Evaluating the Effectiveness of a Mixed-Signal TMR Scheme Based on Design Diversity

Gabriel de M. Borges, Luiz F. Gonçalves, Tiago R. Balen, Marcelo S. Lubaszewski
Universidade Federal do Rio Grande do Sul
Programa de Pós-Graduação em Engenharia Elétrica
Avenida Osvaldo Aranha, 103, Centro, Porto Alegre, RS
CEP: 90035-190, Telephone: +55 51 3308-3129
{gabriel.borges, tiago.balen}@ufrgs.br, {luizfg, luba}@ece.ufrgs.br

ABSTRACT
This paper explores the concept of design diversity redundancy applied to mixed-signal (MS) circuit blocks, as a proposal to increase system reliability. Three different implementations of a second order low-pass filter (which perform the same transfer function) associated to a majority voter are used to build a diversity TMR scheme. The whole system is prototyped by using a single programmable mixed-signal device. Functional verifications and a fault injection campaign are performed, and the effectiveness of the voting system to detect single faults is investigated. Based on these results, analytical calculations were done to estimate the behavior of the system under the occurrence of double faults. Results show a very good ability of the system to tolerate double faults from the considered model.

Categories and Subject Descriptors
B.8.1 [**Reliability, Testing and Fault-Tolerance**]

General Terms
Redundant Design and Reliability.

Keywords
Fault Tolerance, Mixed-Signal; Design Diversity TMR.

1. INTRODUCTION
High reliability applications, such as avionic systems, medical instruments or power plants control, for example, require that the probability of failure be reduced to the minimum possible. On the other hand, as the demand for smaller and low-power devices increases, state-of-art circuits are becoming more sensitive to the radiation effects (space applications), crosstalk, electromagnetic interference and other environmental interactions, besides the natural system aging.

Electronic systems employed in applications susceptible to radiation exposure may suffer long term effects known as TID (Total Ionizing Dose) and Single Event Effects (SEE), besides displacement damage and destructive effects [1]. A special class of SEE is the Single Event Upset (SEU) [2], which is an important matter of concern in systems with DRAM or SRAM memory blocks. In SRAM-based FPGAs, an error may occur, due to an SEU, when an energetic particle hits a memory cell of the device, causing a bit-flip [3]. The effects of SEU in programmable analog technologies were first investigated in [4], showing that the inversion of a single programming bit may be catastrophic to the system functionality.

SBCCI'10, September 6–9, 2010, Sao Paulo, Brazil.

Usually, the techniques employed to add fault tolerance to electronic systems are based on hardware and/or time redundancy [5-7]. Among these techniques the most popular is the Triple Modular Redundancy (TMR), which consists in triplicating the designed circuit, and delivering the system copies outputs to a majority voter. If there is an error in one of the blocks, two of them continue operating properly and the correct value is chosen by the voter [7-9].

Traditional TMR techniques consider the replication of exact copies of the circuit to be protected [7-9]. However, if the copies are identical, environmental interactions, aging or radiation incidence may affect different circuit parts at same time and in similar ways, causing multiple faults. This way, similar errors may occur in more than one system copy output, and the voter may switch faulty signals to system output.

Conversely, if the circuit copies in TMR systems are built with different architectures, the probability of multiple faults affecting different TMR blocks can be reduced, since each system copy may have different levels of resilience associated to the diverse fault generation mechanisms and sources.

The present work addresses the application of the concept of fault tolerance by design diversity [10] to programmable MS technologies. The proposed scheme consists in a TMR system in which the design of interest is a low-pass filter. The validation of the scheme is made through the prototyping of the system in a mixed-signal programmable device, the PSoC (Programmable System-on-Chip) CY8C29466, from Cypress Semiconductor [11].

The rest of the paper is organized as follows: In Section 2 the diversity TMR technique is introduced. Section 3 presents the system architecture considered in this work. Section 4 describes the fault injection experiments, while the results are discussed in section 5. Section 6 concludes the paper.

2. DIVERSITY TMR: THE CONCEPT
TMR is a well-known technique frequently applied to modern digital systems which require a high degree of reliability [5-9].

