

Increasing Reliability of FPGA-Based Adaptive Equalizers in the Presence of Single Event Upsets

Shih-Fu Liu, Gabriele Sorrenti, *Member, IEEE*, Pedro Reviriego, *Member, IEEE*, Fabio Casini, *Member, IEEE*, Juan Antonio Maestro, *Member, IEEE*, and Monica Alderighi, *Member, IEEE*

Abstract—Reliability is a major concern for electronic circuits, especially for those that operate in harsh environments. One source of problems are Single Event Upsets (SEU), which change the value of flip flops and memory cells. SEUs are a major issue for SRAM-based Field Programmable Gate Arrays (FPGAs), as they may alter the circuit functionality, creating errors that will only be removed if the device is reprogrammed. The cost of traditional techniques to deal with SEUs, like triplication, can be excessive in some applications. One example are Space systems, in which power consumption and weight are limited. In those cases, the use of ad hoc protection techniques that can reduce the cost is interesting. In this paper, new protection techniques for adaptive equalizers implemented in SRAM-based FPGAs are presented. The proposed techniques use the knowledge of the equalizer to provide effective protection at a lower cost. The results show a reduction of up to 70% in the use of resources in comparison to the commercial XTMR solution.

Index Terms—Adaptive filters, design cross-section, fault injection, field programmable gate array (FPGA), single event effects.

I. INTRODUCTION

ADVANCES in the semiconductor industry leading to high density chips by technology scaling, reduced voltage operation and higher frequencies make integrated circuits more susceptible to soft errors caused by energetic particles, reducing their reliability [1]. Although soft errors do not cause physical damage on the chip, memory cell content may be altered, leading to incorrect processing of the stored value [2].

SRAM-based Field Programmable Gate Arrays (FPGAs) have the capability to implement any kind of digital circuit by loading a bitstream onto its configuration memory. The configuration memory, which totally controls the implemented circuit, is composed of static RAM cells. Therefore, SRAM-based FPGAs have high sensitivity to soft errors, especially single event upsets (SEUs), which may change the behavior of the originally implemented circuit [3]. This high sensitivity to

SEUs limits the usage of SRAM-based FPGAs in safety or mission-critical applications, unless suitable hardening techniques are adopted.

Several studies by radiation [4] or by emulation of the effects of SEUs in the FPGA's configuration memory as bit-flips in the memory content [5]–[8] have been carried out to investigate the sensitivity of SRAM-based FPGAs to ionizing radiation. Of all above mentioned platforms emulating the effects of SEUs, the ones proposed in [5]–[7] are consecutive enhancements of one of the others doing mostly static analysis of the loaded design. In [8], the authors presented a platform called FLIPPER, which performs fault injection tests by applying workload to the loaded Design Under Test (DUT). Furthermore, FLIPPER has been validated by radiation tests showing that the cross section of the design implemented in the FPGA is quite well emulated by its fault-injection tests [9].

In the recent years, the number of FPGAs used in space applications is increasing, starting with the Mars Lander [10], Mars Rover [10] and Venus Express [11]. This trend will continue growing in a near future due to their re-programmability [12]. As communications are fundamental to spaceborne applications, such as satellites, spacecrafts, unmanned missions, digital filters play an important role in space systems [13]. Adaptive filters are commonly used solutions for equalization and noise or interference cancellation. One example is a Feed Forward Equalizer (FFE), in which the effects of SEUs were studied in [14]. More generally due to their adaptive nature adaptive filters are well suited for circuit specific protection techniques [15] and [16]. The main goal of an equalizer is to compensate the effects introduced by the channel in the transmitted signal. One of those effects is the Inter-Symbol 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 [13], causing bit errors.

