IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 27, 2020, accepted April 28, 2020, date of publication May 6, 2020, date of current version May 15, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2991882

Radiation Hardened Digital Direct Synthesizer With CORDIC for Spaceborne Applications

LUIS ALBERTO ARANDA, FRANCISCO GARCIA-HERRERO, LUIS ESTEBAN, ALFONSO SÁNCHEZ-MACIÁN, AND JUAN ANTONIO MAESTRO, (Senior Member, IEEE)
ARIES Research Center, Universidad Antonio de Nebrija, 28040 Madrid, Spain
Corresponding author: Luis Alberto Aranda (laranda@nebrija.es)

ABSTRACT The Coordinate Rotation Digital Computer algorithm (CORDIC) is a simple mechanism to compute a set of elementary functions, such as trigonometric functions, using fixed-point devices. It is widely adopted, also in applications running in harsh environments such as space, where radiation is a cause of errors in nanoelectronic devices. A single event upset in a configuration bit of a Field Programmable Gate Array (FPGA) can completely change the behavior of the implemented circuit, so it is important to detect and reconfigure the FPGA when this happens. Dual modular redundancy is the typical method to detect errors in electronic circuits, but it has an important overhead in area and power consumption and it does not provide any additional functionality apart from the activation of the FPGA reconfiguration trigger in presence of error. This paper presents two ad-hoc techniques to protect the Digital Direct Synthesizer with CORDIC when it is implemented into an FPGA, with limited overhead in terms of area and power consumption when compared with the traditional solution. The first solution slightly increases the percentage of undetected errors, about 11%, reducing to almost half the area overhead of the circuit. The second solution introduces a trade-off between the percentage of error detection and the precision of the trigonometric output of the CORDIC by means of a polymorphic structure with lower area resources than the existing solutions. This last proposal allows the system to increase the precision of the digital synthesis signal under absence of errors or to activate the error protection in scenarios with external disturbances such as radiation.

INDEX TERMS CORDIC, digital signal processing, dual modular redundancy, fault tolerance, radiation.

I. INTRODUCTION

The Coordinate Rotation Digital Computer (CORDIC) algorithm [1] is a hardware efficient algorithm that can compute a set of elementary functions [2] including trigonometric and logarithmic functions, complex number multiplication and division, or matrix inversion. The power of the CORDIC algorithm relies on the fact that it can be easily implemented in fixed-point devices such as Application Specific Integrated Circuits (ASICs) and Field Programmable Gate Arrays (FPGAs) [3].

The CORDIC algorithm is present in digital modulation [4] or the robotics field [5], and can be used to compute the discrete cosine transform [6], the Hartley transform [7], or the singular value decomposition [8]. Moreover, the CORDIC algorithm is also used in space applications where $sin(\omega t)$ and $cos(\omega t)$ signals need to be digitally generated. This process is called Digital Direct Synthesis (DDS) [9]–[11]. For instance, digital Quadrature Amplitude Modulation (QAM) transceivers require a CORDIC module to digitally down-convert the received signals to intermediate frequencies or base-band (i.e. digital mixer), and heterodyne receivers, which are extensively used in satellites, use it for the frequency synthesizers. Another example of the current and future use of the CORDIC algorithm is the Laser Interferometer Space Antenna (LISA) gravitational wave interferometer that will use a CORDIC-based phase-meter to compute the optical path length differences [12], [13].

In harsh environments such as space, the electronic devices may fail due to the elevated radiation level if they are not properly protected. This means that the previously mentioned systems may stop working or start behaving in an erroneous way. An inexpensive alternative to traditional radiation-hardened integrated circuits is the implementation of customized hardware designs in Commercial Of-The-Shelf (COTS) devices that take into account the radiation effects.

The associate editor coordinating the review of this manuscript and approving it for publication was Jenny Mahoney.

These protected designs are used to mitigate or prevent errors caused by the environment such as Single Event Upsets (SEUs) or cumulative errors.

In digital architectures implemented on FPGAs, a method to detect the misbehaviour of the circuit is required. The typical method used to detect errors is called Dual Modular Redundancy (DMR) or duplication with comparison (DWC) [14], [15], and consists in using a redundant copy of the circuit and extra voting logic. If an error affects one of the two copies, it will be detected by comparing the output of both copies [16]. However, critical space applications have severe power and area restrictions that in some cases are not possible to achieve with classical protection approaches such as the DMR.

Apart from the actual constraints, there are also some kinds of applications that can tolerate a certain error rate, or in other words, they do not need that 100% of the samples are correct. An example of this is video or audio transmission, in which a certain number of wrong samples would degrade the quality of the output, but it could be acceptable if the error rate is reasonable. The bottom line is that using exhaustive protection techniques like DMR increases the area and power overheads considerably, in exchange for a full protection that may not be needed.

