Efficient Soft Error-Tolerant Adaptive Equalizers

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

***Abstract*—Soft errors are becoming an increasingly important issue for circuit reliability. Traditional techniques to protect against soft errors, like triple modular redundancy (TMR), have a large cost in terms of area and power. This has motivated the development of specific protection techniques for various types of circuits. In this paper, techniques to protect adaptive filters are presented, which provide reasonable reliability with reduced cost and power consumption. An adaptive equalizer case study is used to discuss and evaluate the proposed techniques in terms of both protection and cost.**

***Index Terms*—Adaptive filters, digital filters, fault tolerance, soft errors.**

I. Introduction

THERE are certain environments, for example, space, where radiation sources are abundant. The effects of that radiation are a well-known cause of errors in microelectronic circuits [1]. These errors range from temporary failures of the system to serious and permanent damage of the devices. One type of temporary effect is soft errors, which cause changes in the value of a logic element [2]. Those changes can have a permanent effect if the incorrect value propagates to a storage element like a flip-flop.

As technology scales and the devices become smaller, they are more sensitive to soft errors, which start to become a concern in ground-level applications and pose a major challenge for circuit design, as stated in the 2008 International Technology Roadmap for Semiconductors update [3]. The technology scaling also increases the probabilities of a soft error propagating to a storage element, as circuits tend to operate at higher clock speeds [4]. All these effects drive the need for protection techniques against soft errors.

Several methods have been proposed in literature to mitigate the effects of soft errors at the physical level [5]. Another alternative is to use redundancy in the design, so that it can detect and correct these temporary failures. The traditional approach, known as triple modular redundancy (TMR), has been to triplicate the elements that can suffer soft errors and add the voting logic to select the majority in case of an error. When soft errors can affect both combinational and sequential logic elements, the usual approach is to triplicate the entire block, which results in a large area and power consumption.

Manuscript received October 01, 2009; revised November 17, 2009; accepted December 06, 2009. Date of publication February 22, 2010; date of current version August 11, 2010. This work was supported by the Spanish Ministry of Science and Education under Grant AYA2009-13300-C03-01. This paper was recommended by Associate Editor N. K. Jha.

The authors are with the Universidad Antonio de Nebrija, 28040 Madrid, Spain (e-mail: previrie@nebrija.es; jmaestro@nebrija.es; sliu@nebrija.es).

Digital Object Identifier 10.1109/TCSI.2009.2039252

To reduce the area and power overhead associated with the protection, circuit-specific techniques have been proposed. For example, in [6], fault-tolerant techniques are addressed for matrix operations. Due to their use in a large number of applications and their performance requirements, signal processing circuits have been studied with special interest in the literature. In [7] and [8], protection techniques for fast Fourier transform computation are presented, while in [9], the protection of some simple filter structures against soft errors in storage elements is studied. In [10], a general approach that can be used to protect finite-impulse response (FIR) filters with fixed coefficients is discussed, which is able to protect against errors in both sequential and combinational logic elements.

Adaptive filters are an interesting type of circuits because they are used in a wide range of applications and, due to their adaptive nature, are well suited for circuit-specific protection techniques [11]. Previous work has focused on protecting adaptive filters from the effects of soft errors on sequential logic [12], but to the best of the authors' knowledge, no protection techniques for errors in both sequential and combinational logic have been proposed. The objective of this paper is to present general protection techniques for adaptive filters.

The remainder of this paper is structured as follows. First, some of the related work in both adaptive filter implementations and protection techniques for filters is reviewed to provide the foundation for the rest of the analysis. Then, to set the basis for the discussion of the techniques, a case study of an adaptive equalizer is presented and used to illustrate the proposed techniques. Finally, a detailed evaluation of the 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 on Adaptive Filters

In this section, a brief overview of the previous work on adaptive filters is presented to support the analysis of the protection techniques proposed later. The existing protection techniques for adaptive filters and constant-coefficient FIR filters are also reviewed, so that they can be compared with the proposed techniques.

Adaptive filters are used in many applications, and they have widely been studied in the literature. Our aim in this section is to provide a brief description of how an adaptive filter works and the main elements that are present in it. The reader can go to [11] for additional references and a complete coverage of adaptive filters. A block diagram of an adaptive filter is shown in Fig. 1. The filter is composed of two main blocks: a traditional FIR filter with coefficients $h[i]$, input signal $x[n]$, and output signal $y[n]$ and an adaptation logic that periodically updates the values of the coefficients $h[i]$. The inputs to the adaptation logic are the input signal to the filter and an error measure $e[n]$, which are

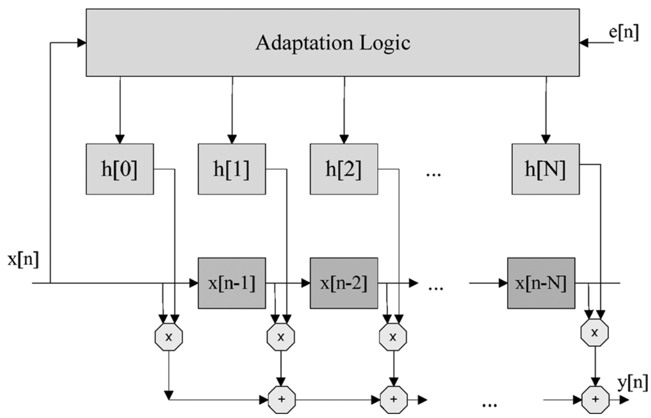

Fig. 1. Block diagram of an adaptive filter.

used to compute the new values of the coefficients. The error measure varies with the nature of the filter. For example, in an adaptive echo canceller, it will be the signal after cancellation that is the difference between the received signal and the output of the echo canceller [13], [14]. For an adaptive equalizer, it can be the difference between the symbol at the output of the slicer and the output of the equalizer [15]. The main idea behind adaptive filters is to gradually adapt the coefficient values to reduce the observed error $e[n]$.

