# Efficient Protection Techniques Against SEUs for Adaptive Filters: An Echo Canceller Case Study

P. Reviriego, *Member, IEEE*, J. A. Maestro, *Member, IEEE*, and O. Ruano

***Abstract*—In this paper, novel protection techniques against soft errors for adaptive filters are presented. The new techniques are based on the use of system knowledge in terms of both filter structure and functionality, as well as the application tolerance to soft errors. Adaptive filters by nature recover from soft errors on their coefficients, but in existing implementations the recovery time can exceed what is acceptable for many applications. The proposed techniques dramatically reduce the recovery time after a soft error with an acceptable increment on circuit complexity, as they rely on reusing existing logic. To illustrate these techniques, a case study is presented in which their effectiveness is evaluated using a software-based fault injection platform. Also, their complexity is estimated in terms of the number of equivalent gates generated for the synthesized circuit implementation using a commercial ASIC library.**

***Index Terms*—Adaptive filters, digital filters, radiation hardening, redundancy, single event upsets (SEUs).**

## I. Introduction

In some environments, for example Space, radiation sources are abundant. The effects of radiation are a well-known cause of errors in microelectronic circuits [1]. These errors range from temporary failures of the system (which most of the times produce a restart of the operations) to serious and permanent damage of the devices. One type of temporary effects are Single Event Effects (SEEs), that cause changes in the values of flips-flops (SEUs) or combinational logic (SETs) [2]. Several techniques have been mentioned in literature in order to mitigate the effects of SEEs at the physical level [3]. Another alternative is to use redundancy in the design, so that it can detect and correct these temporary failures [4]. For example, Triple Modular Redundancy (TMR), which triplicates the flip-flops in the design and adds logic to vote in case of conflict, can be used to protect against SEUs. SETs, which are a particular case of error transients, may also be mitigated by using the so-called Functional Triple Modular Redundancy [4] (FTMR, which also triplicates the combinational logic). One advantage of both TMR and FTMR is that they are general techniques that can be applied to most digital circuits. However, this comes at a high cost in terms of circuit area and power and more so for FTMR.

Manuscript received September 6, 2007; revised January 15, 2008. This work was supported by the Spanish Ministry of Science and Education under Grant ESP-2006-04163.

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

Digital Object Identifier 10.1109/TNS.2008.924053

A different approach to deal with SEEs that results in a lower implementation cost will be presented in this paper. This approach is based on the study of *system knowledge*, by which the inherent redundancy or fault tolerance of some circuits is detected, thus generating *circuit specific protection techniques*. The advantage of this, is the production of custom-tailored solutions for each family of circuits, with good protection levels and a quasi-optimal implementation, something that cannot be achieved with general techniques like TMR.

This system knowledge approach has been applied to matrix operations [5], Fast Fourier Transform computation [6], [7] and simple filter structures [8]. Alternative approaches like processing the signals in the sigma-delta domain have also been proposed for simple filters [9]. In [10] and [11], more general approaches that can be used in a wider number of signal processing applications have been proposed. However, to the best of the authors' knowledge, there is no previous work on similar specific techniques to protect adaptive filters. The only related work [12] focuses on the analysis of the effects of SEUs on different adaptation algorithms in order to evaluate their robustness when exposed to radiation. Nevertheless, no protection techniques are proposed.

Adaptive filters [13] are however good candidates to apply this methodology, as some of their elements can be used effectively to provide protection against soft errors, and they are used in a variety of applications, as it will be seen in the next section.

The rest of the paper is structured as follows: first, some of the related work in adaptive filter implementation is reviewed. Then, in order to set the basis for the discussion, a case study is presented and used to illustrate the techniques. Finally, a detailed evaluation of the proposed techniques for the case study is done covering both protection effectiveness and relative complexity versus the unprotected filter and other protection techniques, like TMR.

## II. Related Work

Adaptive filters [13] are by nature resilient to isolated errors. For example, an error in a coefficient of an adaptive filter would be corrected after some time by the adaptation of the circuit. To date, however, most adaptive filter designs have not fully exploited these capabilities, since the focus has not been to protect the filters from the effects of SEUs but to minimize area, power consumption, convergence time to final coefficient values and noise due to the adaptation process. The first two factors have a direct impact on the final cost of the system and therefore are a primary objective of any design. For the other two, the required values are normally given by the performance in terms of startup time and Signal to Noise Ratio (SNR) that the system has to meet.

A better understanding of the tradeoffs involved in the design of adaptive filters can be gained by examining an example. Adaptive filters are extensively used in Gigabit Ethernet receivers for equalization, echo and crosstalk cancellation [14]. The performance requirements in terms of Bit Error Rate (BER) and the channels over which the receiver has to operate (specified in the standard) determine to a large extent the number of taps that are needed in each of the filters, which in most cases account for a significant part of area and power consumption of the receiver. Therefore, any optimization to those filters has the potential to reduce the overall cost of the receiver. A number of techniques have been developed over the last decades to that end. For example, in [15], a partial update of the coefficients is proposed, so that the adaptation logic is minimized, and in [16], [17], approaches for adaptive echo cancellation that exploit the fact that in many cases the echo response is sparse, are presented. The time allowed for filter adaptation is also given by the standard and influences the choice of the adaptation algorithm to be used. For example, in [18], a variable adaptation step is proposed for the sign algorithm, which will be used in our case study in order to reduce convergence time. This strategy allows a fast convergence time while minimizing the adaptation self noise in steady state where the channel is expected to change slowly. Convergence time is also important as it allows further use of partial update approaches [15] for adaptation logic optimization.