This leads to the concept of ''adaptive protection'', by which a system would adopt a weaker protection technique if the application requirements and the environment allow to do so, thus reducing the overheads. In this way, there can be a wide set of ''intermediate'' protection techniques (ranging from the unprotected circuit to the DMR-like full protection case), that designers should explore in order to choose the most appropriate for each application, and therefore produce a custom-tailored solution. Note that this adaptive protection approach fits very well the FPGA implementation, since being able to dynamically reconfigure the device provides the ability to change the protection level when needed.

In this context, several customized protection techniques will be analyzed and developed in this paper to either reduce the overhead of the protection circuits and by hence the area and the power consumption of the global system [17], [18], or to take advantage of the overhead to provide a new or an improved functionality. These ad-hoc techniques can be divided into Algorithmic-Based Fault-Tolerance Techniques (ABFT), in which the error detection is achieved by exploiting arithmetic properties of the algorithm [19], [20], and techniques based on exploiting structural properties of the design to create a protection scheme [16]. Other approaches, which are out of the scope of this paper, are based on analog circuits that give support to digital designs to detect faulty behaviors [21] and powerful error correction codes to protect the FPGA hardware from soft errors, which involve iterative processes at a larger timing cost [22], [23].

In this paper we present two protection techniques as an alternative to the DMR protection approach. As previously pointed out, the idea is that DMR would still be used whenever strong fault-tolerant restrictions apply for the specific application, while the proposed techniques can be set in place in scenarios where the application can trade a slight increase on error rate in exchange for some area and power savings (i.e. in Cubesats where there are important power restrictions).

The first technique is an ABFT technique based on using the trigonometric properties of the CORDIC algorithm. A circuit that performs the operation $sin^2(\omega t)+cos^2(\omega t)$ over the In Quadrature (IQ) output components of the CORDIC has been included. The output of this circuit must be a low amplitude oscillation (due to the limited precision of the CORDIC) around a mean value of 1. Errors can be detected by setting an appropriate threshold in the output comparison logic. The objective of this technique is just to keep area resources as low as possible.

The second one is a polymorphic protection that exploits the structure of the CORDIC algorithm. Basically it consists of two cascade CORDICs that can be configured as a Digital Direct Synthesizer with more precision, when no faults are expected, or as two CORDICs where the inputs of the second one are reversed to generate a signal for error detection. It will be shown how the outputs of the second CORDIC, in the error detection mode, have to be constant in the absence of errors, matching with the input constant of the first CORDIC. In the paper, it is concluded that, with these two approaches, acceptable results can be obtained with a fraction of the resources needed for the implementation of a full DMR.

This paper is structured as follows: in Section II the CORDIC algorithm and the architecture that has been implemented for DDS are explained. In Section III the circuits designed to detect the errors in the architecture are shown. The error injection setup is detailed in Section IV and in Section V the error injection results of the customized circuits are presented. Area and detection rate are analyzed and compared between the different architectures. In Section V the main conclusions of the paper are summarized.

II. CORDIC ALGORITHM FOR SIN AND COS GENERATION

In this Section, the CORDIC algorithm and the particular architecture implemented for DDS are explained in detail.

A. CORDIC ALGORITHM

The basic operation of the CORDIC algorithm is to rotate vectors. Given a vector in Cartesian coordinates, $\mathbf{v} = [x_0, y_0]$, where x_0 and y_0 are the abscissa and ordinate respectively, the CORDIC algorithm rotates it an angle φ to obtain a new vector $\mathbf{v'} = [x_n, y_n]$.

The rotation is performed iteratively by small angles as:

$$\alpha_i = \tan^{-1} 2^{-i} \simeq \frac{1}{2^i} \tag{1}$$

Therefore, the total rotated angle is:

$$\varphi = \sum_i d_i \cdot \alpha_i \tag{2}$$

where d_i can be either $+1$ or -1 depending on the direction of the rotation. Every rotation is called pseudo-rotation,

and their equations are given by:

$$x_{i+1} = \left(x_i + d_i \cdot 2^{-i} \cdot y_i\right) \tag{3}$$

$$y_{i+1} = \left(y_i + d_i \cdot 2^{-i} \cdot x_i\right) \tag{4}$$

$$z_{i+1} = z_i - d_i \cdot \alpha_i \tag{5}$$

where, z_i is called angle accumulator and includes all the rotations made in the previous iterations.