The protection of FIR filters against soft errors has been studied from different perspectives. For example, in [16], a comparison of the cost of protecting FIR filters with TMR and Hamming codes is presented. In [9], protection techniques against soft errors in storage elements are proposed for some simple FIR filters, a work that is extended in [17] to cover general FIR filters. Techniques that also protect against soft errors in combinational logic have been proposed in [10], [18], and [19], where reduced precision replicas of the filter are used to detect and correct the errors. This approach is similar to TMR, but instead of tripling the design, more simple replicas are added. This has the obvious benefit of reducing area and power consumption at the cost of correcting the errors only up to the precision level of the added replicas. All the approaches discussed so far focus on the protection of the filter assuming that soft errors affect the input data stored in the filter delay line and the arithmetic logic but not the filter coefficients themselves. This is a reasonable assumption for a fixed-coefficient filter, but this is clearly not the case in adaptive filters, where coefficients change with time and are therefore normally stored in registers that can be affected by soft errors, as well as the combinational logic to update them.

For an adaptive filter, the effect of a soft error on a coefficient will be an error on the output that will last until the coefficient is readapted by the filter adaptation logic back to the correct value. This process, depending on the filter adaptation speed, can take a large number of clock cycles [12]. The effect of an error in the delay line that stores the values will be, in the worst case, an error in the output of up to N cycles, with N being the order of the filter. Therefore, an error on a coefficient will normally have a much larger impact on the system. This is the reason why previous studies have focused on protecting the filter from soft errors in the registers that store the coefficients [12]. However, errors in the adaptation logic may also cause errors in the coefficients, an issue that, to the best of the authors' knowledge, has not been investigated so far. The only other related work on the effects of soft errors on adaptive filters that we are aware of is presented in [20], where different adaptation algorithms for adaptive equalizers are evaluated in terms of their performance in the presence of soft errors, but no specific protection techniques are proposed.

The objective of this paper is to present new techniques that are able to protect from soft errors in the adaptation logic, register coefficients, data delay line registers, and filter arithmetic logic, thus covering all possible sources of errors and providing a comprehensive approach for adaptive filter protection.

III. Case Study: The Feedforward Equalizer

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.

An adaptive equalizer has been chosen as a case study because they are widely used in communication systems [21] and magnetic storage [22], among other applications. They also have a significant arithmetic complexity, and therefore, the protection of the combinational logic becomes an important issue.

A. *Description of the Feedforward Equalizer*

The main goal of an equalizer is to compensate the effects introduced by the transmission channel in the signal. In this way, the transmitted values are usually distorted before reaching the receiver due to the physical phenomena produced by cables and the environment. This is known as intersymbol interference (ISI), which poses a major problem to high-speed communication, as the transmitted symbols are spread by the channel and interfere with nearby symbols [21]. To solve this problem, the receiver performs an operation called equalization, whose goal is to rebuild the received signal and make it as close as possible to the initial one. There are different approaches to perform this operation, for example, Tomlinson–Harashima precoding (THP), which attempts to compensate the channel effect in the transmitter [23]–[25]. Other techniques like decision feedback equalization (DFE) and feedforward equalization (FFE) are implemented in the receiver [21]. In many systems, FFE is implemented in conjunction with either DFE or THP to achieve optimal results [15].

The case study will be an adaptive feedforward equalizer implemented with an adaptive FIR filter. The implementation will consist of the following modules.

1) FIR Filter: The core of the filter is the delay line, a group of N registers (taps) that shift input $x[n]$. Since this is a linear filter, the output of the filter $y[n]$ is a linear combination of the previous inputs

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

The parameters $h[i]$ are the coefficients, which will change to make $y[n]$ as close as possible to the correct value. The fact is that the input $x[n]$, once received through the channel, is not

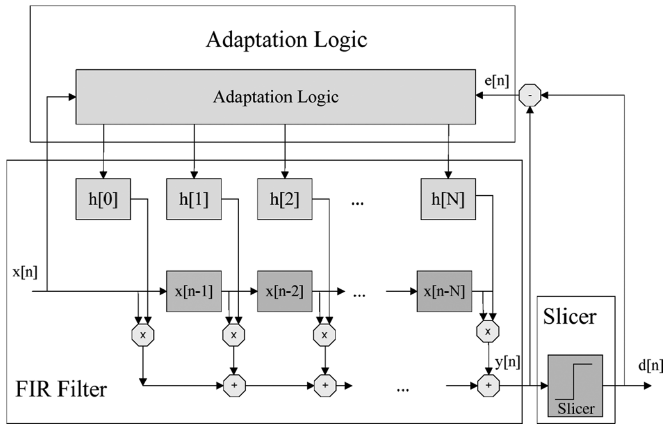

Fig. 2. Block diagram of an adaptive feedforward equalizer.

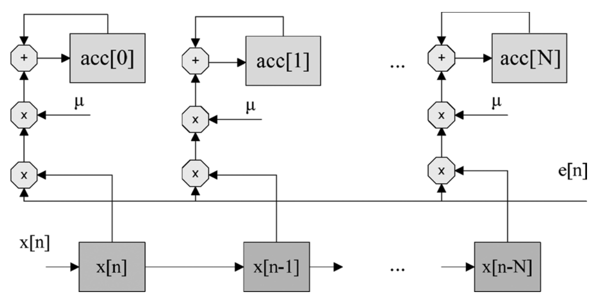

Fig. 3. Block diagram of the equalizer adaptation logic.

the same as the initial signal sent by the transmitter, due to the channel distortion. Therefore, coefficients $h[i]$ try to compensate this distortion, giving $y[n]$ as a result, which should be close enough to the initial signal.