In this section, a few examples of the many algorithms and implementations that have been reported in the literature for adaptive filters have been presented. The objective was to illustrate the criteria that have guided adaptive filter implementations in order to provide the background for the case study that will be presented in the next section and in which some of those ideas are used. The reader can go to [13] for additional references and a complete coverage of adaptive filters.

## III. Case Study

As mentioned in the introduction, a case study is going to be used to discuss the proposed techniques and also to evaluate their effectiveness in a realistic design. The following describes the filter that will be used for that purpose.

The selected filter in this case study is an adaptive echo canceller. Echo cancellation is required in many wireline communications systems like xDSL [17], ISDN and voiceband modems [19] and in Ethernet Transceivers [20], and it is normally implemented by means of an adaptive filter. Echo originates from full duplex transmission over a two-wire cable, and interferes with the signal received from the remote transmitter, degrading the signal to noise ratio. The purpose of echo cancellation is to remove the echo from the received signal. In many cases, the echo is a linear function of the transmitted signal and cancellation can be achieved using a linear filter that reproduces the echo response and subtracts it from the received signal. The echo response can be estimated using an adaptive filter that would adapt to the particular response for each channel configuration.

In our case, the echo canceller will employ the sign algorithm for adaptation using a partial update approach and a two-level

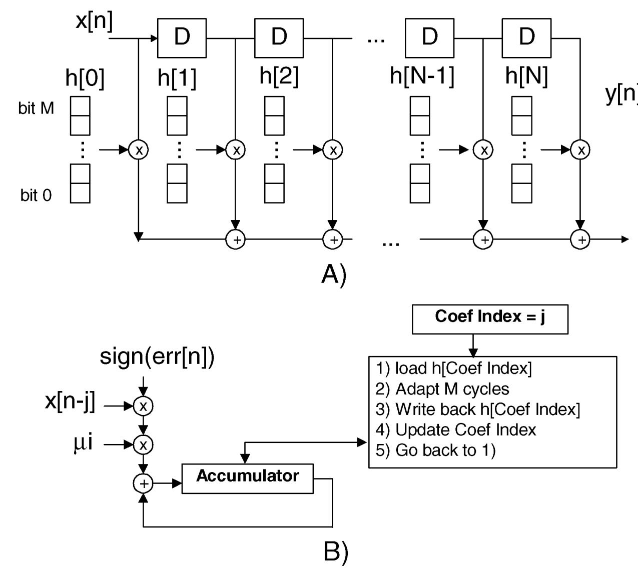


Fig. 1. Case study: Echo canceller implementation. (a) Filter implementation; (b) adaptation logic.

transmit signal. All this together enables a very efficient implementation of the canceller, as it will be shown in the following. This implementation efficiency is what has guided most designs so far, and therefore it is a good example in order to evaluate protection techniques in terms of both effectiveness and implementation cost overhead versus the unprotected filter.

The basic structure of the echo canceller is shown in Fig. 1. The upper part shows the filter implementation and the lower one the adaptation logic that in our case will be shared among all the filter taps. In this way, each coefficient $h[i]$ will be loaded into the accumulator, adapted for a number of cycles and then written back to the coefficient. This process is then carried out for the next coefficient in an iterative way. The simple sign algorithm will be used for adaptation by taking the sign of the error and the transmitted signal value, which in our case has only two levels, $-1$ and $+1$. The filter itself, can be implemented very efficiently as the multipliers just invert the sign of the coefficient or leave it unchanged.

The problem can be defined as follows: An echo noise is present in the line, and needs to be removed. This can be formalized as (1), where the coefficients $h_e[i]$ characterize the echo, and are unknown magnitudes.

$$e[n] = \sum_{i=1}^{N} h_e[i] \cdot x[n-i]. \tag{1}$$

The canceller can be characterized as (2). Now, $h[i]$ are the coefficients of the filter that have to be controlled, with the purpose that the canceller output $y$ is equal to the echo $e$ (i.e., the canceller models the behaviour of the echo).

$$y[n] = \sum_{i=1}^{N} h[i] \cdot x[n-i]. \tag{2}$$

Once this is achieved, if the canceller output $y$ is subtracted from the communication line, the effect is that the echo is removed (since the canceller output has the same magnitude of this error). In this way, the error $err[n]$ after cancellation, assuming for simplicity that there is no other signal apart from the echo at the input to the echo canceller, will be

$$err[n] = e[n] - y[n]. \tag{3}$$

The adaptation algorithm determines how coefficients $h$ are adapted in such a way that they "emulate" the echo coefficients $h_e$. As mentioned before, the simple *sign algorithm* can be used. This works as follows: First, determine if the error induced by the echo is positive or negative (this is achieved by simply examining the communication line). Then, modify $h$ (the canceller coefficients) by adding or subtracting (depending on the error sign) a fraction of the input $x$. After this, check again the output and verify that the error has effectively decreased. If this is repeated iteratively, this error will tend to zero. This can be seen as progressively trying to compensate the echo noise with a complementary signal produced by the canceller. This process on $h$ can be formalized as:

$$h[i, j+1] = h[i, j] + \mu_k \cdot sign\,(err[j]) \cdot x[j-i] \tag{4}$$

where $\mu_{\mathrm{k}}$ is called "adaptation gain" and represents the fraction of $x$ that is added or subtracted in each iteration. This is a design parameter, and it usually takes two values: one larger for initial adaptation and one smaller for steady state adaptation. In this case study, the parameters are as follows: the number of canceller taps $N$ is 20, and as mentioned before, there is only one adaptation logic block for the whole filter, thanks to partial adaptation. The bit-widths of the taps are 8 and 13 for the accumulator, and the adaptation gains $\mu_{\mathrm{k}}$ are $2^{-8}$ for initial adaptation (with a block size or time used to adapt each coefficient of 32 cycles), and $2^{-12}$ for steady state adaptation (using a block size of 256 cycles).

