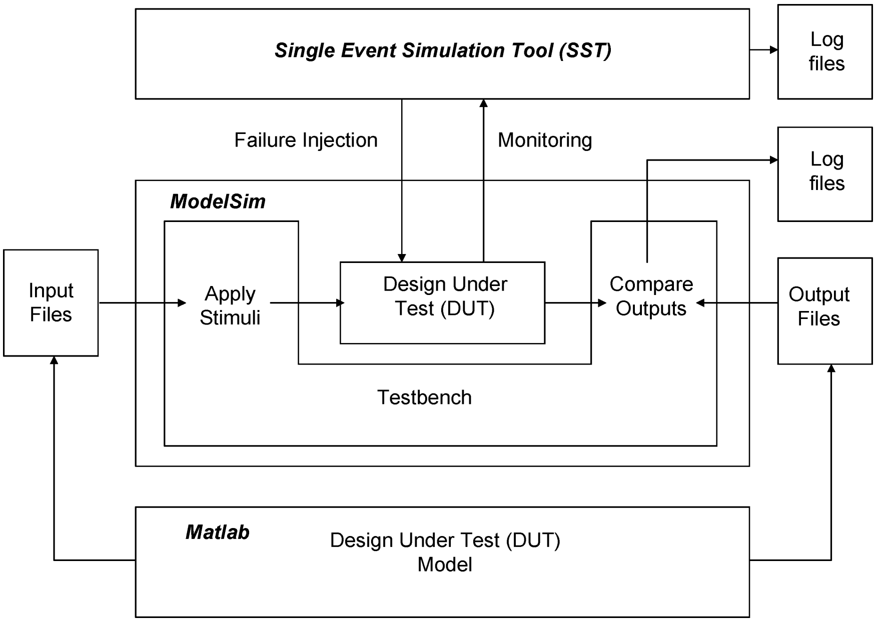

Fig. 8. Simulation environment block diagram.

TABLE I
ASIC SYNTHESIS RESULTS FOR ALL TECHNIQUES (8 bits) WITH PERCENTAGE OF AREA COMPARED TO HAMMING

ASIC - TSMC 0,25 µm		FIR Filter 5-Taps			FIR Filter 11-Taps		
		Clk [MHz]	Gates	%	Clk [MHz]	Gates	%
without protection	Fig. 1	145	755	-45.25	109	1,915	-41.67
TMR		127	1,545	12.04	100	3,549	8.10
Hamming	Fig. 2	121	1,379	----------	92	3,283	----------
Hamming single encoder	Fig. 3	117	1,293	-6.24	93	3,014	-8.19
Hamming additional data protection	Fig. 4	119	1,266	-8.19	90	3,038	-7.46
Hamming shared decoder	Fig. 5	114	1,263	**-8.41**	90	2,940	**-10.45**

taps and 9.8% for 11 taps, but the decrease from ordinary Hamming to the proposed technique from 14.3% to 14.0% is almost insignificant.

The VIRTEX results are as competitive as the ones discussed before for Table II. The decrement in the usage of slices when comparing the proposed technique to the ordinary Hamming is growing proportionally with the increase of the data bits, as seen in Table IV, where it changes for 5-taps from 11.86% to 11.92% (for 8 and 32 data bits respectively). The value of 21.17% for the version of 16 data bits is a relevant improvement and can be explained with the rigid structure of the VIRTEX-5. The reduction of the results for the 11-taps versions decreases from 14.97% to 8.18% (for 8 and 32 data bits), respectively. This decrement of the results for the 32-bit version of the 11-taps is due to the extra amount of combinational logic added to the Hamming shared decoder, which converges much faster for FPGA than for ASIC.

C. Performance

In the previous subsection, the area cost in equivalent gates and in VIRTEX-5 slices of the evaluated protection techniques were in comparison to the ordinary Hamming and TMR. The comparison shows that the Hamming shared decoder is the most competitive in respect of area costs both in an ASIC implementation and in a FPGA implementation.