2) Slicer: Signal $y[n]$ should be the same as the transmitted signal. However, due to the previous process of sum of products, it can adopt any given shape. The slicer checks the value of $y[n]$ at each cycle and rounds it to the closest predetermined level (in our case study, -1 or 1), producing signal $d[n]$, which is discrete and multivalued and is forwarded to the next module in the communication system.

3) Adaptation Logic: However, due to the previous rounding process, there will be a difference between $y[n]$ and $d[n]$, which is the error of the equalizer. The objective is that this error is as close as possible to zero. To achieve this, the error is forwarded to the adaptation logic block, which will dynamically recompute the coefficients to reduce the error at the output.

The block diagram of the equalizer working in steady state is shown in Fig. 2.

The details of the adaptation logic are shown in Fig. 3. For each coefficient $h[i]$, we have an accumulator $\mathrm{acc}[i]$, which is updated each cycle with the error estimate as follows:

$$\mathrm{acc}[i] = \mathrm{acc}[i] + \mu \cdot e[n] \cdot x[n-i]. \quad (2)$$

The idea is to compensate the channel distortion to progressively make this error null. The magnitude of the update is determined by μ, which is called the adaptation gain. Large values of μ are used in the start-up (during the initial training of the equalizer), and small ones are used in steady state. In most cases, μ takes a value of $2^{-\alpha}$ so that the multiplication becomes a simple shift operation.

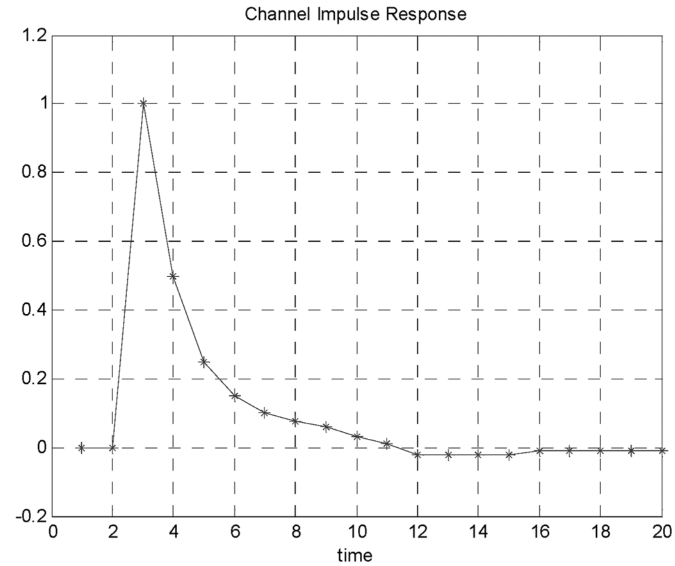

Fig. 4. Channel response used for the case study.

B. Implementation Details of the Feedforward Equalizer

The specific parameters of the case study are given as follows: the number of filter taps is 16, the coefficients are coded with 10 bits (only the upper seven are used for the replica block as explained later), and the incoming data are coded with 8 bits (only six are used for the replica). The adaptation gain during steady state is 2^{-16}, and the accumulators use 20 bits. For simplicity, a two-level signal is used at the transmitter. The channel presents a low-pass response typical in many communication systems, which is shown in the time domain in Fig. 4.

The equalizer response after initial adaptation is shown in Fig. 5, and the slicer error is shown in Fig. 6. Those would be the starting points for the protection analysis in the following sections.

C. Justification of the Need for Protection Techniques of the Feedforward Equalizer

From a soft-error protection point of view, we can focus on the steady state of the filter, since the system will spend most of the time in this state and an error during start-up will, in the worst case, cause a reattempt to start up.

There are two error scenarios:

1) An error in the data delay line will last a maximum of N cycles, with N being the number of taps in the equalizer. This would, in most systems, cause only some errors on the received symbols, as N is normally a relatively low number (less than 100 in most cases).
2) An error in a coefficient will persist until the adaptation process brings it back to the correct value. As we have already mentioned, adaptation in steady state is slow, as it only needs to track changes in the channel. A technique in [12] proposes to speed up adaptation when a soft error is detected. However, this strategy has the side effect of

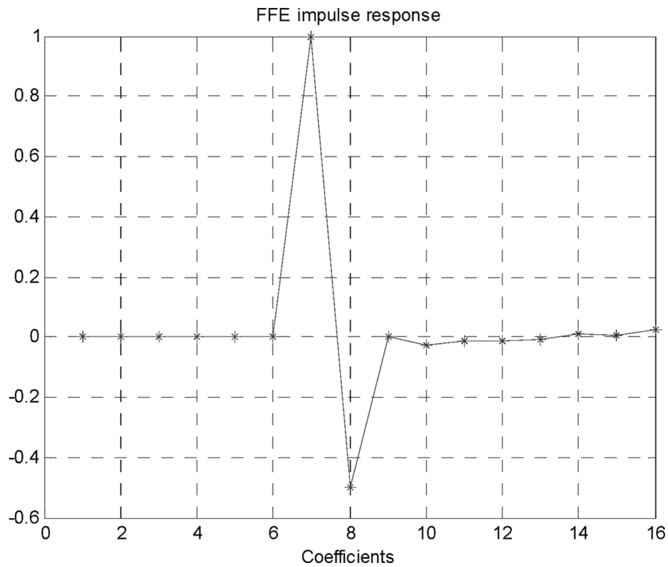

Fig. 5. Typical equalizer coefficients after initial adaptation.