## IV. Proposed Techniques

As we have already mentioned, adaptive filters would recover from errors as part of the adaptation process. The problem is that in steady state the adaptation speed is normally slow in order to minimize the adaptation self noise. Adaptation can also be slowed down in steady state to reduce power consumption by running the adaptation one out of $K$ cycles so that the adaptation logic is effectively idle most of the time. Irrespective of the reason why the adaptation speed is slow, it would result in a long recovery time after an SEU on a coefficient bit, as illustrated in Fig. 2 for an SEU on bit 5 of h[9] during cycle 40000. In this case, let us assume that the coefficient being adapted at cycle 40000 is h[0] (adaptation is sequential and cyclic through all the coefficients). In this way, the recovery time will be an intermediate case between the best case (adaptation on h[8], next one would be h[9]) and the worst case (adaptation on h[10]).

As it can be seen in this figure, the echo is not cancelled for a large number of cycles (around 25000 in this case). This is a problem since it would force a restart or other abnormal conditions in many systems. However, if we had a mechanism to detect that an SEU has occurred, we could speed up adaptation and therefore reduce dramatically the recovery time using the already existing logic in the filter. This is the main idea behind our protection techniques that are elaborated in the following.

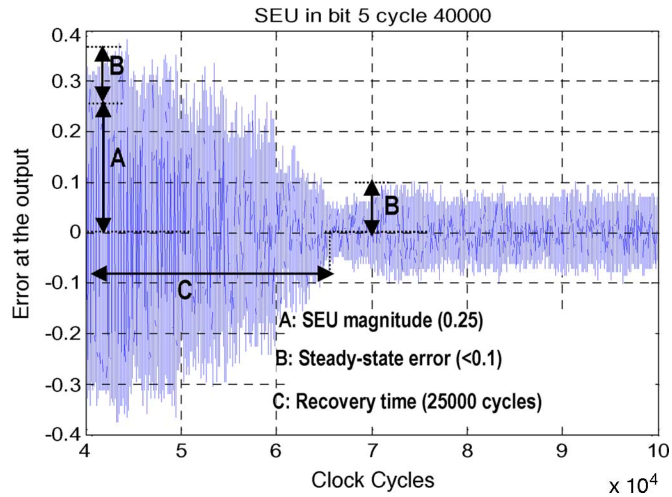


Fig. 2. Recovery time after an SEU on h[9] for an unprotected filter.

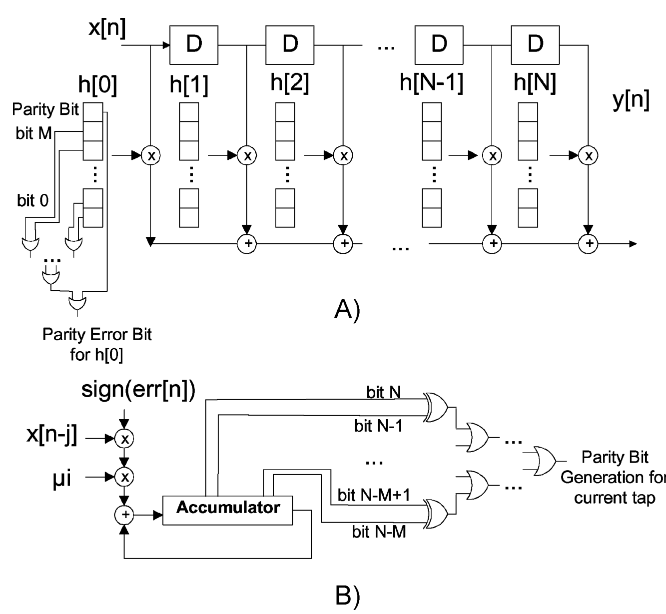


Fig. 3. Filter implementation with parity and error checking added (only shown for h[0]). (a) Filter implementation; (b) adaptation logic.

### A. Fast Selective Adaptation Technique

One way to detect that an SEU has occurred in one of the echo canceller coefficients is to add a parity bit to each of these coefficients, and then check the parity back in each clock cycle. The parity bit can be computed as part of the adaptation logic for coefficient updates while the checking is done in place for each coefficient, as illustrated Fig. 3.

Once we have a method for detecting that an SEU has occurred, we can use that information to speed up adaptation. In our case, this is done in two ways: first, we can increase the adaptation step (fast adaptation) and second we can adapt only the coefficient in which a parity error has been observed, which would be the one in which the SEU occurred (selective adaptation).

With this simple technique we can see that the recovery time after an SEU is dramatically reduced (to less than 500 cycles) as shown in Fig. 4.