If multiple faults occur sequentially, with a relatively long inter-occurrence interval, the TMR system can tolerate them. However, multiple faults occurring in more than one copy at the same time may cause the voter to take the wrong decision, if two or more TMR blocks present bit inversion in their outputs [12-14]. Faults generated by environmental interactions, aging or radiation incidence may affect different circuit parts at same time, and are potential causes of multiple faults.

The use of different technologies or architectures to implement a TMR scheme is an alternative way to increase the reliability of a system. This technique is referred to as "TMR based on design diversity" [15] and from here on it will be denoted by DTMR.

In the DTMR the system is replicated three times and their outputs are delivered to a voter. However, to achieve the design diversity, the three circuit copies are implemented with different technologies, algorithms or architectures. In this work the DTMR is applied considering also different domains (analog and digital), and system levels (software and hardware), to build the system copies, as depicted in Figure 1.

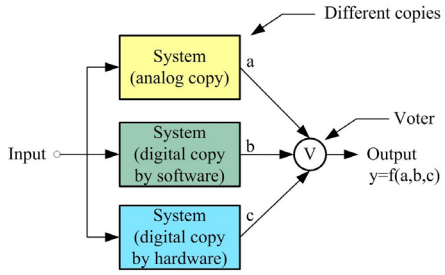

Figure 1. Mixed-Signal DTMR architecture.

Despite TMR be simple and easy to implement, it implies in large area overhead and increase in the power consumption [9, 12, 13, 16]. Obviously, this new DTMR technique also leads to a significant area overhead. However, if reliability is the major concern, this aspect becomes less important due to the increasing of integration capacity of current microelectronics technologies. Additionally, recent analog and mixed-signal programmable technologies, offer a large number of programmable resources, which can be used to increase the system reliability without increasing the silicon area or implementation costs.

3. MIXED-SIGNAL CASE STUDY

In this work the design of interest is a second order low-pass filter. The three TMR copies of the filter perform the same function, but are implemented in three different architectures and domains, as follows: 1) an analog filter, 2) a digital filter implemented by software and 3) a digital filter implemented by hardware.

Furthermore, a voting system is also implemented by software. The considered analog filter is a Butterworth implementation with cutoff frequency of 450Hz. Both digital filters are designed to present the same frequency response of the analog one.

The whole system is prototyped in the PSoC CY8C29466, from Cypress Semiconductor [11], a mixed-signal programmable SoC. This component consists in a full mixed-signal platform with a microcontroller, memory blocks, data converters and additional system resources, such as multiply accumulate units and digital clock dividers, besides a set of analog and digital user programmable blocks.

3.1 Analog Low-Pass Filter

The analog filter was implemented by using the PSoC programming library, in which a pre-built low-pass filter is available. This implementation uses the switched-capacitor programmable blocks of the device.

The corner frequency and quality factor are functions of the capacitances and switching frequency of the programmable capacitors. A proprietary tool, called *PSoC Designer* [11], is used to program the component. In addition to specify the parameters of filters, one must allocate it properly in the programmable analog/MS programmable array.

Figure 2 shows the programmed functions and their placement in the programmable array. As it can be seen in Figure 2, the analog filter is allocated in the left blocks of the analog/MS array.

3.2 Digital Low-Pass Filter by Software

The process to implement a digital filter includes: analog-to-digital conversion, digital signal processing and digital-to-analog conversion. The digital signal processing is implemented in software, i.e., the calculations are made by using the Arithmetic Logic Unit (ALU) from the PSoC microprocessor.