To guarantee fault tolerance of applications implemented in FPGAs, there is the need of system-level protection techniques. One option for system-level protection is Duplication with Comparison (DwC), which is capable of detecting the fault event but cannot identify which of the duplicated components is the faulty one [17]. Another one is called Recomputing with Shifted Operands (RESO) that calculates twice the result of the functional unit, posing an intrinsic delay of more than double of the execution time [18]. The preferred choice to improve the reliability is Triple Modular Redundancy (TMR) or Xilinx's TMR (XTMR) for Virtex FPGAs, since it does not require any architectural innovation and it is function-independent. The drawback is that the area and the power consumption are triplicated too [19], [20]. The described protection techniques

Manuscript received September 14, 2010; revised November 18, 2010; accepted December 23, 2010. Date of publication February 22, 2011; date of current version June 15, 2011. This work was supported in part by the Spanish Ministry of Science and Innovation under Grant AYA2009-13300-C03-01.

S.-F. Liu, P. Reviriego, and J. A. Maestro are with the Universidad Antonio de Nebrija, Madrid, Spain (e-mail: sliu@nebrja.es, previrie@nebrja.es, jmaestro@nebrja.es).

G. Sorrenti and F. Casini are with the Sanitas EG s.r.l., Milano, Italy (e-mail: gabriele.sorrenti@sanitaseg.it, fabio.casini@sanitaseg.it).

M. Alderighi is with the Istituto Nazionale di Astrofisica, Milano, Italy (e-mail: monica@iasf-milano.inaf.it).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TNS.2011.2106799

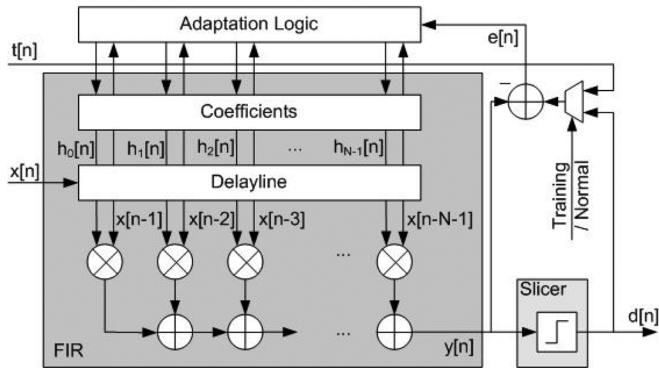


Fig. 1. Block diagram of an adaptive FFE.

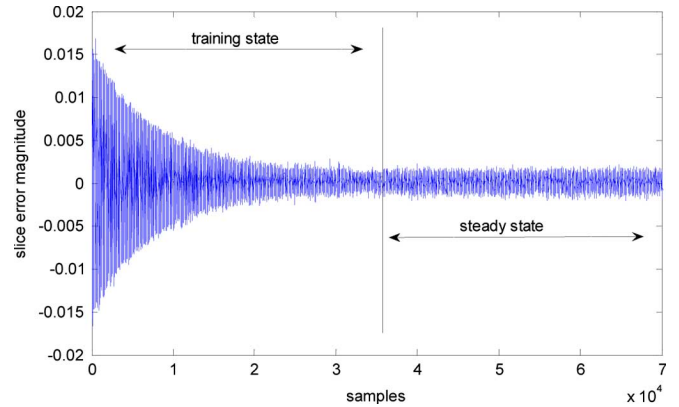


Fig. 2. Slicer error during training state and then in normal state.

are commonly complemented by a periodical scrubbing of the FPGA configuration memory that ensures that errors do not accumulate over time [21]. In general, when scrubbing is used, the scrubbing rate is tuned with respect to the expected upset rate of the application. It can be, thus, reasonable to assume that at any given time the design will be affected by only one error as previous errors have been removed by the scrubbing process.

This paper presents novel system-level protection techniques based on DwC for an adaptive FFE implemented in a Xilinx FPGA. It is assumed that scrubbing is used such that the design is affected by one error at a time on average. The details on how scrubbing is implemented are not discussed in this paper, the reader can go to [21] for more information on the different options to implement scrubbing. The FLIPPER platform was used to evaluate the design robustness in the presence of SEUs. For that a single error was inserted in each simulation run. The results show that the proposed techniques are capable of protecting against most errors. The proposed techniques are also compared with XTMR in terms of usage of FPGA resources. The proposed techniques achieve a large reduction in the use of all types of FPGA resources reaching a value of up to 70% for some of them.