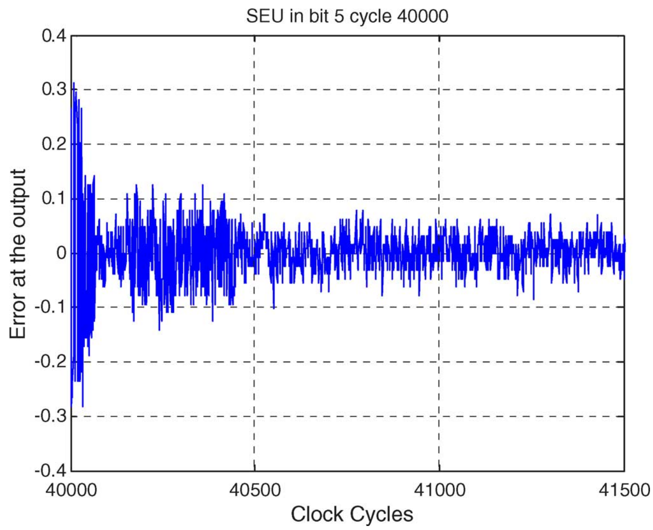


Fig. 4. Recovery time after an SEU on h[9] for the fast selective adaptation technique.

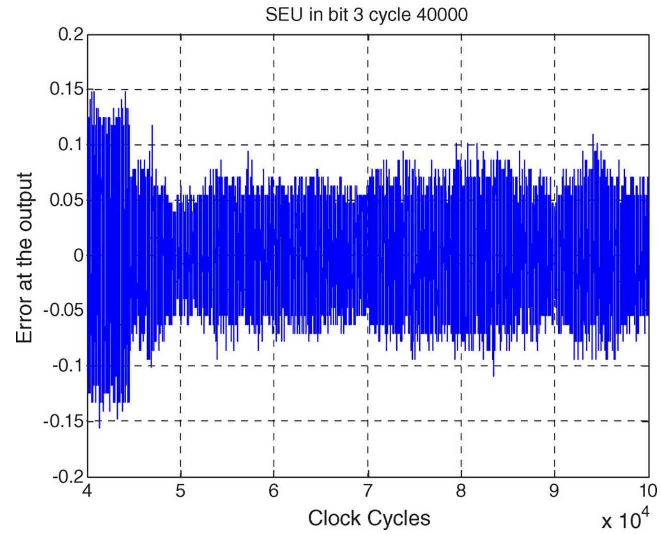


Fig. 5. Recovery time after an SEU on h[9] for an unprotected filter.

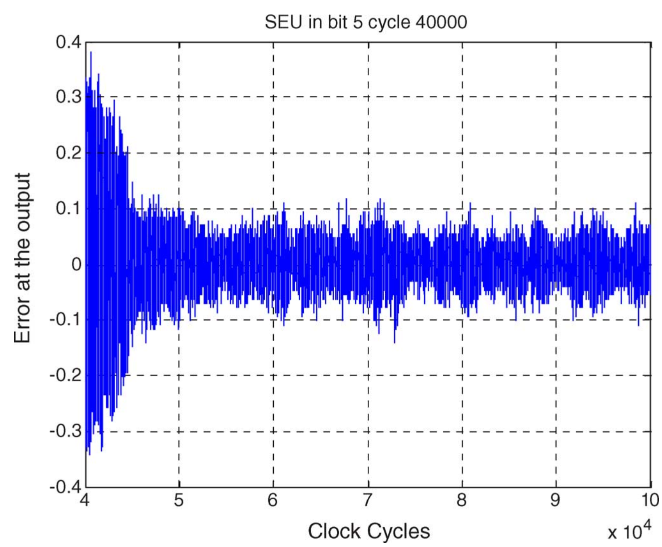


Fig. 6. Recovery time after an SEU when adapting all taps with a larger step.

A further analysis of the different SEU scenarios leads to the conclusion that the effects on the system can be very different depending on which bit the error happens. For example, an error in the most significant bit (MSB) will result in a large error while an error in the least significant bit (LSB) will cause an error that may not be noticeable. This motivates a discussion about which bits should be protected in a given protection technique and which bits can remain unprotected, since they would not produce a meaningful error.

In the case of the technique under study, the use of the larger adaptation gain will result in a certain transient error. If this error is larger than the one caused by an SEU (e.g. if the SEU occurs in one of the least significant bits), then this technique will not be effective, since it would be better to correct the error through the normal adaptation process. On the other hand, the use of a smaller adaptation gain will increase the recovery time for large errors such as those observed if an SEU occurs in one of the upper bits of the registers. After examining these scenarios, the following criteria are proposed to make the protection technique as efficient as possible:

- If the error in the system is large enough, then use the large adaptation gain to reduce the recovery time. An error will be considered large if it affects all but the three least significant bits.
- If the error in the system can be considered small (i.e., it affects the three least significant bits), then do not apply the protection mechanism. Although these bits are left unprotected, the error present in the system will be removed by the normal adaptation, as shown in Fig. 5.

The case in which an SEU affects the parity bits must also be considered. For this technique, it will trigger the adaptation of the coefficient that corresponds to the affected parity bit. Since that coefficient is actually correct, this process would result in a slight increase of the transient error for a short number of cycles.

The ideas behind this technique (namely upon the detection of an SEU, the increment of the gain and the adaptation of the coefficient that has been affected) can be applied to a wide class of adaptive filters of which the case study presented is just an example.

### B. Direct Error Correction Technique

An alternative way of detecting that an SEU has occurred would be to monitor the output of the echo canceller and if a sudden increase of the error is observed, then this can be attributed to an SEU. However, there are two problems with such an approach. The first one is that there may be other effects apart from an SEU on a coefficient that can cause a similar behavior (for example an SEU on the transmitted data delay line or a genuine change in the echo response). The second problem is that even if the error increment source was an SEU, we would not have information of the coefficient that was affected and therefore we would need to adapt all the coefficients, increasing the recovery time significantly as shown in Fig. 6 (around 5000 cycles).