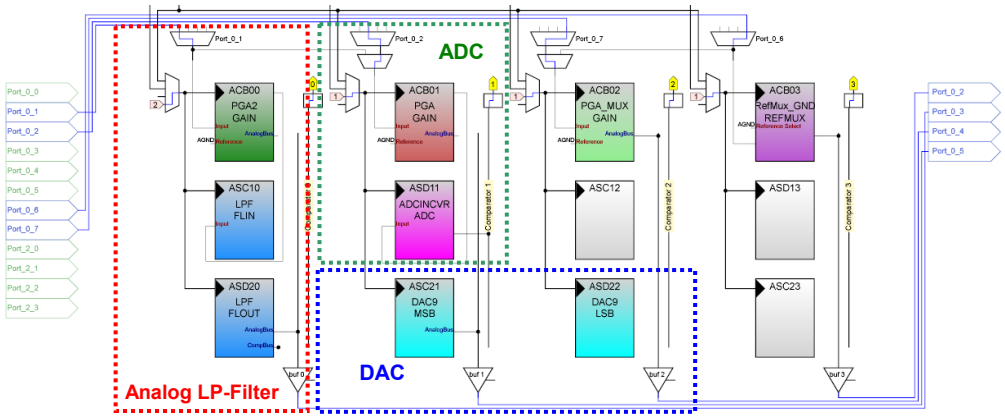

Figure 2. Allocation of selected user modules in the PSoC array.

The A/D converter is a 7 to 13-Bit *Variable Resolution Incremental ADC*. According to the PSoC datasheet [11], with a 7-bit resolution for the ADC, with *DataClock* = 3MHz and being *CalcTime* = 50 cycles, the sample rate can be calculated as follows:

$$SampleRate = \frac{DataClock}{2^{Bits+2} + CalcTime} = 5338\,Hz \quad (1)$$

The digital filter implemented is a second order FIR filter with integer coefficients, and its transfer function in the discrete-time domain is given by:

$$y = 4x[0] + 7x[1] + 4x[2] \quad (2)$$

where: *x[0]*, *x[1]* and *x[2]* are the signal samples, with the respective delays, and *y* is the filter's digital output.

Figure 3 shows part of the program code which implements the digital signal processing. After this computation process, the samples are sent to a 9-bit D/A converter. The DAC is allocated in two mixed-signal programmable blocks of the PSoC (Figure 2).

```
int a[3]={4,7,4};
…
for (count=3; count>0; count--)
  x[count]=x[count-1];
x[0]=ADCINCVR_iGetDataClearFlag(); // get sample
…
for (i=0;i<3;i++)  software+=a[i]*x[i];
```

Figure 3. Algorithm of the digital filter by software.

3.3 Digital Low-Pass Filter by Hardware

In order to implement the hardware digital filter, an additional system resource of the PSoC device, called MAC (multiply-accumulate unit) is used. The MAC block is a dedicated logic circuit used in digital signal processing (DSP) devices that implements a multiplier followed by an adder and an accumulator register, which stores the result.

This way, the MAC unit of PSoC device was chosen to implement the third copy of the filter, in order to implement the Diversity TMR technique. The MAC unit presents three main data registers. The signal samples from the ADC are stored in the MUL_X register and the filter coefficients are stored in the MAC_Y register. The MAC unit automatically performs the multiplication and stores the product into the ACC_DR0 register. Figure 4 shows the configuration code of the MAC unit.

3.4 The Voter

An efficient voting mechanism for digital signals is designed to select as system output the value that is the majority among the redundant outputs [17]. The extension of TMR concept to the analog case requires the design and implementation of a voting scheme suitable for dealing with analog signals.

```
MAC_CL0=0;
…
MUL_X=x[0]; MAC_Y=4;
MUL_X=x[1]; MAC_Y=7;
MUL_X=x[2]; MAC_Y=4;
hardware=ACC_DR0;
```

Figure 4. Configuration of the digital filter by hardware.

The voting scheme adopted in this work was made in software and consists in three comparisons plus a decision element, as seen in the representative scheme in Figure 5. An A/D converter is used to digitize the output of the analog filter and mutual comparisons are made between the outputs of the three TMR blocks (V_1, V_2 and V_3), generating the selection signals for the decision element. The comparisons are actually subtractions between each signal sample generated by the filters, producing three error signals (e_1, e_2 and e_3). Based on the error signals the decision element evaluates the correctness of the signals V_1, V_2 and V_3, and selects one correct signal to be presented at the system global output.