The paper is organized as follows. Section II gives an overview of the adaptive FFE, while Section III presents the XTMR and the proposed protection techniques. The test setup and procedure are illustrated in Section IV, while Section V presents the result analysis. Finally, conclusions are drawn in Section VI.

II. PRELIMINARIES

This section describes the adaptive Feed Forward Equalizer focusing on the slicer error used for adaption. This error will play an important role in identifying fault occurrences.

The Feed Forward Equalizer (FFE), illustrated in Fig. 1, is composed of three main blocks, a traditional Finite Impulse Response (FIR) filter with coefficients $h[i]$, input signal $x[n]$ and output signal $y[n]$, an adaptation logic that updates the values of the coefficients $h[i]$ periodically according to the error $e[n]$ and a slicer unit estimating the received symbol $d[n]$ from the filter output $y[n]$. The error $e[n]$ is calculated as a difference of $t[n]$ and $y[n]$ during training and the difference between $d[n]$ and $y[n]$ in normal operation.

The inputs to the adaptation logic are the input signal to the filter and an error measure $e[n]$ which are used to compute the new values of the coefficients. The idea behind the adaptation is to gradually adapt the coefficient values to reduce the observed error $e[n]$, as seen in Fig. 2. At start-up, while in the training phase, the error is calculated by the difference between the output of the filter $y[n]$ and the training signal $t[n]$ yielding the error $e[n]$.

Once the coefficients are adapted to the channel the adaption speed decreases and the error calculation is done by generating the difference between the filter output $y[n]$ and the slicer output $d[n]$. In this phase, called steady state, the slicer error should only fluctuate minimally in function of changes in the channel. Therefore, changes in the slicer error can be representative of an alteration in the overall system.

In this section, the slicer error and its function in the FFE have been explained, highlighting that the slicer error is a good indicator for anomalies like fault occurrences. This fact is used for the proposed techniques in the next section.

III. SYSTEM LEVEL PROTECTION TECHNIQUES

This section presents system-level protection techniques based on redundancy. The simplest and the most effective form of fault tolerance by redundancy is TMR. As it is very costly there is the need for other techniques. Therefore, this section also presents novel techniques which are based on the properties of the FFE for fault detection, localization and mitigation.

A. Xilinx Triple Modular Redundancy (XTMR)

The simplest technique is Triple Modular Redundancy (TMR). The FPGA vendor Xilinx provides a tool called Xilinx's TMR (XTMR) which applies TMR on a design for a specific Xilinx FPGA. This XTMR tool has become an industrial standard in the past few years for designs targeting Xilinx FPGAs and it is commonly used as a [19] and [20].

Concerning the XTMR implementation of the FFE, the following two options have been applied [22]. For "Controlling Component Triplication", the option of "XTMR Types Standard" has been selected as it triplicates for a hierarchical design, all the inputs and all underlying primitives. All the inputs are assigned to unique input pins. The option "Triple-Voted"

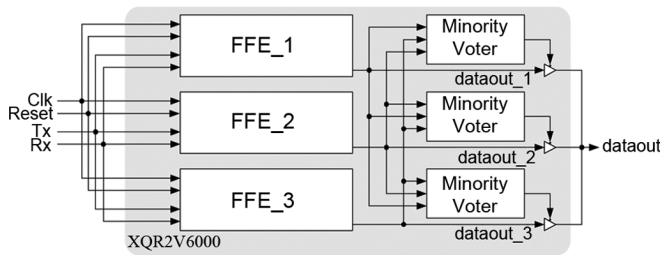


Fig. 3. Schematic of the XTMR of the FFE.

for “Controlling Output Triplication” has been applied to the dataout signal, as shown in Fig. 3.

In this scheme, the minority voter inhibits, by acting on a tristate buffer, the propagation to the output of the signal that differs from the other two replicas.

As this fault tolerance technique is based on TMR, the resource consumption and power consumption is about three times of the unprotected version.

B. Proposed Techniques

The proposed Enhanced Dual Modular Redundancy (EDMR) methods are based on DwC but instead of the comparator they use enhanced voting logic that can detect failures, distinguish the faulty module and avoid failure propagation, by exploiting the observations on the slicer error discussed in the previous section.