The pseudo-rotation for the ith iteration alters the magnitude of the rotated vector, so it must be scaled by a factor K_i:

$$K_i = \left(1 + \tan^2\left(d_i \cdot \alpha_i\right)\right)^{(-1/2)} \tag{6}$$

where, K_n is the scaling factor for n stages. Its value is given by the following equation:

$$K_n = \prod_{i=0}^{n} K_i \tag{7}$$

For the rotation of the angle φ, the small angles α_i that must be added or subtracted during each rotation can be stored in a Look-Up Table (LUT). Their values are shown in Table 1.

TABLE 1. Values of α_i depending on the stage, [3].

i	$\alpha_i = \tan^{-1}\left(1/2^i\right)$ [rad]
0	0.7854
1	0.4636
2	0.2450
3	0.1244
4	0.0624
5	0.0312
6	0.0156
7	0.0780
8	0.0039
9	0.0020

Equations (3) to (5) can be implemented in an FPGA or in an ASIC using the parallel architecture shown on Fig. 1. However, this architecture is only capable of rotating a maximum angle of $\pm\alpha = 1.7401$ rad for 10 stages. Given the values of Table 1 for 10 stages the relative error in the sin and cos waves would be of $3.15 \cdot 10^{-4}$.

This issue can be easily solved by adding an initial stage that makes a first iteration of $\pm\pi$ as follows:

$$x_{i+1} = -d_i \cdot y_i \tag{8}$$

$$y_{i+1} = +d_i \cdot x_i \tag{9}$$

$$z_{i+1} = z_i - d_i \cdot \pi/2 \tag{10}$$

B. DDS FOR SIN AND COS GENERATION

Based on the parallel architecture given on Fig. 1 and the angle extension described in Equations (8) to (10), the 10 stage CORDIC algorithm shown on Fig. 2 has been designed to generate digitally sin (ωt) and cos (ωt) signals.

The input to x_0 is $1/K_9$, where K_9 is the scaling factor for 10 stages, $1/K_9 = 0.607253$. The input to z_0 is a saw-tooth signal generated using the integrator shown on Fig. 2 which is bounded in the interval $[-\pi, \pi]$.

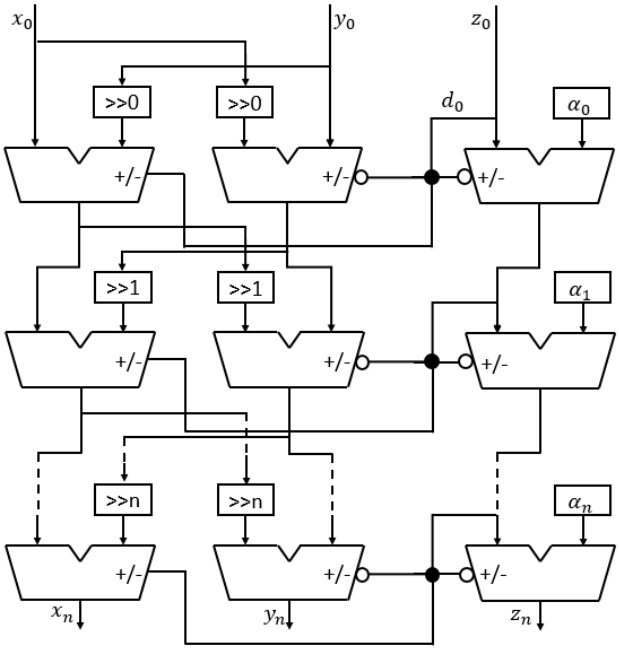

FIGURE 1. Parallel architecture of a CORDIC algorithm suitable for fixed point implementation.

This circuit is used in heterodyne systems, where the signals are down-converted to base-band by multiplying them by cos $(2\pi \cdot P \cdot f_{min})$ and sin $(2\pi \cdot P \cdot f_{min})$ that depend on constant P.

As it was mentioned in Section I, the CORDIC algorithm is extensively used in all digital QAM transceivers and in heterodyne interferometry. For instance, in the LISA gravitational interferometer it will be used to down-convert the optical frequencies to base-band (∼40 MHz).

The detailed pipeline is shown on Fig. 3. This is the golden design that has been used to evaluate the different protection circuits proposed in this paper.

III. PROPOSED PROTECTION TECHNIQUES

In the next subsection two protection techniques are proposed: an ABFT technique based on trigonometric properties of the CORDIC algorithm, and another one based on structural properties of the hardware design that extends the functionality in addition to the error detection.