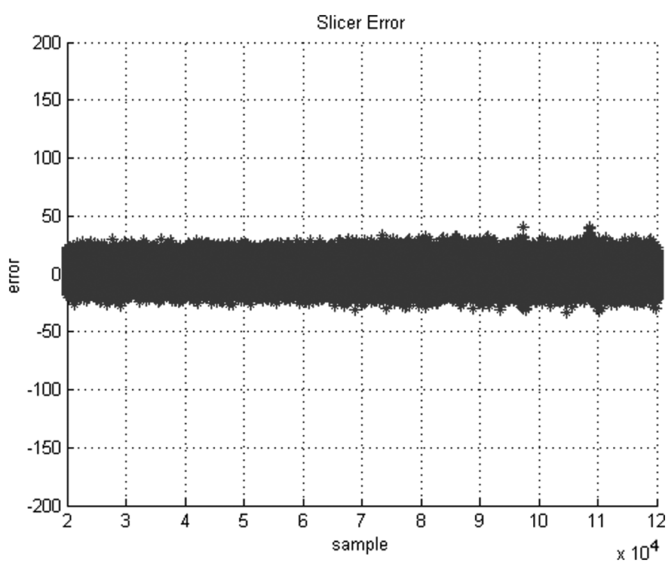

Fig. 6. Slicer error after initial adaptation.

temporarily increasing the adaptation noise, with only a moderate decrement in the recovery time.

Therefore, although the equalizer is able to recover itself from soft errors, the long time needed in this process may not be feasible for many applications.

Therefore, it would be interesting to explore different alternatives to protect both the filter and the adaptation logic, which do not imply a long recovery time. This will be explained in the next sections.

IV. Proposed Techniques

By focusing on which circuit elements are affected, there are two main categories for soft errors: 1) those that affect a flip-flop (or a similar storage element) and 2) those that affect combinational logic. The first forces us to protect the registers in the design, while the second implies that the combinational logic must also be protected.

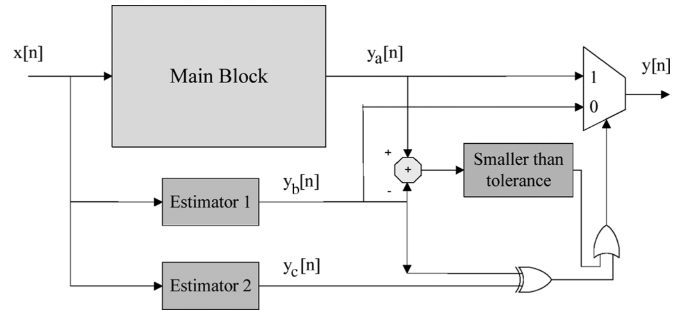

Fig. 7. Block diagram of the traditional ASET Technique.

Traditionally, errors in the registers have been the major concern, and that is the reason why many of the existing protection techniques focus on the protection of the filter registers only [9], [12], [16], [17]. However, errors on combinational logic are becoming increasingly important in new technologies [4], and therefore, the protection against this type of errors has recently been studied [10], [18], [19]. In this section, we will cover the protection against both types of errors.

A. FIR Filter Structure Protection

In [10], different algorithmic soft-error tolerance (ASET) variants are proposed that can be used to protect FIR filters from errors in the data delay line registers and in the arithmetic logic (but not against errors on the filter coefficients). The main idea behind the ASET techniques is the use of estimators of the filter output in parallel with the filter implementation so that when the output takes a value that deviates significantly from the estimators (which means that an error has occurred in the filter), the output of the estimators can be used instead. This reduces the effect of soft errors on the filter output at the expense of the accuracy of the estimator. The overall approach is shown in Fig. 7.

For the estimator, two alternatives are proposed: a reduced precision replica of the filter or a predictor. In both cases, it is assumed that the cost of the estimators is a fraction of the main block.

For our case study and, in fact, for many adaptive filters, the ASET techniques can be enhanced. In our case, instead of trying to detect and correct the errors at the output of the filter (as shown in Fig. 7), we can also replicate the slicers and apply detection and correction at their outputs.

The proposed approach is illustrated in Fig. 8. In this case, only one replica is added, and the symbols at the output of the slicers are compared. If there is a discrepancy, the symbol that corresponds to the output with the smaller error $e[n]$ is selected. The idea is that, during normal operation, the error at the slicer will be small to have an acceptable bit error rate, and therefore, if a soft error occurs, it is likely that the error on the affected block becomes considerably larger.

This approach might select the incorrect symbol if the soft error produces an output close to another slicing level. In our two-level symbol case study, multiple bits need to be simultaneously affected to produce this scenario, which is quite unlikely.

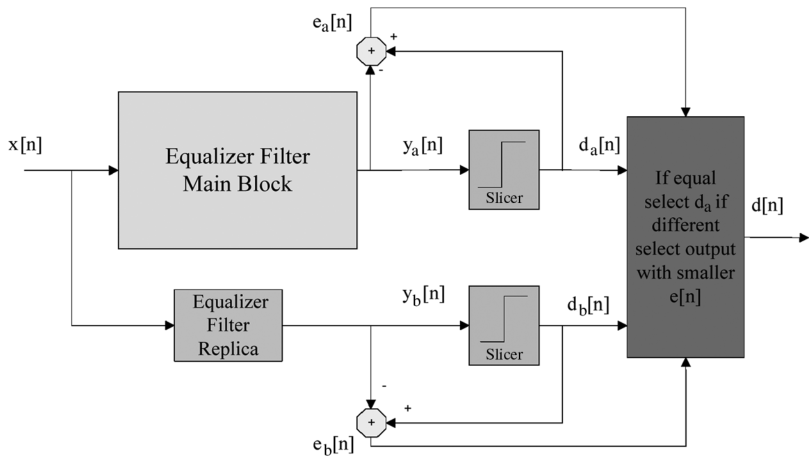

Fig. 8. Proposed ASET technique for the equalizer filter.