Although the proposed Hamming shared decoder achieves the most competitive area cost results, it also suffers from a maximum clock frequency penalty. The impact of the penalty depends on the target architecture (ASICs or FPGAs).

This penalty is produced when extra logic is added to the design, increasing the critical path and therefore, reducing the maximum clock frequency that is allowed. This is shown in Table I for the ASIC versions, comparing the results of the plain FIR filter with 145 MHz, the TMR with 127 MHz, the ordinary Hamming with 121 MHz and the Hamming shared decoder with 114 MHz. The difference between the ordinary Hamming and the Hamming shared decoder is minimal in comparison to the achieved area cost reduction. This can be further observed in Table III, with the results obtained when doubling or quadrupling the bits or incrementing the taps.

For the FPGA results, interconnections have to be taken into account when analyzing the performance results of the synthesis. In other words, adding more combinational logic also means increasing the usage of slices, which have to be interconnected correctly and therefore, adding additional wire delay to the critical path. This can be observed in Table II and Table IV when comparing the TMR performance results to the ordinary

TABLE II
FPGA SYNTHESIS RESULTS FOR ALL TECHNIQUES

FPGA - VIRTEX 5	FIR Filter 5-Taps					FIR Filter 11-Taps				
	Clk [MHz]	Slices	FF	LUT	%	Clk [MHz]	Slices	FF	LUT	%
without protection	383	33	48	105	-44.07	371	83	96	263	-43.54
TMR	365	77	144	238	30.51	355	195	288	538	32.65
Hamming	353	59	72	208	----------	346	147	144	474	----------
Hamming single encoder	338	57	72	178	-3.39	329	132	144	405	-10.20
Hamming additional data protection	341	53	72	188	-10.17	320	129	144	430	-12.24
Hamming shared decoder	327	52	72	183	**-11.86**	221	125	144	423	**-14.97**

TABLE III
ASIC SYNTHESIS RESULTS WITH PERCENTAGE OF AREA COMPARED TO HAMMING FOR 8 bits, 16 bits AND 32 bits

ASIC - TSMC 0,25 µm		FIR Filter 5-Taps			Fir Filter 11-Taps		
		Clk [MHz]	Gates	%	Clk [MHz]	Gates	%
8 bits	Hamming	121	1,379	--------	92	3,283	--------
	TMR	136	1,487	7.80	100	3,549	8.10
	shared decoder	114	1,263	-8.40	90	2,940	-10.45
16 bits	Hamming	108	2,784	--------	93	6,556	--------
	TMR	126	3,104	11.50	82	7,073	7.90
	shared decoder	101	2,492	-10.50	79	5,851	-10.80
32 bits	Hamming	104	5,494	--------	79	12,882	--------
	TMR	129	6,204	12.90	93	14,146	9.80
	shared decoder	101	4,711	-14.30	72	11,076	-14.00

TABLE IV
FPGA SYNTHESIS RESULTS FOR 8 bits, 16 bits AND 32 bits

FPGA - VIRTEX 5		FIR Filter 5-Taps					FIR Filter 11-Taps				
		Clk [MHz]	Slices	FF	LUT	%	Clk [MHz]	Slices	FF	LUT	%
8 bits	Hamming	353	59	72	208		346	147	144	474	
	TMR	365	77	144	238	30.51	245	195	288	535	32.65
	shared decoder	327	52	72	183	-11.86	221	125	144	392	-14.97
16 bits	Hamming	222	137	126	450		185	306	252	1020	
	TMR	364	234	288	516	70.80	241	512	576	1165	67.32
	shared decoder	203	108	126	383	-21.17	184	278	252	866	-9.15
32 bits	Hamming	183	260	228	897		189	587	456	1987	
	TMR	364	456	576	1023	75.38	219	929	1152	2308	58.26
	shared decoder	176	229	228	773	-11.92	138	539	456	1775	-8.18

Hamming results. Same as for the ASIC versions, the difference between the ordinary Hamming and the Hamming shared decoder in comparison to the slice usage reduction, is minimal.