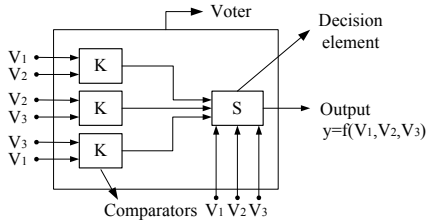

Figure 5. Voting scheme architecture.

Taking this into account, if the difference between two samples (from different blocks) exceeds a certain pre-defined tolerance, the voter may detect an error. If this error affects only one of the filters, the decision element is able to identify the correct blocks selecting one of them to be switched to the system output. The program code which implements the voter algorithm is shown in Figure 6. As it can be seen in Figure 6, a multiplexer is used to select one of the filter outputs according to the results of the subtractions.

```
e1=abs(V1-V2); // Subtractions of
e2=abs(V2-V3); // digitized signals
e3=abs(V3-V1); // of the 3 filters

if (e1<=7 || e3<=7)
  {   MUX_OUT_InputSelect(MUX_OUT_PORT0_3);
  }
if (e1<=7 || e2<=7)
  {  MUX_OUT_InputSelect(MUX_OUT_PORT0_5);
     DAC9_WriteStall(software);
  }
if (e2<=7 || e3<=7)
  {   MUX_OUT_InputSelect(MUX_OUT_PORT0_5);
      DAC9_WriteStall(hardware);
  }
```

Figure 6. Voter algorithm.

3.5 Functional Verification of the MS-DTMR

Firstly, some practical experiments were performed in order to verify the functionality of the MS-DTMR system. A functional deviation was injected into each filter, while the signals at the system outputs were monitored. In both digital filters an error was introduced by changing a coefficient of the transfer function, while a modification in the value of a capacitor was injected in the analog SC filter. Figure 7 shows the acquired waveforms during one of these experiments, in which a fault was injected into the digital filter implemented by software. Considering single functional injections in all DTMR copies, it was observed that if one of the three filters presents a fault the others are kept in normal operation, in a way that the voter is able to select one of both correct signals.

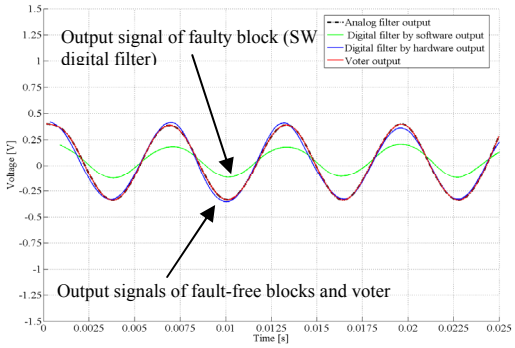

Figure 7. Outputs of DTMR blocks and voter whit a functional deviation in the digital filter by software.

The tolerance window considered for the error signals is 7 LSBs, as it can be seen in Figure 6. With the resolution of the output DAC being 9 bits, and with a 5V full scale reference, the voltage correspondence of the tolerance window is 68.49mV. The suitable value of this window should be evaluated by the designer, since different applications and required reliability levels may demand different tolerance windows.

4. FAULT INJECTION CAMPAIGN

4.1 Fault Modeling

In order to investigate the degree of reliability achieved by the proposed MS-DTMR scheme, a fault injection campaign was carried out. The considered fault model comprises two groups of faults: deviations in the capacitor values, in the analog SC filter, and bit inversions in the outputs of the registers employed in the implementation of the digital filters. Faults in the voter are not considered in this work.

The values of capacitors in the analog blocks are programmed through switching weighted groups of unit capacitors (programming capacitor banks), and the state of the programming switches is defined through the programming memory. Therefore, it is considered that both types of faults may be caused by bit inversions.

Additionally it is possible to classify the faults that affect the digital system copies in two different subgroups: *permanent* and *non-permanent*. Here, faults affecting the registers which contain the coefficients of the filter are considered permanent, since the values of the coefficient don't change along with the signal processing, thus, these registers are not reloaded, unless the system is reset. Conversely, the registers used to store the samples of the processed signal are updated at each conversion cycle (new signal sample). Therefore, the time duration of these sort of fault will be approximately *1/SampleRate* (see Equation 1), and its impact on the system functionality will be less severe than permanent faults.