It is important to remark, before explaining the techniques, that an error in an ASIC device would temporarily affect the behavior of the circuit. Once the error is flushed out, the circuit would continue its normal operation [24]. The error model in an FPGA is more complicated: a persistent error can change the configuration of the FPGA and therefore the behavior of the circuit will be erroneous requiring the reconfiguration of the device to restore its normal operation [25], [26]. This means that the fault caused by a soft error in the configuration memory of the FPGA would not

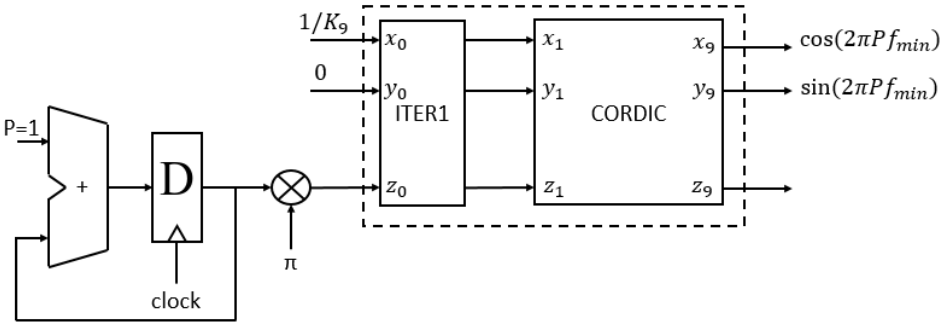

FIGURE 2. Unprotected CORDIC parallel architecture for sin (ωt) and cos (ωt) generation.

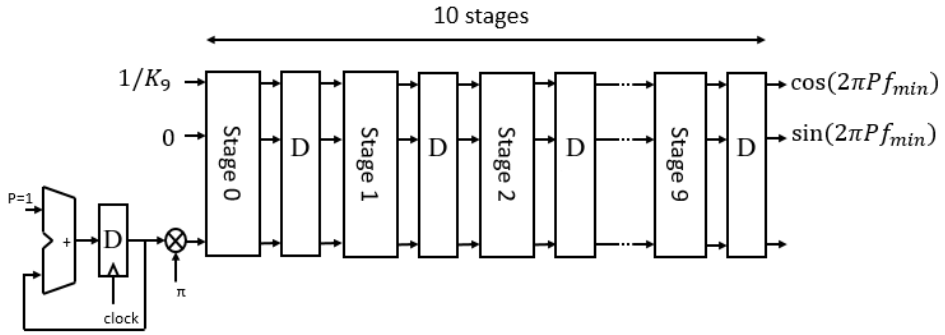

FIGURE 3. Detailed view of the CORDIC parallel architecture for sin (ωt) and cos (ωt) generation.

be acceptable for the CORDIC (or the application using it) as the change in the circuit makes the error persist indefinitely.

One last comment is that the proposed techniques have been oriented to the single event mode in FPGAs, i.e., it is assumed that only one error will be present in the system at any time. This is a common assumption in the literature, since single errors represent the majority of events in space applications. In [27], it can be seen that for FPGAs using the 28*nm* technology node, Multiple Bit Upsets (MBUs) are less frequent than Single Event Upsets (SEUs). Despite that, the following techniques would reach a higher error detection rate in the presence of MBUs, as they look for divergences from two constant values. With MBUs, a higher number of errors will increase the probability of larger differences to be produced in the outputs. Note that each error in one iteration has a greater impact in the next one. For example, when a single bit event happens in the first iteration, that error increases to a more significant one in the next iteration due to the shift of bits in the following stage, having greater impact (see Fig. 1).

A. TRIGONOMETRIC PROTECTION

This first technique demonstrates how an ABFT approach can be followed to protect a circuit with algorithmic properties. An interesting and simple protection technique that can be applied to this particular case (generation of $\cos(2\pi \cdot P \cdot f_{min})$ and $\sin(2\pi \cdot P \cdot f_{min})$) is the following trigonometric property:

$$\cos^2(2\pi \cdot P \cdot f_{min}) + \sin^2(2\pi \cdot P \cdot f_{min}) = 1. \quad (11)$$

In order to implement the previous equation in hardware, a circuit that adds the squared outputs has been included in the system (see Fig. 4). If the sum of the outputs verifies the property given in Eq. (11) it is assumed that there are no errors in the CORDIC. If it does not, then it is implied that an error has occurred.