IV. CONCLUSIONS

Different approaches to protect generic FIR filters using Hamming codes against single events have been presented and put in perspective for ASICs and FPGAs. It has been shown that through understanding the system, enhancements to the design can be carried out to provide effective protection in exchange of a small frequency penalty.

It has been proved that the Hamming shared decoder is the most competitive solution based on area cost and performance. For example, assuming an 8-bit, 11-tap design, the area savings of this approach range from 10.45% (ASICs) to 14.97% (FPGAs).

Future work includes the research of FPGA-oriented solutions for fault tolerant digital filters and the consideration of power consumption as a metric for optimization, since it is a key factor in space applications.

REFERENCES

[1] A. H. Johnston, “Scaling and technology issues for soft error rates,” in *Proc. 4th Ann. Research Conf. Reliability*, Oct. 2000.

[2] R. D. Schrimpf and D. M. Fleetwood*, Radiation Effects and Soft Errors in Integrated Circuits and Electronic Devices*. Singapore: World Scientific, 2004.

[3] P. E. Dodd and L. L. Massengill, “Basic mechanisms and modeling of single-event upset in digital microelectronics,” *IEEE Trans. Nucl. Sci.*, vol. 50, no. 3, pp. 583–602, Jun. 2003.

[4] M. P. Baze, S. P. Buchner, and D. Mcmorrow, “A digital CMOS technique for SEU hardening,” *IEEE Trans. Nucl. Sci.*, vol. 47, no. 6, pp. 2603–2608, Dec. 2000.

[5] S. M. Parkes, “DSP (Demanding Space-based Processing!): The path behind and the road ahead,” in *Proc. 6th Int. Workshop Digital Signal Processing Techniques for Space Applications*, Noordwijk, The Netherlands, Sept. 1998.

[6] 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," in *Proc. 15th Symp. Integrated Circuits and Systems Design*, 2002, pp. 95–100.

[7] F. Lima Kastensmidt, L. Sterpone, L. Carro, and M. Sonza Reorda, "On the optimal design of triple modular redundancy logic for SRAM-based FPGAs," in *Proc. Conf. Design, Automation and Test in Europe*, 2005, vol. 2, pp. 1290–1295.

[8] P. Reviriego, J. A. Maestro, and O. Ruano, "Efficient protection techniques against SEUs for adaptive filters: An echo canceller case study," *IEEE Trans. Nucl. Sci.*, vol. 55, no. 3, pp. 1700–1707, Jun. 2008.

[9] B. Shim, N. R. Shanbhag, and S. Lee, "Energy-efficient soft error-tolerant digital signal processing," in *Signals, Systems and Computers Conf. Rec. 27th Asilomar*, 2003, pp. 1493–1497.

[10] E. Perahia, "IEEE 802.11n development: History, process, and technology," *IEEE Comm.*, vol. 46, no. 7, Jul. 2008.

[11] J. Huang and R. R. Spencer, "The design of analog front ends for 1000 base-T receivers," *IEEE Trans. Circuits Syst.—II: Analog and Digital Signal Processing*, vol. 50, no. 10, Oct. 2003.

[12] Mentor Graphics Web Site Online, February 2010 [Online]. Available: http://www.mentor.com/company/higher_ed/asic/

[13] A. Cosoroaba and F. Rivoallon, Achieving Higher System Performance With the Virtex-5 Family of FPGAs Feb. 2010 [Online]. Available: http://www.xilinx.com/support/documentation/white_papers/wp245.pdf

[14] D. Gonzalez-Gutierrez, Single Event Upset Simulation Tool Functional Description ESA Report TEC-EDM/DCC-SST2, Jul. 2004.

[15] O. Ruano, J. A. Maestro, and P. Reviriego, "Performance analysis and improvements for a simulation-based fault injection platform," in *Proc. IEEE Int. Symp. Industrial Electronics (ISIE'08)*, Cambridge, U.K., Jun. 2008, pp. 2299–2304.