TABLE I
ERROR RANGES FOR EACH BIT

| | Lower Limit | Upper Limit |
|---|---|---|
| Bit 7 | 0.6 | - |
| Bit 6 | 0.4 | 0.6 |
| Bit 5 | 0.2 | 0.4 |
| Bit 4 | 0.15 | 0.2 |

Although we have seen that the monitoring of the error at the output of the echo canceller is not a reliable method for the detection of SEUs, it can be very helpful in conjunction with the parity bits detection method that we proposed for the Fast Selective Adaptation Technique. The idea is as follows: the parity bits are used to detect that an SEU has effectively occurred and the register that has been affected; then the estimated error is used to deduce which bit in that register was flipped by the SEU. To illustrate the method we can observe the error at the output of the canceller in Fig. 2 during the first 5000 cycles. This corresponds to the effect of the SEU, which in this case has a magnitude of 0.25 plus the steady state error. By examining this value, we can deduce which bit in the coefficient was affected by the SEU, which in this case is bit 5 (considering a fixed-point two's complement representation, bit $i$ will have a weight of $2^{i-L+1}$, being $L$ the number of bits of the representation, 8 in our case). If the SEU has affected one of the most significant bits in the register then we would expect the error caused by the SEU to be much larger than the steady state error so that from the observed error we can infer the affected bit and simply flip it back to its right value. The errors on the most significant bits are in fact the ones for which recovery time is critical as they cause a larger error at the filter output. In the case of the least significant bits, steady state adaptation may be sufficient to ensure that the system performance does not degrade significantly. In the case under study, given that the steady state error level is below 0.1 as observed in Fig. 2, we propose to apply the technique to the four upper bits of each coefficient. Given the magnitude of the error, a table can be used to easily find out the bit affected by the SEU (see Table I).

This technique is illustrated in Fig. 7, where it can be seen that it is extremely efficient in terms of recovery time for our case study.

Like in the first technique, the case in which an SEU affects the parity bits must also be considered. For this technique, it will have no effect as long as the error stays in the steady state levels and will be corrected on the first adaptation after the SEU of the coefficient that corresponds to the affected parity bit.

This technique, although very effective in terms of time to recover from an SEU, requires the input signal to the adaptive filter to take only two values of the same magnitude and different sign ($+/-1$ in our case study). When the input signal takes an arbitrary number of values, as in an adaptive equalizer, the technique could be extended by observing the maximum error in a time interval and using that value to infer the affected bit. However, even in that case it would lead to a less effective and reliable technique, as it can be shown by a typical communication system example (see [21] for a background on this subject). Consider an adaptive equalizer whose input signal is adjusted to a given root mean square (RMS) value. The peak

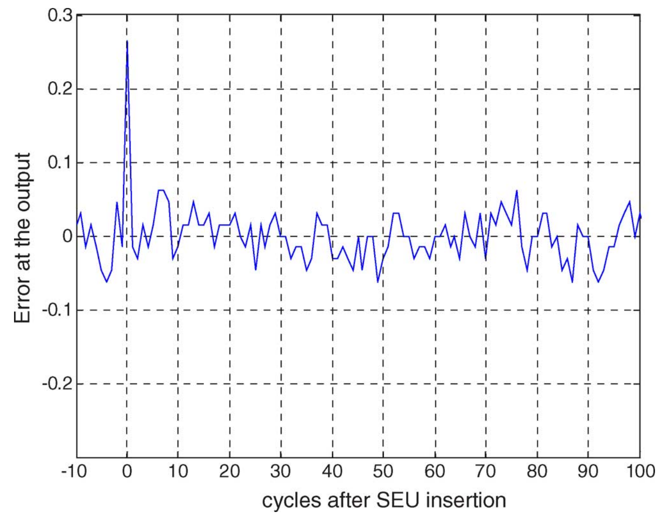


Fig. 7. Recovery time after an SEU on h[9] for the direct error correction technique.

to root mean square value of the input signal will change from channel to channel. Let us assume that the two extreme values are two and four. In the first case, an error in a coefficient will cause a maximum error at the output of twice the RMS value, while in the second case the error will be four times the RMS value. This proves that an arbitrary input signal would clearly cause problems to the Direct Error Correction Technique. For these reasons, this option is not explored further in this paper. Instead, an alternative technique is presented in the following section.

## C. Iterative Error Correction Technique

The previous technique assumes that the input signal will only have two levels: $+1$ and $-1$. That makes the calculation of the error magnitude easy. However, this constraint does not always apply to real cases.

The proposed Direct Error Correction Technique can be modified so that the basic principle is applicable to other adaptive filters, for example adaptive equalizers, in which the signal that passes through the filter takes arbitrary values and therefore the error caused by an SEU in one of the coefficients on the filter output can not be determined a priori.