With this concern, the fault injection experiments were directed to investigate double faults affecting different TMR blocks. Considering the bit inversion fault model it is possible to inject:

- 123 faults in the analog filter (all permanent);
- 96 faults in the digital SW filter (24 permanent and 72 non-permanent);
- 80 Faults in the digital HW filter (24 permanent and 56 non-permanent).

This way, considering double faults affecting different DTMR blocks the total universe of faults is calculated by obtaining all possible 2-by-2 combinations of faults between two different DTMR modules, as described below:

$$T_{\#F} = (F_{DSW} \cdot F_{DHW}) + (F_{DSW} \cdot F_{AN}) + (F_{DHW} \cdot F_{AN}) = (96 \cdot 80) + (96 \cdot 123) + (80 \cdot 123) = 29328 \quad (3)$$

where: $T_{\#F}$ is the total number of faults and F_{DSW}, F_{DHW} and F_{AN} are the individual number of fault possibilities of each block.

4.2 Fault Injection

Faults were injected in the analog module by changing the values of the programmable capacitors, in such a way that single bit inversions in the capacitors banks are emulated. In both digital filters all registers employed in the signal processing (including auxiliary registers used in loop counter variables) are masked by a XOR function with an 8-bit constant, as follows:

$$R_x = R_x \otimes (Inv\_Mask) \quad (4)$$

where R_x is the register to be modified by an injected fault and *Inv_Mask* can assume the values 0, 1, 2, 4, 8, 16, 32, 64, 128. This way, *Inv_Mask*=0 reproduces the fault-free case and the values 1 to 128 inject single inversions from the LSB to the MSB. Permanent faults were injected in a way that its effect persists until the next system reprogramming, while non-permanent faults were inject periodically by adding a counter to the main code.

Since the approach to inject faults in these experiments consists in manual injection, by changing the programming code and downloading the object file to the board, it becomes a time consuming task (approximately 100 faults per hour). Thus, a different method is used to evaluate multiple fault effects without exhaustively injecting all 29328 faults from the model.

In this approach each module that compound the DTMR scheme (each version of the filter) is duplicated and single faults are exhaustively injected in one of the duplicated blocks. Than, for each module it is computed the number of faults that effectively generate an error, deviating the output signal in a way that the comparators of the voter are able to detect it.

Therefore, the estimation of multiple faults that will not be tolerated by the DTMR is made by calculating the number of 2-by-2 combinations of these set of faults.

Figure 8 depicts the fault injection setup for the analog module, were the 123 faults from the fault model are injected and the error rate is computed. The same procedure is repeated for the digital filters (SW and HW) by injecting 96 and 80 faults, respectively, into each module, according the adopted fault model.

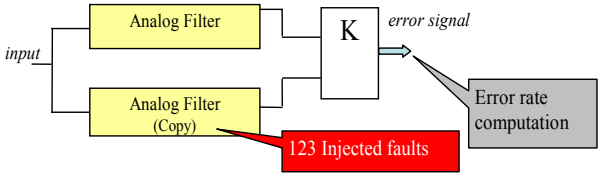

Figure 8. Setup of fault injection experiments for the analog system copy.

5. EXPERIMENTAL RESULTS

Table 1 summarizes the results obtained from fault injection. For each DTMR module the total number of injected single faults, the number of observed errors and the obtained percent error rate are shown.

Table 1. Results of individual single fault injection.

	SW Dig. Filter	HW Dig. Filter	AN Filter
Inj. Faults	96.00	80.00	123.0
Errors	28.00	20.00	61.00
Error rate	29.16%	25%	49.59%

It is possible to conclude, from Table 1, that the analog filter is the most sensitive to the injected faults, since almost half of them generate an error. On the other hand, the digital filter by HW can tolerate 75% of the injected faults. For sake of conciseness, discrimination between permanent and non-permanent faults is not explicit in Table 1. In fact, for the 28 observed errors in the digital filter by SW, 20 are due to permanent faults. Interestingly, 8 non-permanent faults generated permanent errors. This is because these faults changed the control variable of a SW loop (like the variable *count* in Figure 3), making the system to be locked into a loop or never getting to it.