In summary, we can reduce the cost of the traditional ASET technique to protect the FIR structure just with a slight decrease in the protection level.

B. Adaptation Logic Protection

The presented ASET technique is not directly applicable to the adaptation logic, since a simplified replica of the adaptation logic is not straightforward to implement. The reason for this is that using reduced precision has implications for the adaptation algorithms, which would be slower or even stop working. Therefore, the replica of the adaptation logic will need to have a complexity similar to the one in the main block, leading to a large area and power overhead. In the rest of this section, alternative techniques to protect the adaptation logic are presented.

The adaptation logic shown in Fig. 3 is used to update the equalizer coefficients so that they adapt to the channel response. Soft errors on the sequential logic can affect the accumulators, which would result in incorrect values for the coefficients and create an error on the equalizer output that would last until the adaptation brings it back to the correct value. Soft errors on the combinational logic could also result in an error in the coefficient values if they propagate to the accumulator registers. To protect the accumulator registers, techniques similar to the ones proposed in [12] can be used. However, in this paper, the focus is on protection techniques that can deal with soft errors in both sequential and combinational logic.

The block diagram of the proposed technique is illustrated in Fig. 9. The main idea is to duplicate the most significant bits of the accumulators, as backup copies, and to use them in case an error is detected.

There are a number of options to perform the error detection. The proposed solution is to combine the adaptation logic protection with the previously proposed ASET technique for the filter structure. In this way, the accumulator backup copies will be the coefficients of the filter replica (see Fig. 11). In the next section, this solution will be explained in detail.

1) Protection Technique Algorithm: The replica coefficients have to be updated from time to time to keep a valid value in both.

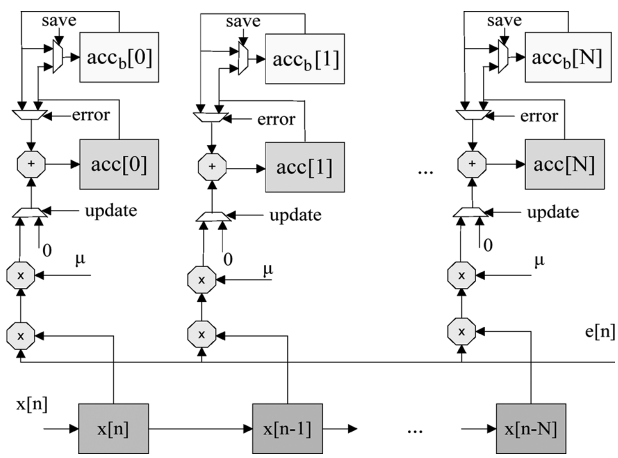

Fig. 9. Block diagram of the equalizer adaptation logic with the proposed protection. The accumulator acc_b is a copy with fewer bits (less precision) than the original accumulator, i.e., acc.

Let us define T1 as the update frequency, i.e., the number of cycles between two consecutive updates. Furthermore, let us define T2 as the number of cycles required to safely perform the update operation, during which the adaptation process is stopped. How to determine the most appropriate magnitude for these parameters will be discussed in Section IV-B-3. Then, the overall process is described as follows (see Fig. 10).

1) *Normal operation*: It consists of the coefficients being adapted, following the standard filter procedure. This happens during T1-T2 clock cycles.
2) *Update process*: At instant T1-T2, the update process is triggered. The first step consists of stopping the adaptation process for T2 cycles. These T2 cycles occur before the actual copy of the coefficients into the replicas. The most important detail in the protection technique is to guarantee that the coefficients' values are error-free when they are saved to the copies. Otherwise, soft errors would propagate through the filter. This is achieved by "disconnecting" the accumulator from the adaptation logic during T2 cycles. Once adaptation is stopped, errors on the adaptation logic have no effect. Soft errors that have occurred on the adaptation logic and have been registered in the accumulator

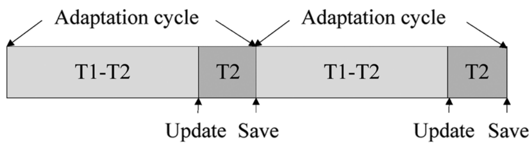

Fig. 10. Time diagram of the adaptation cycles, showing the normal operation (T1-T2) and the update process (T2).

will show at the output in less than T2 cycles. This would launch a recovery process, as explained later. Obviously, a detection mechanism for errors that have occurred directly on the accumulator registers is also needed (as they can be hit by an error even when adaptation is stopped). To that end, a parity bit could be added to each of the registers so that if a parity error is detected, the recovery phase is launched (see the next item). After T2 cycles of the update phase, the accumulator values are saved into the replicas.

3) *Recovery phase*: At any instant, the recovery phase may be triggered. This is an asynchronous process that will happen anytime an error is detected. This may be because a soft error has affected the accumulators (triggering the parity bit, as explained before) or because a soft error has been detected at the output of the filter (triggered by an $e_a[n]$ exceeding the normal threshold). The recovery process consists of restoring the value of the main accumulator with the content of the replica.

With this strategy, isolated errors will be corrected as soon as they occur, thus eliminating the effect on the system performance.

2) Isolated Errors in the Control Logic: Soft errors affecting the control logic can occur. However, these will not have any negative effect on the protection technique if soft errors arrive isolated.

1) An error on the logic that generates the update signal can trigger this phase. This will stop the adaptation before expected, but it will not produce any error at the output.
2) An error can activate the save signal, but that will only refresh the duplicates of the accumulator with no error at the output.
3) An error can trigger the recovery phase, causing a reload of the accumulators. This would perform an unnecessary operation, but once again, the loaded values will be valid, and therefore, they should cause no error on the received symbols.