In the normal steady state, the slicer error rarely surpasses the threshold, which is chosen to be twice the error magnitude in the steady state, as shown in Fig. 4(a). Occasionally, the threshold is exceeded for a few cycles due to random bit errors caused by noise or other imperfections in the received signal. When an SEU hits the configuration memory of the FPGA, it might change the original design architecture or functionality, affecting the slicer error or even altering the dataout:

- 1) A minor change in the FFE circuitry can cause an increase in the slicer error, as shown in Fig. 4(b), but might not be sufficient to alter the dataout. For avoiding a bigger impact by accumulation of SEUs, the circuitry should be re-configured, but only in the cases when the slicer error exceeds the threshold for a certain number of cycles. In this way, those can be distinguished from a common bit error.
- 2) A major change in the FFE circuitry can cause an error that alters the output signal from its occurrence. Hence, the slicer error increases without any possibility of recovering the correct output values, as illustrated in Fig. 4(c).

These two cases have different symptoms and therefore, also need different approaches of detection and mitigation. With this understanding of the circuitry and with the help of additional logic, the faulty module can be localized and the propagation of the fault can be mitigated.

1) *Enhanced Dual Modular Redundancy (EDMR)*: As mentioned before, the proposed architecture consists of two FFE modules placed in parallel with a voting logic, resembling DMR. All input signals are duplicated outside of the FPGA and fed inside on different pins in order to avoid a single point of failure in case an SEU changes a configuration bit controlling the forwarding of the signals. Both modules forward their data

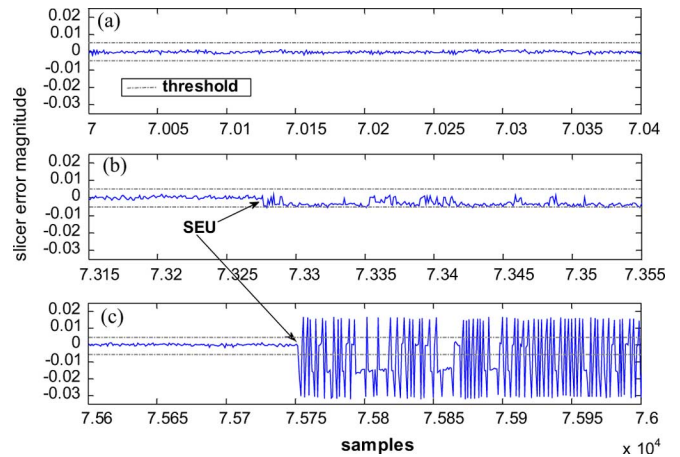


Fig. 4. Slicer error obtained by simulation when: (a) slicer error is in normal state, (b) slicer error after a minor change, (c) slicer error after a major change.

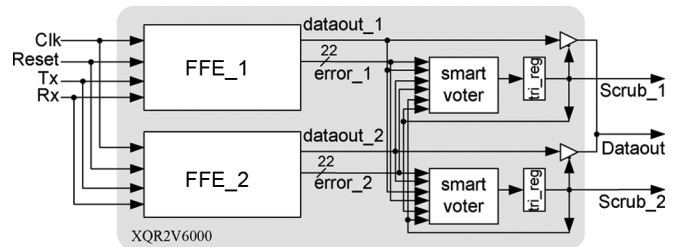


Fig. 5. Design with two parallel sub-FFEs and a voter.

outputs and their slicer error to the voting logic modules and the main dataout is protected by the tristate buffer mechanism as the XTMR version, as shown in Fig. 5.

Failures are detected by observing the slicer error as mentioned previously and can have the following effects:

- mismatching of dataouts;
- one of the slicer errors exceeding the threshold.

At all times, the smart voters constantly observe the error signals and whenever one exceeds the threshold for a given number of cycles a fail signal called *scrub* is activated, which is stored in the *tri_reg* register. The value of the *tri_reg* is fed back to both smart voters so that the faulty module is not considered anymore and the smart voters forward the correct dataout directly to the main output of the device. When the output of the two replicas differs, each smart voter compares the error values coming from each replica. The module with the greater error value is identified as the corrupted replica and the corresponding scrub signal is activated.