In that case, a possibility to solve this is to sequentially test all the coefficient bits until the one affected by the SEU is located. Although, this technique is not fully explored in terms of implementation cost, a simple example is presented to better illustrate it. Let us assume that the input signal to the adaptive echo canceller of the case study is changed so that it takes arbitrary 8-bit values. The technique is then applied to the four upper bits such that when an SEU is detected, the error is observed over $K$ cycles, and as soon as it exceeds a given threshold, the most significant bit is inverted. Then, the error is again observed over $K$ cycles, and if the error is below the threshold that means the bit affected by the SEU has been corrected. If it is not, then the bit that was inverted is set back to its previous value and the operation is repeated for the next bit. The technique ends when either all bits have been inverted (what would indicate that the SEU had affected the parity bit) or one of the bit-flips has resulted in

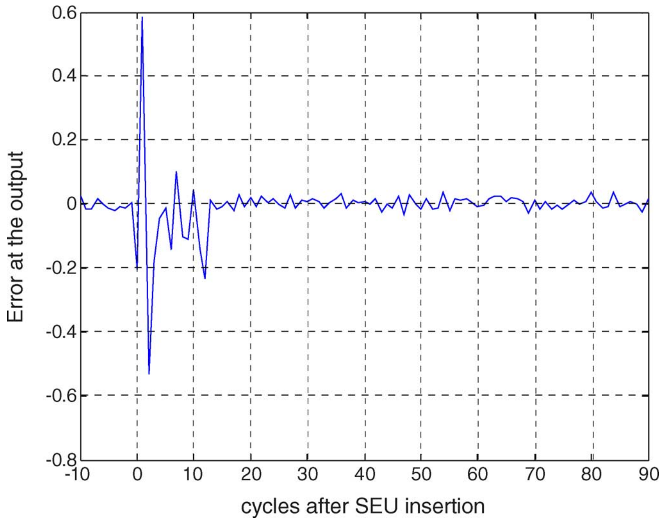


Fig. 8. Recovery time after an SEU on h[9] for the iterative error correction technique.

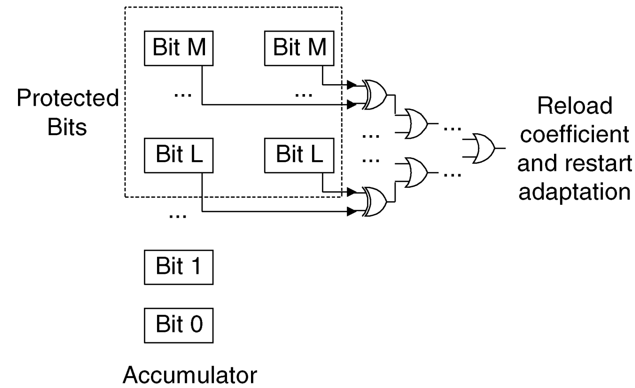


Fig. 9. Proposed technique to protect the adaptation logic.

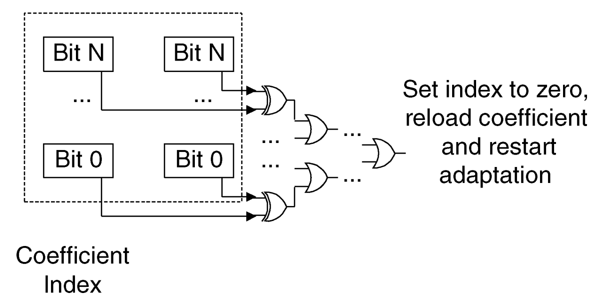


Fig. 10. Protection of the coefficient index.

an error that is below the threshold. The technique is illustrated in Fig. 8, where the error threshold is set to 0.2 and the SEU occurs in the bit that corresponds to an error of magnitude 0.25. The bit inversion starts with the more significant bit and results in an error that exceeds the threshold in the first cycles. So the next bit is inverted and again the threshold is exceeded. Therefore, the inversion is finally done on the bit that corresponds to 0.25 (which is bit 5) and the error then stays below the threshold, what makes the technique end. All this process takes less than fifteen clock cycles in this case.

### *D. Protecting the Adaptation Logic*

So far, the proposed techniques have focused on protecting the filter coefficients. However, the adaptation logic can also be affected by SEUs. To protect it, the obvious alternative would be to use TMR in the upper bits of the accumulator (those that are written back to the coefficient once an adaptation cycle is completed). But by using the system knowledge we can in fact propose a better alternative. Those bits could be just duplicated, adding also the logic to detect a difference between the duplicates. If such a difference occurs, that would mean that an SEU has affected one of the registers. By reloading that coefficient and restarting the adaptation cycle again, the problem would be solved. With this, the information stored in the system is used in order to avoid triplication. The overall approach is illustrated in Fig. 9.

A similar approach can be used for the register that stores the index of the coefficient that is being adapted. In this case, if a difference is detected between the two duplicated versions, the register could be cleared to index zero and the adaptation process restarted. The coefficient being processed when the problem was detected should not be updated.

With this approach, the worst effect of an SEU in the coefficient index would be that an adaptation cycle would be lost and that adaptation would resume in coefficient h[0], instead of the one being treated when the SEU arrived. Both of those effects will not have any impact on the error at the output of the canceller. The approach is illustrated in Fig. 10.

## V. Quality Analysis of the Protection Techniques

In this section, the quality of the presented techniques will be measured. These techniques have been implemented in VHDL and then synthesized for a commercial ASIC library. Two experiments have been carried out on the circuits:

1) Using a software fault injection simulation platform, (see [22]–[24] for more details), several SEUs campaigns have been inserted, and the effectiveness of the protection techniques has been put in perspective. More than 20 test scenarios were reenacted. In total, over 2000 impacts of SEUs were analyzed.
2) The circuits have been synthesized, and their complexity has been compared with the traditional protection techniques.

With those experiments, the quality of the proposed techniques is measured in effectiveness and complexity.