However, the protection is not as simple as a straight comparison due to the precision of the circuit. In other words, the output of the circuit will have a small oscillation with amplitude γ, due to the finite precision of the CORDIC, around a mean value equal to one, $1 \pm \gamma$, i.e. for a 10-stage CORDIC and 8-bit inputs $\gamma = 0.0036$. To handle this, the protection will compare if the output is in the $[1 - \gamma, 1 + \gamma]$ range. In the event that there is an error, the output of the protection circuit will be most of the times out of the $[1 - \gamma, 1 + \gamma]$ boundary. However, it may happen that an error with a magnitude lower than γ happens, and in this case the technique would not detect it, as will be seen later in the results of the conducted experiments. Finally, regarding the timing, it should be mentioned that two extra cycles are required, one to register the output of the multipliers and another one to register the output of the comparator logic.

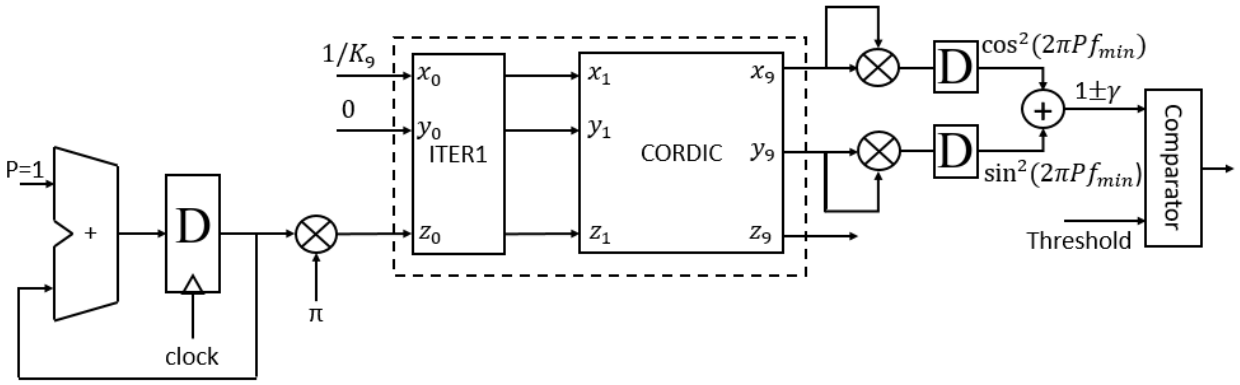

FIGURE 4. Proposed trigonometric protection.

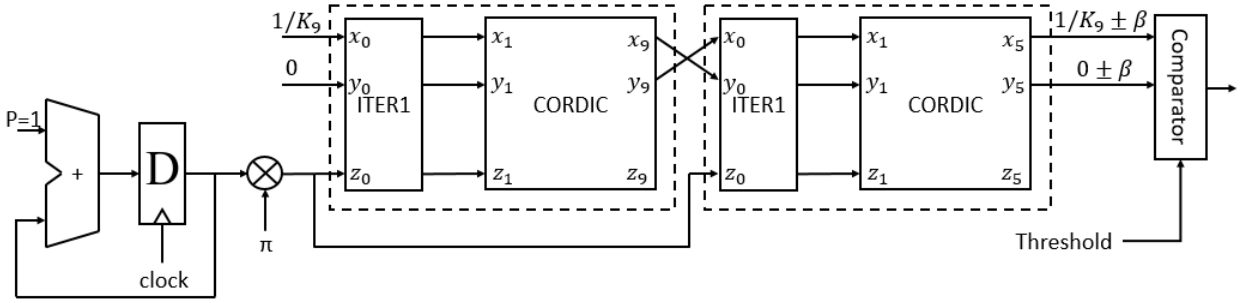

FIGURE 5. Inverse CORDIC protection architecture, where 6 stages have been used to implement the protection circuit.

B. INVERSE CORDIC POLYMORPHIC PROTECTION

In certain cases, some ad-hoc protection schemes that occupy less area than a DMR can also be found by exploiting the structural properties of the algorithm when implemented in hardware. The technique used in this paper consists in using a second CORDIC that inverts the effects of the first one. In other words, the outputs of the CORDIC are fed into another one that reverses the process, and therefore the output of the latter should match the original input, if no errors are present.

Structural-based protection circuits can still achieve a high error detection rate. In the case of the CORDIC setup shown on Fig. 5, the outputs of the first CORDIC are fed into the second one in reverse order. The outputs of the second CORDIC are the inputs of the first one with small oscillations:

$$x_9 = \frac{1}{K_9} \pm \beta \tag{12}$$

$$y_9 = 0 \pm \beta \tag{13}$$

where β is the amplitude of the oscillation. In the absence of error both outputs will be bounded in the following intervals:

$$x_9 \in \left[\frac{1}{K_9} - \beta, \frac{1}{K_9} + \beta\right] \tag{14}$$

$$y_9 \in [-\beta, +\beta] \tag{15}$$

In the event that there is an error, the output of the protection circuit will be out of the bounds given in Equations (14) and (15). Therefore, an error could be easily detected and flagged by setting a threshold, i.e. for 9 stages and 8-bit inputs $\beta = 0.0016$ and for 4 stages and 8-bit inputs $\beta = 0.0139$. The thresholds have been established in all the examples and experiments provided in the paper according to the measures of the outputs of real FPGA's implementations. The margin of error was set comparing the outputs of the cycles of the DDS and the output of a Matlab and a System Generator golden model.