These *scrub* signals could be used to trigger partial reconfiguration by an external scrubbing logic to recover the faulty module and resetting the *tri_reg*, while the system continues to operate with the other FFE module. As discussed in the introduction the implementation details of scrubbing are not covered in this paper.

2) *Enhanced DMR With Reduced Replica (EDMRwRR)*: This version is based on the same ideas as the proposed EDMR version. However, instead of using the same replica for the duplication, it uses a reduced replica which works with less precision by using fewer bits in the coefficients and delay-line.

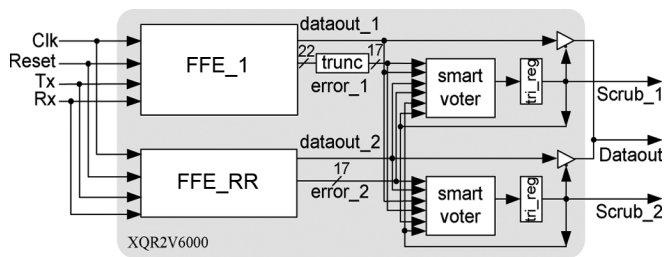


Fig. 6. Schematic of the EDMR with reduced replica.

This replica provides worse equalization performance but a significant area reduction.

The *error_1* signal is truncated from 22 bits to 17 bits of the *error_2* signal to enable the comparison for the smart voters. The rest of the structure is completely identical to the EDMR version as illustrated in Fig. 6.

By using a reduced replica the overall area consumption is reduced and therefore also the probability of SEUs hitting the configuration bits used by the circuitry.

IV. TEST SETUP AND FAULT MODEL

This section presents the fault injection setup and the fault model used for retrieving the results discussed in the next section. The proposed techniques have been evaluated by fault injection simulations. This has been accomplished by comparing SEU sensitivity of the following design versions: plain, XTMR, EDMR and EDMRwRR.

The fault injection simulations have been performed by using the FLIPPER platform [8]. The fault model adopted in FLIPPER is the bit-flip of configuration memory cells [9]. For each of the four design variants, an injection campaign has been accomplished aimed at analyzing the design sensitivity to a single bit-flip occurring in random locations of the device configuration memory based on the assumption that between the occurrences of the SEUs, there is sufficient time for reconfiguration of the faulty module. These campaigns consist of 300,000 fault injections runs each.

The details of the test procedure are illustrated in Fig. 7. Initially the design is loaded into the FPGA and a configuration bit is corrupted. Then the design is exercised and the outputs are compared with the ones obtained in the uncorrupted design.

The outputs logged and compared are the slicer errors from each module, the received symbol (Dataout) and the control signals Scrub_1 and Scrub_2. If there are no differences, the error has had no apparent effect on the circuit functionality. If there are differences, the circuit has been affected but an error will occur only if there are differences in the received symbols. In the following, tests in which there are differences in outputs will be denoted as *incidents* and only some of them (ones with differences in the received symbols) will be classified as failures.

This method of fault injection resembles what occurs during radiation testing as failures occur randomly over the design. One difference is that in our case errors are inserted and then the circuit is put in operation while during radiation exposure the circuit is operating when the errors occur. The equalizer performs