### *A. Effectiveness*

To evaluate the effectiveness of the protection techniques, we have inserted random SEUs in the filter registers and observed the effect on the filter error. The performance is quantified in number of cycles to get back to a level of error that is in the range of the steady state error. That is effectively the number of cycles for which a degraded behavior would be observed in the system. Inserting the SEUs in all taps, it has been checked that the tap number in which the SEU occurs does not influence

TABLE II
RECOVERY TIME (IN CLOCK CYCLES) AFTER AN SEU FOR THE DIFFERENT PROTECTION ALTERNATIVES AND BIT POSITIONS

| | Unprotected | Fast Selective Adaptation | Direct Error Correction |
|---|---|---|---|
| Bit 7 | >60000 | 5000 | 1 |
| Bit 6 | 50000 | 400 | 1 |
| Bit 5 | 22000 | 400 | 1 |
| Bit 4 | 10000 | 400 | 1 |
| Bit 3 | 5000 | 400 | 5000 |

TABLE III
IMPLEMENTATION COST FOR THE DIFFERENT ALTERNATIVES

| | Number of Gates | Cost Increment |
|---|---|---|
| Unprotected | 4174 | -- |
| Fast Selective Adaptation | 5023 | 20.3% |
| Direct Error Correction | 5182 | 24.1% |
| (A) TMR upper four bits | 5917 | 41.8% |
| (B) TMR upper five bits | 6205 | 48.7% |
| (C) TMR on all registers | 7627 | 82.7% |

the observed performance for the proposed techniques, as expected. For the unprotected filter, the recovery time depends on which coefficient is being adapted when the SEU occurs, so the recovery time for the average case is reported. The results for each technique and register bit (MSB to LSB) are summarized in Table II. The results are only presented for the five upper bits which are the ones in which an SEU produces a significant error at the output.

It can be seen that the Fast Selective Adaptation technique provides a reduction in the recovery time of up to two orders of magnitude. This reduction is obtained thanks to the immediate adaptation of a coefficient as soon as an SEU is detected in it, through the parity bit logic.

On the other hand, the Direct Error Correction technique results in an immediate correction of the error. By inferring the bit that was affected by the SEU (analyzing the system error magnitude), the problem is isolated in a single cycle, and can be restored to its right value with a simple bit-flip. It is also worth noticing that bit 3 is not protected in the case of Direct Error Correction and therefore the recovery time is the same as for the unprotected filter.

SEUs have also been injected on the adaptation logic to ensure that they produce no errors on the canceller output, as it was discussed in previous sections. Thanks to the protection logic introduced, none of these errors have affected the behavior of the system.

### *B. Complexity*

In order to compare the complexity of the proposed techniques, two figures of merit will be used. The first one is the relative increment in the total number of gates of the proposed techniques versus the unprotected filter. This will give an indication of the overhead required to provide protection. The second is to compare the proposed techniques with TMR. Although TMR clearly provides a superior protection threshold, not all the applications demand the highest fault tolerance level. In some cases, an intermediate protection could be enough and using TMR would result in an unnecessary high cost. For example, when the filter is used in a receiver and the SEU rate is low, if the recovery time is small enough the system may still meet the BER specifications. In these situations, the use of non-standard techniques, as the presented ones, provides a less expensive alternative to TMR.

The results are presented in Table III. The Fast Selective Adaptation technique results in an increment of around 20% (vs. the unprotected filter) in terms of area with a negligible impact on the filter speed. As it was shown, the reduction of the recovery time for this technique can be of two orders of magnitude. So, a large reduction on the impact of SEUs is obtained with a limited cost increase.

On the other hand, the results for the Direct Error Correction technique are similar: an increment of 24% in terms of area with a negligible impact on the filter speed. This adds just a 4% over the Fast Selective Adaptation technique, but with a much better recovery time, as shown in Table II.

In order to compare the proposed techniques with TMR, three cases are considered. Using system knowledge we can apply TMR only to the coefficient (upper bits) and adaptation logic registers. In the first case (A), TMR is applied to the four upper bits of the coefficient registers (the same bits that are protected by the Direct Error Correction technique). In the second case (B), TMR is used in the five upper bits of the mentioned coefficients. Finally, in the third case (C), TMR is applied to all the registers in the filter (not only coefficients), what results in an increase of 82% in the number of gates.

The first observation that can be made is that the use of system knowledge results in substantial savings even when applying TMR. The second is that when each of those partial TMR implementations are compared with the Fast Selective Adaptation and the Direct Error Correction techniques respectively, the proposed techniques still result in a significantly lower complexity.

The protection technique to use (if any) will depend on the worst case SEU environment envisaged for the device or system and on the application tolerance to the effects of SEUs (in our case to temporary errors at the echo canceller output).

## VI. CONCLUSIONS AND FUTURE WORK

In this paper, new techniques to protect adaptive filter implementations from the effects of SEUs have been presented. These techniques exploit both application and system knowledge in order to provide a more intelligent protection that results in a lower circuit complexity compared to TMR, as it has been shown in the paper. The proposed techniques have also been applied to an adaptive echo canceller using a simulation environment that enables a flexible testing of the effects of SEUs on signal processing circuits.

As future work, the presented techniques will be applied to other adaptive filters. The first step will be to study their application to adaptive equalizers. This will be especially useful in the case of the Iterative Error Correction technique that has not been fully explored in the present work and that is well suited for such kind of filters.