Considering that mutual 2-by-2 combinations of the faults that individually generated an error in each module will lead the DTMR system to fail, the number of not tolerated double faults can be estimated by:

$$T_{\#E} = (E_{DSW} \cdot E_{DHW}) + (E_{DSW} \cdot E_{AN}) + (E_{DHW} \cdot E_{AN}) = (28 \cdot 20) + (28 \cdot 61) + (20 \cdot 61) = 3488 \quad (5)$$

where: $T_{\#E}$ is the total number of observed errors due to double faults in the DTMR and the terms E_x are the number of observed errors for each block under individual single fault injection, according to Table 1.

Therefore, by calculating the ratio $T_{\#E}/T_{\#F}$ one concludes that only 11.89% of the considered double faults are not tolerated by the proposed DTMR. It is clear that the DTMR presents a higher reliability if compared to each version of the filter without redundancy.

However, more consistent conclusions are drawn when comparing the DTMR to traditional TMR systems. In this case three different TMR systems were considered, comprising the triplication of each filter version (SW, HW and Analog). By extending equations (4) and (5) to the case of traditional TMR, and with the results of Table 1, it is possible to estimate the number of tolerated double faults for each traditional TMR and compare to the DTMR results. These results are show in Figure 9.

Figure 9 shows that both SW and HW filters in traditional TMR schemes present a slightly better reliability considering the fault model adopted in this work. However, the DTMR showed better results than the traditional analog TMR. In fact, when considering the same fault model to both cases, the reliability level of the DTMR will be always less than or equal the reliability of the most reliable block used in the DTMR construction, as can be proved by simple mathematics.

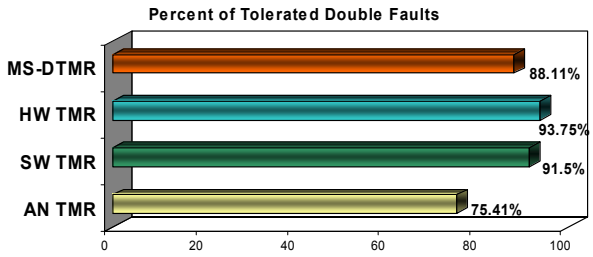

Figure 9. Comparison between the tolerance levels of the MS-DTMR scheme and traditional TMR systems.

In this specific case, the high error rate of the analog block (shown in Table 1) is the main responsible for reducing the reliability of the DTMR, if compared to the results obtained for the traditional TMR of both digital filters.

However, since the DTMR modules are different, distinct defect mechanisms will affect the system modules in different ways. As an example, one can consider the TID effects in radiation environments.

Though TID may affect the analog and digital parts of a system similarly at transistor level, its effects at system level are different in both domains. Therefore, it is possible to assume that the design diversity naturally adds an extra degree of resilience to the TMR.

6. CONCLUSIONS

In this work the concept of redundancy with diversity was applied to a mixed-signal programmable system as a proposal to increase the system reliability. The DTMR system was prototyped in a single commercial programmable device. The system consists in three different versions of a low-pass filter (analog, digital by software and digital by hardware), in addition to a voter system.

Fault injection experiments show that the system can tolerate 100% of single faults and 88.11% of double faults from the considered fault model.

When comparing this result to traditional TMR schemes built with the three filter architectures considered in this work, one observes a gain in the system reliability with relation to the analog TMR, but a slightly worse result if compared to the digital filters with traditional TMR.

However, triplicating a digital filter can be costly since three processors or three MAC units will be needed. In this context, state-of-art programmable mixed-signal devices may offer sufficient resources to build MS-DTMR systems, without the need to add modifications at design level and without significant increases in system area or cost.

Additionally, the design diversity increases the degree of reliability of the system, since each redundant module may have different levels of resilience associated to the diverse fault generation mechanisms and sources.

Finally, despite the block of interest in this work be a low-pass filter, the increasing on the capabilities of state-of-art MS programmable devices allow the extension of the MS-DTMR technique to other kind of MS blocks, such as PWMs and integrators, for instance.