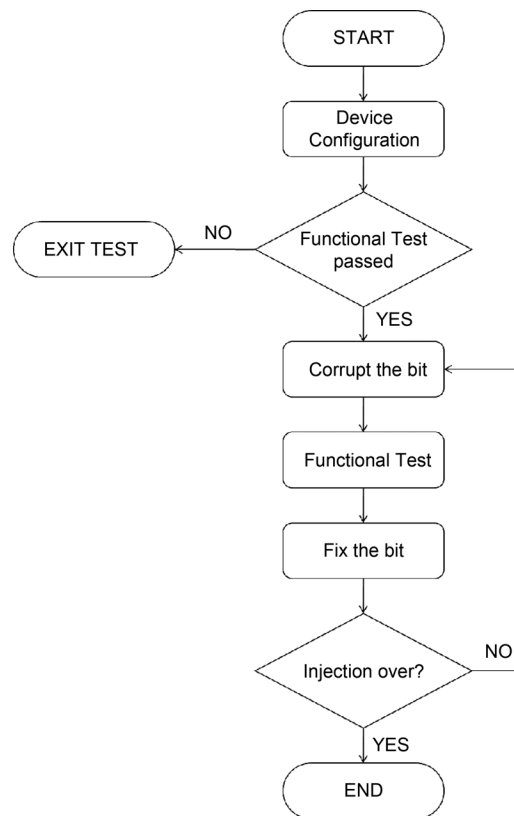


Fig. 7. Diagram of the test procedure used in the fault injection simulations.

an initial adaptation and then enters the steady state in which it normally remains during the rest of the circuit operation. Therefore, in our case errors may affect the initial adaptation while for radiation exposure the errors will occur during the steady state. To eliminate this difference the coefficients in the equalizer filter are loaded when the circuit is initialized to the values that they take in the steady state so that from the first cycle the equalizer operates in the steady state. This means that from the beginning of the simulation the equalizer behaves as it would when an error occurs during radiation exposure.

During steady state operation, the equalizer filters an incoming signal composed of pseudo-random symbols because most communication systems use scramblers to avoid correlation between transmitted symbols. The filtering operation is a datapath operation with finite memory by nature, this combined with the random nature the incoming signal makes it reasonable to assume that the effect of an error on the design does not depend on the specific time at which the error occurs. In [23] the same fault injection tool (FLIPPER) was used on another datapath circuit (an FFT) and the results of the simulation and radiation testing were in good agreement. Those results are in line with our reasoning.

V. SIMULATION RESULTS AND ANALYSIS

An FPGA can be seen as a pool of resources that are used to implement a given design. In the case of the Xilinx Virtex II FPGAs these resources include for example Input Output Blocks (IOBs) used to connect the inputs and outputs of the design to the pins of the device. The number of IOBs

TABLE I
RESOURCE USAGE OF THE IMPLEMENTED CIRCUITS USING DEDICATED MULTIPLIERS

Designs	IOBs		Slices		LUTs		FFs		MULTs18x18	
	[#]	rel.	[#]	rel.	[#]	rel.	[#]	rel.	[#]	rel.
Plain	33 (4%)	1.0	715 (2%)	1.0	1,177 (1%)	1.0	637 (1%)	1.0	48 (33%)	1.0
XTMR	105 (12%)	3.2	3,767 (11%)	5.3	4,947 (7%)	4.2	1,911 (2%)	3.0	144 (100%)	3.0
EDMR	70 (8%)	2.1	1,548 (4%)	2.2	2,575 (3%)	2.2	1,322 (1%)	2.1	96 (66%)	2.0
EDMRwRR	63 (7%)	1.9	1,109 (3%)	1.6	1,813 (2%)	1.5	1,059 (1%)	1.7	72 (50%)	1.5

used depends directly on the number of inputs and outputs of the design. Internally the FPGA is divided into Slices that group resources such as Look-Up Tables (LUTs) used to implement logic functions and Flip-Flops (FFs) used to implement sequential elements. Those resources inside a Slice can be used to implement part of the design functionality and by using various slices the whole design is implemented. In some cases only parts of the elements in a slice are used by the design, this means that the rest of the elements are available to implement other modules. For example in our case the equalizer will be part of a receiver that includes other modules. If in a given slice the equalizer uses only part of the resources, the rest can be potentially used by other modules of the receiver. Therefore, in addition to evaluating the number of slices used by the equalizer it is interesting to look at the use of the resources within the slices (LUTs and FFs). Finally, the FPGA also has some resources that are targeted for arithmetic computations such as multipliers (MULTs). The equalizer implements a digital filter that computes a sum of products. Therefore, it will use those multipliers and it is also interesting to look at the number of them used in each version of the design.

To compare the resources used by the different designs, all have been synthesized for the Xilinx XQR2V6000 FPGA [24] and their resource usage has been documented in Table I.

The table also includes the relative values comparing the system-level protection techniques to the plain version next to the absolute values of the resource usage with the percentage of overall chip usage.