The proposed techniques could also be used for adaptive filters that are implemented in software on a Digital Signal Processor. In that case, the filter coefficients will likely be stored in memory, which sometimes is protected by parity bits. In that situation, the parity protection will be provided by the memory itself, while the rest of the technique can be implemented in software and executed upon detecting a parity error. In fact, the Iterative Error Correction technique can be enhanced in a software implementation by computing the errors for all bit positions simultaneously and selecting the one that brings the error at the filter output to the steady state level. That can be done with few additional operations, as we just need to recalculate the failing coefficient contribution for each bit position, but not the rest of the filter coefficient contributions to the output. In that case, a more appropriate name for the technique would be Parallel Error Correction.

More generally, the same broad idea of using system knowledge to derive protection techniques can also be applied to other types of filters like generic IIR filters, filter banks, etc.

## ACKNOWLEDGMENT

The authors would like to thank P. Reyes for her contribution to some of the experiments that are reported in this paper.

## REFERENCES

[1] R. D. Schrimpf and D. M. Fleetwood, *Radiation Effects and Soft Errors in Integrated Circuits and Electronic Devices*. Singapore: World Scientific, 2004, 981-238-940-7.

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

[3] 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.

[4] S. Hanbic, FTMR: Functional Triple Modular Redundancy ESA Report, FPGA-003-001, Dec. 2002.

[5] K. Huang and J. A. Abraham, "Algorithm-based fault tolerance for matrix operations," *IEEE Trans. Comput.*, vol. 33, no. 6, pp. 518–528, Jun. 1984.

[6] A. Reddy and P. Banerjee, "Algorithm-based fault detection for signal processing applications," *IEEE Trans. Comput.*, vol. 39, no. 10, pp. 1304–1308, Jun. 1984.

[7] S. Wang and N. K. Jha, "Algorithm-based fault tolerance for FFT networks," *IEEE Trans. Comput.*, vol. 43, no. 7, pp. 849–854, Jul. 1994.

[8] P. Reyes, P. Reviriego, J. A. Maestro, and O. Ruano, "New protection techniques against SEUs for moving average filters in a radiation environment," *IEEE Trans. Nucl. Sci.*, vol. 54, no. 4, pp. 957–964, Aug. 2007.

[9] E. Schuler and L. Carro, "Reliable digital circuits design using sigma-delta modulated signals," in *Proc. 20th IEEE Int. Symp. Defect and Fault Tolerance in VLSI Systems*, Monterey, CA, 2005, pp. 314–324.

[10] A. Chatterjee and M. A. d'Abreu, "The design of fault-tolerant linear digital state variable systems: Theory and techniques," *IEEE Trans. Comput.*, vol. 42, no. 7, pp. 794–808, Jul. 1993.

[11] B. Shim, N. R. Shanbhag, and S. Lee, "Energy-efficient soft error-tolerant digital signal processing," in *Conf. Rec. 37th Asilomar Conf. Signals, Systems and Computers*, Nov. 2003, vol. 2, pp. 1493–1497.

[12] J. B. Destro-Filho and D. W. Matolak, "Effects of single-event upsets on satellite communications: Issues for blind equalizer design," in *Proc. Radiation and Its Effects on Components and Systems Conf.*, Grenoble, France, Sep. 2001, pp. 292–295.

[13] S. Haykin, *Adaptive Filter Theory*, 4th ed. Englewood Cliffs, NJ: Prentice-Hall, 2002, 013-090-126-1.

[14] *Physical Layer Parameters and Specifications for 1000 Mb/s Operation Over 4-Pair of Category-5 Balanced Copper Cabling, Type 1000-Base-T*, IEEE Standard 802.3ab, 1999.

[15] K. Dogancay and O. Tanrikulu, "Adaptive filtering algorithms with selective partial updates," *IEEE Trans. Circuits Syst. II*, vol. 48, no. 8, pp. 762–769, Aug. 2001.

[16] A. Sugiyama, S. Ikeda, and A. Hirano, "A fast convergence algorithm for sparse-tap adaptive FIR filters identifying an unknown number of dispersive regions," *IEEE Trans. Signal Process.*, vol. 50, no. 12, pp. 3008–3017, Dec. 2002.

[17] L. Shou-Sheu and W. Wen-Rong, "A low-complexity adaptive echo canceller for xDSL applications," *IEEE Trans. Signal Process.*, vol. 52, no. 5, pp. 1461–1465, May 2004.

[18] A. Kanemasa and K. Niwa, "An adaptive-step sign algorithm for fast convergence of a data echo canceller," *IEEE Trans. Commun.*, vol. 35, no. 10, pp. 1102–1108, Oct. 1987.

[19] K. Murano, S. Unagami, and F. Amano, "Echo cancellation and applications," *IEEE Commun. Mag.*, vol. 28, no. 1, pp. 49–55, Jan. 1990.

[20] H. Jingyu and R. R. Spencer, "The design of analog front ends for 1000BASE-T receivers analog and digital signal processing," *IEEE Trans. Circuits Syst. II*, vol. 50, no. 10, pp. 675–684, Oct. 2003.

[21] J. Kurzweil, *An Introduction to Digital Communications*. New York: Wiley, 2000, 047-135-258-6.

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

[23] Computer Architecture and Technology Group, Universidad Antonio de Nebrija [Online]. Available: http://www.nebrija.es/~jmaestro/esa

[24] 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," in *Proc. IEEE Int. Symp. Industrial Electronics*, Vigo, Spain, 2007, pp. 3316–3321.