7. ACKNOWLEDGEMENTS

This research is supported by the CNPq Brazilian Research Agency by contract number 180115/2009-0.

8. REFERENCES

[1] Velasco, R.; Fouillat, P.; Reis, R. (Org.). 2007. Radiation effects on embedded systems. Springer.

[2] Wang, F.; Agrawal, V. D. 2008. Single event upset: an embedded tutorial. IEEE 21st Intern. Conference on VLSI Design (Jan., 2008).

[3] Kastensmidt, F. L.; Neuberger, G.; Carro, L.; Reis, R.; Hentschke, R. 2004. Designing fault-tolerant techniques for SRAM-based FPGAs. IEEE Design and Test of Computers, USA, v. 21, n. 6, p. 552-562 (Nov., 2004).

[4] Balen, T. R.; Leite, F.; Kastensmidt, F. L.; Lubaszewski, M. S. 2009. A self-checking scheme to mitigate single event upset effects in SRAM-based FPAAs. IEEE Transactions on Nuclear Science, vol. 56, n. 4, p. 1151-1162 (Aug. 2009).

[5] Vial, J.; Bosio, A.; Girard, P.; Landrault, C.; Pravossoudovitch, S.; Virazel, A. 2008. Using TMR architectures for yield improvement. Intern. Symposium on Defect and Fault Tolerance of VLSI Systems, p. 7-15 (Oct., 2008).

[6] Morgan, K. S.; McMurtrey, D. L.; Pratt, B. H.; Wirthlin, M. J. 2007. A comparison of TMR with alternative fault tolerant design techniques for FPGAs. IEEE Transactions on Nuclear Science, vol. 54, n. 6, p. 2065 – 2072 (Dec., 2007).

[7] Anghel, A.; Alexandrescu, D.; Nicolaidis, M. 2000. "Evaluation of a soft error tolerance technique based on time and or hardware redundancy. Proc. of IEEE Integrated Circuits and Systems Design, pp. 237-242 (Sept., 2000).

[8] Elnozahy, E.; Melhem, R.; Mosse, D. 2002. Energy-efficient duplex and TMR real-time systems. 23rd Real-Time Systems Symposium, IEEE, p. 256-266 (Dec., 2002).

[9] Lima, F.; Carmichael, C.; Fabula, J.; Padovani, R.; Reis, R. 2001. A fault injection analysis of virtex FPGA TMR design methodology. 6th European Conference on Radiations and Its Effects on Components and Systems, IEEE, Grenoble, France, 10-14, p. 275-282 (Sept., 2001).

[10] Avizienis, A.; Kelly, J. P. J. 1984. Fault tolerance by design diversity - concepts and experiments. Computer. Vol. 17, pp. 67-80 (Aug., 1984).

[11] CYPRESS Microsystems. 2002. CY8C2XXXX Family Datasheet. Available in www.cypress.com.

[12] Kim, H.; Shin, K. G. 1996. Design and Analysis of an optimal instruction-retry policy for TMR controller computers. IEEE Transactions on Computers, vol. 45, n. 11, p. 1217-1225 (Nov., 1996).

[13] Radu, M.; Pitica, D.; Posteuca, C. 2000. Reliability and failure analysis of voting circuits in hardware redundant design. International Symposium on Electronic Materials and Packaging, p. 421-423 (Nov., 2000).

[14] Yoon, J.; Kim, H. 2000. Time-redundant recovery policy of TMR failures using rollback and roll-forward methods", IEEE Proceedings on Computers and Digital Techniques, vol. 147, n. 2, p. 124-132 (Mar., 2000).

[15] Mitra, S.; Saxena, N. R.; McCluskey, E. J. 1999. A design diversity metric and reliability analysis for redundant systems. Proceedings of International Test Conference, p. 662-671 (Sep., 1999).

[16] Shim, B.; Shanbhag, N. R. 2006. Energy efficient soft error-tolerant digital signal processing. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 14, n. 4, p. 336-348 (Apr., 2006).

[17] Vega, D. V. G. 1995. Diseño para testabilidad y tolerancia a fallos en circuitos analógicos. Thesis (doctorate in physics), Sevilla University.