As expected, Table I shows that the EDMR version is a little bit more than two times bigger than the plain version. The EDMRwRR is achieving a reduction and situates itself around 1.6 times of the plain version. The expected overhead for XTMR of three times is only kept for the *IOBs*, *FFs*, and *MULTs*18x18 but when looking to the *LUTs*, the overhead increases to 4.2 and 5.3 for slices. Among the different resource usage figures, probably the single one that gives a better estimate of the resource usage is the number of slices. Focusing on the slice usage and taking XTMR as the reference for comparison, the proposed techniques achieve for the EDMR and the EDMRwRR a reduction of 59% and 70%, respectively. This shows how the proposed techniques can achieve significant cost reductions.

Fault injection simulation was performed using the setup and test procedure described before. The fault injection results of 300,000 configuration memory bit-flips for each of the four designs are summarized in Table II, as discussed in the previous section the *incidents* are the cases when there is a difference between one of the slicer errors and the golden error vector and

TABLE II
FAULT INJECTION RESULTS WITH 300,000 INJECTIONS FOR EACH DESIGN

Design	total # incidents	total fail	failure rate	rel. %
Plain	890	282	0.094%	n/a
XTMR	1669	0	0.000%	0.0%
EDMR	1908	13	0.004%	4.3%
EDMRwRR	1291	8	0.003%	3.2%

a failure occurs when in addition the dataout differs from the golden.

The plain version has a failure rate of 0.094% where 282 out of 890 *incidents* caused a failure on the dataout. For the EDMR and the EDMRwRR, the total number of *incidents* increases with the factor discussed for the resources. For the XTMR version, the total number of *incidents* did not increase by the factor of the resource usage. This is due to the insertion of voters in each feedback path. The failure rate for the XTMR is 0.000% which is the expected value based on the test setup. For the EDMR and EDMRwRR versions, bit-flips affecting the smart voter logic are thought to be causing failures on the main dataout, resulting in failure rates of 0.004% and 0.003%, respectively. The EDMRwRR achieves a higher failure rate reduction due to the fact that the reduced replica used for the EDMR uses less complexity. This results into a smaller total area and also a smaller number of total *incidents* in comparison to the EDMR version as Table II shows. The last column shows the remaining failure rate when applying one of these system-level protection techniques to the plain version resulting 0.00% for XTMR and 4.26% and 3.19% for EDMR and EDMRwRR, respectively.

In summary the proposed techniques can provide a substantial cost reduction at the expense of a small increase in the failure rate.

VI. CONCLUSIONS

This paper proposes novel system-level protection techniques for adaptive feed forward equalizers, which exploit application and system knowledge, resulting in a more intelligent protection with up to a 70% saving of circuit resources in comparison to commercial XTMR solution, as it has been shown in the paper.

As future work, the smart voters of the proposed techniques will be protected by XTMR, so that the failure rate could possibly be reduced to 0.000%, equivalent to the protection level of XTMR version. Furthermore, the fault model will be changed and all four designs tested for fault accumulation to estimate the mean time to failure for each design. Finally the long term objective will be to perform radiation tests to validate the results obtained with the fault injection platform.