3) T1 and T2 Parameters: The T1 and T2 parameters have been used throughout this section as a basic mechanism within the protection technique. However, nothing has been said yet about how to determine the value of these parameters.

Parameter T2 should be large enough to guarantee that a significant error in a coefficient can be detected at the output and eliminated. The value can be determined by simulation, using different channels and data sequences and adding a margin to the worst-case observed value.

Once T2 is known, T1 should be selected such that the adaptation has the desired speed, taking into account that only (T1-T2)/T1 of the cycles are adapted. In fact, there is another factor that limits the adaptation, which is the occurrence of soft errors. This is so because a soft error in the adaptation logic can cause a reload of the coefficients from the backup copies, therefore losing the adapted cycles. This effect would reduce the actual adaptation frequency, but it is negligible when the soft-error arrival rate is such that the probability of having at least one soft error in T1 clock cycles is much smaller than one.

V. Experimental Results

To evaluate the proposed techniques, an experimental environment has been set up, and several tests have been conducted.

The following configuration has been used to implement the feedforward equalizer.

1) The adaptation logic has been protected using the scheme illustrated in Fig. 9 (including a parity bit on the accumulators, as discussed before).
2) The accumulator values are also used as coefficients for both the main and the replica filter blocks that are implemented using the approach shown in Fig. 8.

The diagram of the configuration used in the evaluation is shown in Fig. 11.

The circuit was implemented in VHDL and synthesized for an ASIC library. The synthesized netlist was then used for the fault injection experiments using a software-based fault injection platform [26], [27], initially developed by the European Space Agency. This platform has intensively been tested with space applications that require high reliability.

In Fig. 12, the slicer errors for the main and replica blocks during steady-state operation are illustrated when no soft errors are inserted. It can be observed that the replica has a larger slicer error, as expected.

A. Reliability of the Protection Technique

In the first experiment, errors were inserted in an unprotected FFE. The effects of the slicer error are depicted in Fig. 13, showing a complete degradation of the system. It tries to adapt and correct the errors, but a new bitflip occurs before the previous one is corrected. The result is that thousands of symbols are received in error: 137 391 symbols were erroneous, which represent 12.84% of the total. This would normally force the system to restart the link.

In the second experiment, soft errors were injected on the protected version of the FFE. The injection is performed in such a way to ensure that there are more than T1 clock cycles (T1 is 1024 and T2 is 126 in this case) between consecutive soft errors (in other words, there is no more than one error per adapt–save cycle). In this case, there should be very few errors in the received symbols. The results are depicted in Fig. 14, showing how only isolated errors appear on the main slicer error, while for the replica, the errors can last until the coefficients are saved from the main block at the end of the cycle. In all of those cases, the system is able to select the output from the block that has not been affected by the soft error, and therefore, the behavior of the system is always correct. This shows the robustness of the approach, since from 1000 soft errors randomly inserted, none was propagated to the output.

To further study the percentage of errors that produce symbol errors, a large number of simulations were performed focusing

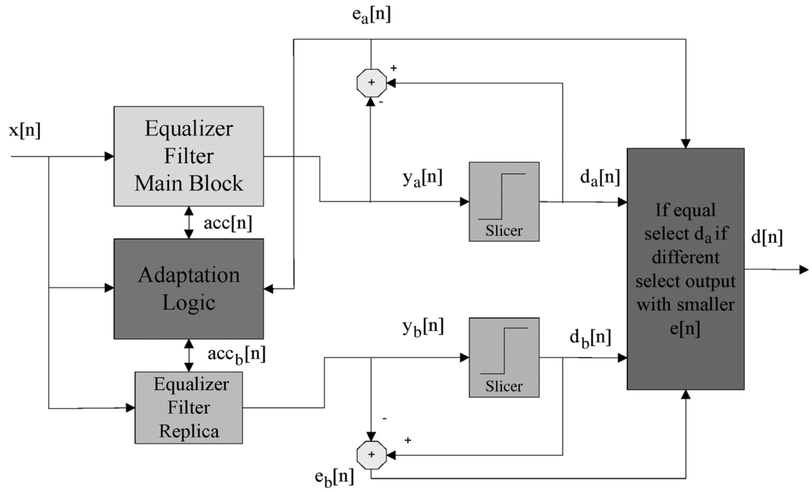

Fig. 11. Configuration used in the experiments.

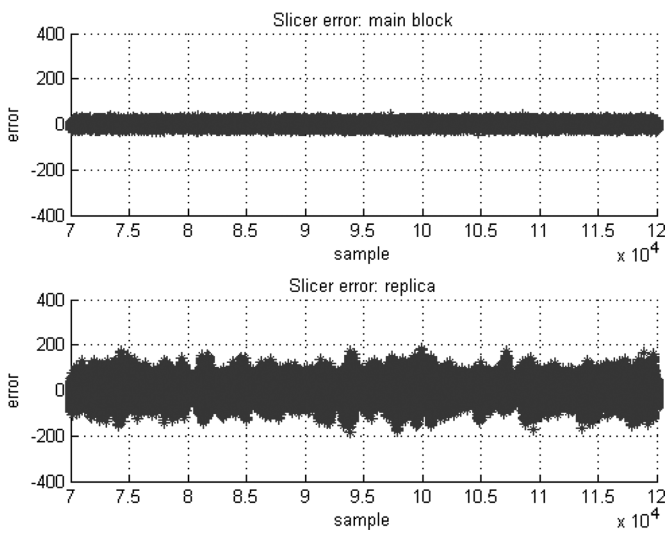

Fig. 12. Slicer errors for the main and replica blocks without soft errors.