In Fig. 5, a 10-stage CORDIC pipeline is shown. It should be mentioned that the second CORDIC, the one used to detect the errors, does not necessarily need to have the same number of stages as the first one (see Fig. 6). A lower number of stages can be used to still achieve a high error detection rate, while reducing the area overhead introduced by the protection technique.

The precision of the CORDIC depends on the number of stages i used. Therefore, the fractional bit-widths of the different CORDIC taps must be increased accordingly to achieve the given precision. As a suggested methodology, once the number of stages and the CORDIC bit precision required by the application are fixed, a target detection rate should be defined and then, via experimentation, the number of bit-precision required for the different inputs and constant

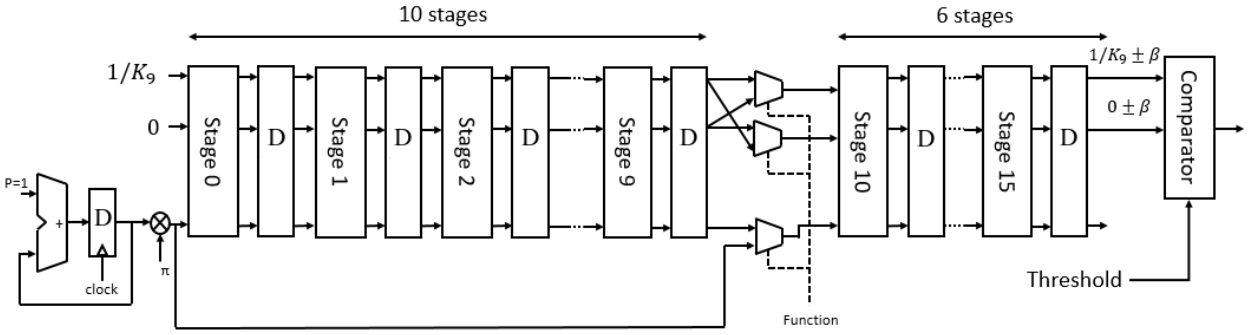

FIGURE 6. Stage detail of the inverse CORDIC protection architecture.

can be fixed to the minimum, to reduce consumption and area, i.e. in the next section, for comparison proposals, the 8-bit input was selected and data width grow through the different iterations until an 18-bit output, to avoid precision loss. At that point, increasing the number of bits will not improve the error rate detection and the overhead will be optimized. The overhead for the inverse CORDIC polymorphic protection is always proportional to the number of stages added to the original design.

On the other hand, it is important to remark that the detection-related CORDIC does not generate an output $1/K_j$, being j the number of stages of the detection-related CORDIC. The output in detection mode is always equal to the input of the CORDIC that is going to be protected, i.e. in this example $1/K_9$. Note that in the detection mode, the inputs of the CORDIC are not constant. They are sine/cosine signals, and the angle accumulator is equal to the first CORDIC, so the unit is not working as a DDS [3]. Hence the vector that is obtained at the output is related with the input that was applied in the first CORDIC. Then the number of stages of the additional CORDIC will only influence in the precision of x_9 and y_9, in other words, in the amplitude of the threshold β. The implementation of the proposed structural protection technique in this manner offers two advantages. On the one hand, as described before, it can be used as an error detection system. On the other hand, it could also be used to extend the number of stages of the original CORDIC and by hence to increase its precision if needed, in absence of error. Therefore, the hardware required for this protection techniques is, at the same time, creating a polymorphic circuit that can work in two different modes:

1) In environments or missions with low radiation, it can be used as an extended resolution CORDIC.
2) In environments or missions with high radiation, it can be split into a lower resolution CORDIC and an inverse CORDIC for error detection purposes.

In Fig. 6, the two mentioned modes or functionalities are selected by using a multiplexer where the select signal is labeled as "Function". In fact, we could go one step further and create a design that adaptively selects the number of stages used for the protection depending on the radiation level of the environment. In order to provide some insight about that, the effect of the inverse CORDIC stage variations in the error detection rate and the area resources has been studied in Section V. One additional consideration that has to be taken into account is that the injected errors can also affect the final comparator of the protection techniques. However, if a configuration memory error affects the mentioned comparator, a false positive may occur. These detected errors do not modify the outcome of the design but have to be corrected by reconfiguring the FPGA to restore the error detection capabilities of the protection technique itself.