REFERENCES

- [1] International Technology Roadmap for Semiconductors: 2009 Edition, Chapter Design, 2009. Mar. 15, 2010 [Online]. Available: http://www.itrs.net/Links/2009ITRS/2009Chapters_2009Tables/2009_Design.pdf
- [2] R. C. Baumann, "Radiation-induced soft errors in advanced semiconductor technologies," *IEEE Trans. Device Mater. Rel.*, vol. 5, no. 3, pp. 301–316, Sep. 2005.
- [3] M. Bellato *et al.*, "Evaluating the effects of SEUs affecting the configuration memory of an SRAM-based FPGA," in *Proc. Design, Automation and Test in Europe (DATE)*, 2004, vol. 1, pp. 584–589.
- [4] C. Yui *et al.*, "SEU mitigation testing of Xilinx Virtex II FPGAs," in *2003 Radiation Effects Data Workshop Record*, 2003, pp. 92–97.
- [5] L. Antoni, R. Leveugle, and B. Fehér, "Using run-time reconfiguration for fault injection applications," *IEEE Trans. Instrum. Meas.*, vol. 52, no. 5, pp. 1468–1473, Oct. 2003.
- [6] J. Tombs and M. A. Aguirre, "FT-UNSHADES," presented at the Eur. Space Agency Microelectronics Day, Feb. 4, 2004.
- [7] L. Sterpone and M. Violante, "A new partial reconfiguration-based fault-injection system to evaluate SEU effects in SRAM-based FPGAs," *IEEE Trans. Nucl. Sci.*, vol. 54, no. 4, pp. 965–970, Aug. 2007.
- [8] M. Alderighi *et al.*, "Evaluation of single event upset mitigation schemes for SRAM based FPGAs using the FLIPPER fault injection platform," in *Proc. 22nd IEEE Int. Symp. Defect and Fault Tolerance in VLSI Systems*, 2007, pp. 105–113.
- [9] M. Alderighi *et al.*, "Using FLIPPER to predict proton irradiation results for VIRTEX 2 devices: A case study," *IEEE Trans. Nucl. Sci.*, vol. 56, no. 4, pt. 2, pp. 2103–2110, Aug. 2009.
- [10] Xilinx Solutions for Aerospace & Defense Applications, Mar. 15, 2010 [Online]. Available: http://www.xilinx.com/esp/mil_aero/collateral/presentations/xsol_aero_def_appl.pdf
- [11] W. J. Markiewicz *et al.*, "VENUS monitoring camera for VENUS EXPRESS," presented at the Eur. Geosciences Union. 1st General Assembly, Nice, France, Apr. 2004.
- [12] B. Fiethe *et al.*, "Reconfigurable system-on-chip data processing units for space imaging instruments," in *Proc. Design, Automation and Test in Europe (DATE)*, 2007, vol. 1, pp. 1–6.
- [13] S. Qureshi, "Adaptive equalization," *IEEE Commun. Mag.*, vol. 20, no. 2, pp. 9–16, Mar. 1982.
- [14] J. B. Destro-Filho *et al.*, "Effects of single-event upsets on satellite communications: Issues for blind equalizer design," in *Proc. Radiation and Its Effects on Components and Systems Conf.*, 2001, pp. 292–295.
- [15] 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, pt. 3, pp. 1700–1707, Jun. 2008.
- [16] P. Reviriego, J. A. Maestro, and S. Liu, "Efficient soft error-tolerant adaptive equalizers," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 8, pp. 2032–2040, Aug. 2010.
- [17] D. Siewiorek and R. Swarz, *The Theory and Practice of Reliable System Design*. Bedford, MA: Digital Press, 1982.
- [18] J. H. Patel and L. Y. Fung, "Concurrent error detection in ALU's by recomputing with shifted operands," *IEEE Trans. Comput.*, vol. C-31, no. 7, pp. 589–595, Jul. 1982.
- [19] F. K. Kastensmidt *et al.*, "On the optimal design of triple modular redundancy logic for SRAM-based FPGAs," in *Proc. Design, Automation and Test in Europe (DATE)*, IEEE, 2005, pp. 1290–1295.
- [20] C. Carmichael, "Triple module redundancy design techniques for Virtex FPGAs," *Xilinx Application Notes 197*, 2006.
- [21] M. Berg *et al.*, "Effectiveness of internal versus external SEU scrubbing mitigation strategies in a Xilinx FPGA: Design, test and analysis," *IEEE Trans. Nucl. Sci.*, vol. 55, no. 4, pt. 1, pp. 2259–2266, Aug. 2008.
- [22] "TMRTool User Guide, Xilinx User Guide UG156," 2004.
- [23] M. Alderighi *et al.*, "Experimental validation of fault injection analyses by the FLIPPER tool," *IEEE Trans. Nucl. Sci.*, vol. 57, no. 4, pt. 1, pp. 2129–2134, Aug. 2010.
- [24] QPro Virtex-II 1.5 V Radiation Hardened QML Platform FPGAs, DS124 (v1.1), Xilinx, 2004.