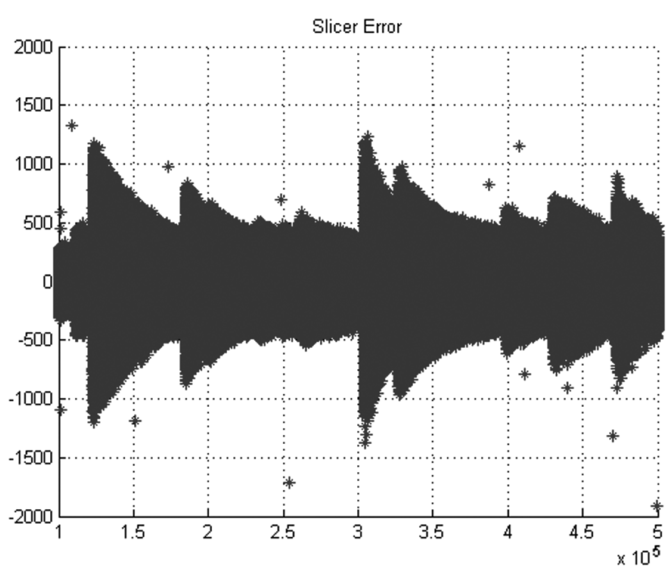

Fig. 13. Slicer error for an unprotected FFE when soft errors are inserted.

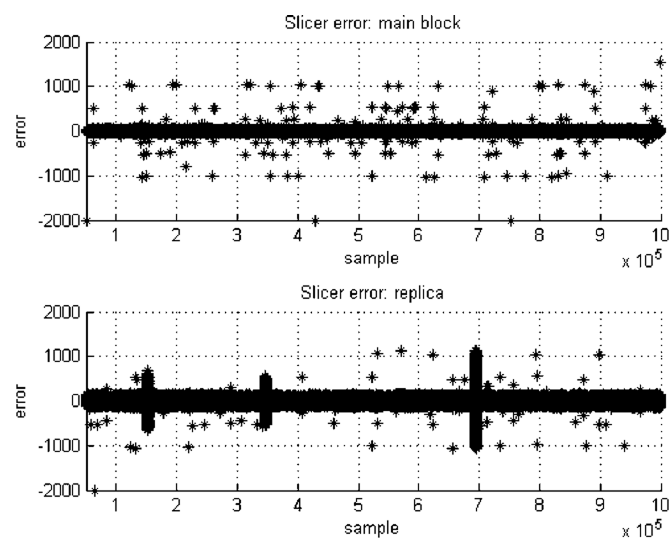

Fig. 14. Slicer errors for the main and replica blocks of a protected FFE with soft errors.

the error injection on the sum of products. The idea is to verify if a symbol error can effectively be produced in these circumstances. In this case, 20 000 soft errors were injected, and no symbol error was observed. This shows that the estimation assuming independent bitflips, as done in [10], may be conservative and that, in real designs, the errors will tend to be correlated. This reduces the probability that a soft error causes a symbol error, since several bits need to be flipped to achieve this.

In fact, even if errors can occur in the previous situation, they could be reduced by noting that most of them will affect the main block, as it has a larger area (and the area is proportional to the probability of suffering errors, for the same type of circuit). Therefore, if different symbols are produced and both slicer errors are similar, then the output from the replica block should be selected. Another option is to use the output from the replica block when the slicer error in the main block exceeds the expected range, even if it is smaller than the replica slicer error. Using those strategies, the failure rate could be further reduced if needed. These options have not further been explored as in the present design no errors were observed.

TABLE I
AREA ESTIMATES FOR THE DIFFERENT FFE IMPLEMENTATIONS

	Gate count	Ratio vs unprotected
Unprotected	49,490	1.00
Proposed Technique	55,862	1.13
ASET	62,642	1.27
TMR	148,478	3.00

B. Cost of the Protection Technique

Finally, it is worth considering the cost of the different protection techniques in terms of the circuit area. The number of equivalent gates for the different protection alternatives is shown in Table I. The following conclusions can be drawn from these results.

1) The proposed technique has an overhead significantly lower than ASET, and besides, the latter only protects against errors in the FIR structure. Therefore, the coefficients and adaptation logic remain unprotected. For all these reasons, the proposed technique proves to be a better choice than ASET.
2) TMR has a much larger overhead than the proposed technique, produced by the massive triplication. Both techniques offer a similar protection level for isolated errors. Therefore, the proposed technique is as efficient as TMR but with a lower cost.

These results can be extrapolated to power consumption, as most of the elements in the filter are active during all clock cycles. Therefore, the proposed technique would also significantly reduce the power consumption of the FFE.

From the previous analysis, it can be deduced that the proposed technique would be efficient in terms of area and power while providing a good protection level.

VI. CONCLUSION AND FUTURE WORK

In this paper, techniques to effectively protect adaptive filters against soft errors have been presented. The proposed techniques cover both errors on the filter structure and errors in the adaptation logic. In the first case, the proposed approach is more efficient than previous methods, and in the second, new techniques have been introduced that are able to protect against errors in the accumulators and in the adaptation logic.

A case study has been used to illustrate the effectiveness and cost of the proposed techniques and to compare them with existing approaches. The results have confirmed that the proposed protection is effective and can significantly reduce area and power consumption.

The presented approach can be applied to other types of adaptive filters, since, by nature, an adaptive filter output gives an indication of whether something unexpected has occurred.

REFERENCES

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

[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] *The International Technology Roadmap for Semiconductors Update 2008*. : [Online]. Available: http://www.itrs.net/

[4] P. E. Dodd, M. R. Shaneyfelt, J. A. Felix, and J. R. Schwank, “Production and propagation of single-event transients in high-speed digital logic ICs,” *IEEE Trans. Nucl. Sci.*, vol. 51, no. 6, pt. 2, pp. 3278–3284, Dec. 2004.