Finally, regarding the timing, it should be mentioned that only the first, middle and final stages have pipeline registers. Therefore, the delay added to the unprotected CORDIC with this design will be equal to 4, three registers for the protection circuit and one to register the output of the comparison logic.

For the sake of comparison with traditional protection techniques, a DMR scheme has also been implemented. DMR is the usual approach followed in FPGA designs when error detection is needed. This is due to the previously mentioned FPGA error model, in which configuration errors have to be removed by rewriting the correct bit in the configuration memory, typically by reconfiguring the device. A DMR consists of using an identical redundant system that uses the same inputs as the first one (see Fig. 7).

In this context, the outputs of both copies should be identical. In the event that they are different, it means that an error has occurred in either one of the copies or in the extra voting logic. The area overhead for this type of architectures is typically higher than 100% because extra logic is required to compare the outputs. In FPGAs, the error detection rate obtained with a DMR scheme is slightly lower than 100% because there are also errors that the technique cannot detect, which are those affecting the input/output routing and the extra logic. Additionally, routing errors such as the generation of open/bridges between redundant branches (e.g. when sharing a multiplexer) can produce undetected errors [28]. Finally, and in order to register the output data, an extra clock cycle is required.

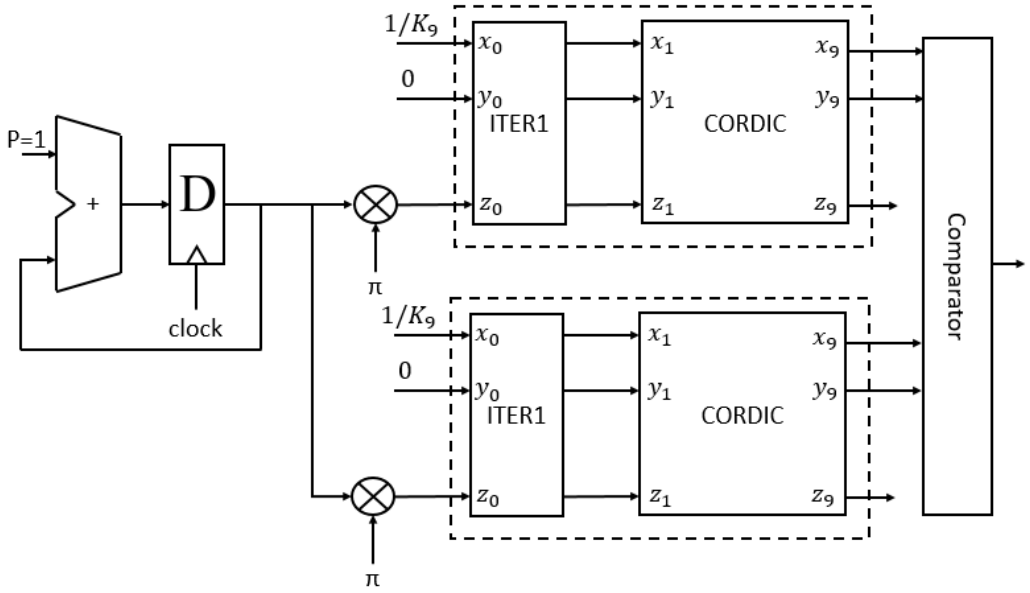

FIGURE 7. DMR protection.

IV. EXPERIMENTAL SETUP

The protection circuits described in the previous section (trigonometric protection, inverse polymorphic protection, and DMR), as well as the unprotected CORDIC parallel architecture shown in Fig. 2 with 10 stages (which will serve as the reference design) have been designed using VHDL and implemented in an Digilent Nexys 4 DDR Artix-7 FPGA. The fault injection process has been conducted by using the Xilinx Soft Error Mitigation (SEM) IP Controller [29]. With this module, and using a MATLAB script running in a computer to control the IP, exhaustive fault injection campaigns[1] have been performed for each design in which every design-related configuration memory bit is tested. In particular, single-error fault injection campaigns have been carried out. The procedure is as follows:

1) An error is injected using the SEM IP in one of the essential bits of the design. An essential bit is a configuration bit that may produce an error. The injection addresses of the DUT have been obtained with the ACME tool [30].
2) The circuit is exercised and the output is compared with the previously calculated golden (error-free) output during 10^3 periods of the sine/cosine generated signal. The golden output is the output of the unprotected CORDIC parallel architecture of Fig. 2. The behavior of the circuit is logged (the results of the simulation are stored in a file) and the previously injected error is removed.
3) A new error is injected in a different essential bit of the design, and the process is repeated.