[5] 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.

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

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

[8] 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.

[9] 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, pt. 2, pp. 957–964, Aug. 2007.

[10] B. Shim and N. R. Shanbhag, “Energy-efficient soft error-tolerant digital signal processing,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 14, no. 4, pp. 1493–1497, Apr. 2006.

[11] S. Haykin, *Adaptive Filter Theory*, 4th ed. Englewood Cliffs, NJ: Prentice-Hall, 2002.

[12] 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.

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

[14] J. Chen, Y. Gu, and K. K. Parhi, “Low-complexity echo and NEXT cancellers for high-speed ethernet transceivers,” *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, no. 9, pp. 2827–2840, Oct. 2008.

[15] J. Kurzweil, *An Introduction to Digital Communications*. Hoboken, NJ: Wiley, 2000.

[16] 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. Integr. Circuits Syst. Des.*, 2002, pp. 95–100.

[17] P. Reyes, P. Reviriego, J. A. Maestro, and O. Ruano, “A new protection technique for finite impulse response (FIR) filters in the presence of soft errors,” in *Proc. IEEE Int. Symp. Ind. Electron.*, 2007, pp. 3328–3333.

[18] B. Shim, S. R. Sridhara, and N. R. Shanbhag, “Reliable low-power digital signal processing via reduced precision redundancy,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 12, no. 5, pp. 497–510, May 2004.

[19] J. W. Choi, B. Shim, A. C. Singer, and N. I. Cho, “Low-power filtering via minimum power soft error cancellation,” *IEEE Trans. Signal Process.*, vol. 55, no. 10, pp. 5084–5096, Oct. 2007.

[20] J. B. Destro-Filho and D. W. Matolak, “Effects of single-event upsets on satellite communications: Issues for blind equalizer design,” in *Proc. 6th Eur. Conf. Radiation Effects Compon Syst.*, Sep. 10–14, 2001, pp. 292–295.

[21] S. Qureshi, “Adaptive equalization,” *IEEE Commun. Mag.*, vol. COM-20, no. 2, pp. 9–16, Mar. 1982.

[22] J. M. Cioffi, W. L. Abbott, H. K. Thapar, C. M. Melas, and K. D. Fisher, “Adaptive equalization in magnetic-disk storage channels,” *IEEE Commun. Mag.*, vol. 28, no. 2, pp. 14–29, Feb. 1990.

[23] M. Tomlinson, “New automatic equaliser employing modulo arithmetic,” *Electron. Lett.*, vol. 7, no. 5, pp. 138–139, Mar. 1971.

[24] H. Harashima and H. Miyakawa, “Matched-transmission technique for channels with intersymbol interference,” *IEEE Trans. Commun.*, vol. COM-20, no. 4, pp. 774–780, Aug. 1972.

[25] Y. Gu and K. K. Parhi, “High-speed architecture design of Tomlinson–Harashima precoders,” *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 54, no. 9, pp. 1929–1937, Sep. 2007.

[26] D. Gonzalez, Single event upset simulation tool functional description ESA, Noordwijk, The Netherlands, ESA Report TEC-EDM/DCC-SST2, Jul. 2004.

[27] 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. Ind. Electron.*, 2007, pp. 3316–3321.

Pedro Reviriego (A'03–M'04) received the M.Sc. and Ph.D. degrees (with honors) in telecommunications engineering from the Technical University of Madrid, Madrid, Spain, in 1994 and 1997, respectively.

From 1997 to 2000, he was an R&D Engineer with Teldat, Madrid, working on router implementation. In 2000, he joined Massana to work on the development of 1000BaseT transceivers. During 2003, he was a Visiting Professor with the University Carlos III, Leganés, Madrid. From 2004 to 2007, he was a Distinguished Member of Technical Staff with the LSI Corporation, working on the development of Ethernet transceivers. He is currently with the Universidad Antonio de Nebrija, Madrid. He is the author of numerous papers in international conference proceedings and journals. He has also participated in the IEEE 802.3 standardization for 10 GBaseT. His research interests include fault-tolerant systems, performance evaluation of communication networks, and the design of physical layer communication devices.

Juan Antonio Maestro (M'07) received the M.Sc. degree in physics and the Ph.D. degree in computer science from Universidad Complutense de Madrid, Madrid, Spain, in 1994 and 1999, respectively.

He has served both as a Lecturer and a Researcher at several universities, such as the Universidad Complutense de Madrid; the Universidad Nacional de Educación a Distancia (Open University), Madrid; Saint Louis University, Madrid; and the Universidad Antonio de Nebrija, Madrid, where he currently manages the Computer Architecture and Technology Group. His current activities are oriented to the space field, with several projects on reliability and radiation protection, as well as collaborations with the European Space Agency. Aside from this, he has worked for several multinational companies, managing projects as a Project Management Professional and organizing support departments. He is the author of numerous technical publications, both in journals and international conferences. His areas of interest include high-level synthesis and cosynthesis, signal processing, and real-time systems, fault tolerance, and reliability.

Shih-Fu Liu received the B.Sc. degree in electronic engineering from Carinthian Tech Institute, Villach, Austria, in 2003 and the M.Sc. degree in computer system engineering from the Technical University of Denmark, Lyngby, Denmark, in 2005. He is currently working toward the Ph.D. degree with the Universidad Antonio de Nebrija, Madrid, Spain.

He has worked in cooperation with multinational companies and various universities all over Europe in the European project SoC-Mobinet (IST-2000-30094), developing a design and FPGA-implementation of a runtime-reprogrammable high-performance multirate filter processor. He is currently a Full-Time Researcher with the Universidad Antonio de Nebrija.