[1]Exhaustive campaign means that errors are injected in all the frames of the configuration memory where the CORDIC design is implemented.

The campaign ends when all the essential bits of the design are injected and tested, performing an exhaustive fault injection campaign. Since all the essential bits are targeted, each one of the implementations receives a different number of injections, being the circuits that occupy more area the ones that receive more errors. This is consistent with the fact that larger circuits offer a larger cross-section to radiation.

As mentioned at the beginning of this section, the trigonometric, the inverse polymorphic, the DMR, and the unprotected CORDIC designs have been fully characterized following the described procedure. In addition, and in order to test the advantages of the polymorphic design, six different inverse CORDIC designs (with stages ranging from 4 to 9) have been characterized. The results are presented and discussed in the next section.

V. RESULTS

In Table 3 the area occupied, the error detection rate, defined as the number of errors that are detected by our proposal respect to all the error injections, and the area overhead of the different algorithms, are shown. It is very important to highlight that any divergence from the golden model is considered an error, regardless of the error affecting the most significant bit or the less significant bit, or its impact in the final application. So, with the applied error detection rate we are considering the worst-case scenario. It can be seen in this table that the best area/detection rate trade-off from the two techniques presented is obtained with the trigonometric protection one, with an area overhead of 49% with respect to the parallel CORDIC architecture of 10 stages and a detection rate of nearly 88%. A 100% error detection rate is not reached because the output of the protection circuit is limited by the

precision of the CORDIC and is not a constant but rather a small oscillation. In other words, since the technique does not do a straight comparison but it compares within a range, it may happen that an error with a small magnitude (within the threshold) is not detected and accepted as fine. Nevertheless, if this happens, the error magnitude would be limited to the aforementioned threshold.

Regarding the inverse CORDIC polymorphic protection, there is a direct relation between the detection rate and the number of stages used in the secondary structure. For a increasing number of stages the detection rate also increases. However, as it can be seen on Table 2, for $n = 9$ the detection rate is actually worse than for $n = 7$. This is because for a 10 stage CORDIC the final angle is so small that the extra stage not only does not improve the detection rate but it rather worsens it. The latter is because the cross-section is increased, and so is the number of errors that the circuit suffers. As can be seen in Fig.6, the z_0 input in the protection mode has the same precision as the first stage of the CORDIC under protection and this precision should be equivalent to a 10th stage. This loss of precision is not enough to reverse the process and makes that with more than 7 stages the absolute number of detected events does not increase and the area in which an error can produce a fault grows, so the final detection rate is worse. Note that with each step data width grows and this affects to the total area, speed and power consumption, oscillating in these implementations between 517 and 959 LUTs (Table 2), reducing frequency from 183MHz to 100MHz and with a difference in power consumption of at most 5mW (Table 3). After analyzing the different error patterns in both protection techniques, it can be claimed that the undetected errors (around 20%) affect the least significant bits of the output, causing a negligible impact in the the output generated. This error is within a range that can be tolerated by the following steps/blocks of the transceivers and metering instruments that use the CORDIC. When more than one error is generated, the spectrum of the output signal changes significantly and the percentage of detection increases.

TABLE 2. Detection rate and area overhead of the different unprotected and protected architectures.

	LUTs	FFs	Detection (%)	Overhead (%)
Unprotected	517	462	N/A	0
Inverse 4 stages	691	464	72.0	33.7/0.5
Inverse 5 stages	736	466	73.2	42.4/0.9
Inverse 6 stages	787	468	76.6	52.2/1.3
Inverse 7 stages	843	470	82.7	63.0/1.7
Inverse 8 stages	893	472	80.8	72.7/2.1
Inverse 9 stages	959	474	81.4	85.4/2.6
Trigonometric [2]	769	503	88.0	49.0/8.0
DMR	1037	925	99.7	100.5/100.2

[2] NOTE: The multipliers are not implemented in DSP blocks to allow more fair comparisons in terms of area.

All the protection techniques presented in this paper occupy less area than the full DMR (which introduces an overhead higher than 105% and 100.2% for LUTs and FFs respectively) at the expense of reducing slightly the error detection rate while keeping the error within a margin (corresponding to the threshold). In the case of the full DMR the detection rate is below 99.7%. The 100% error detection rate is not reached because errors can fall in the input/output routing or other routing errors such as open/bridge faults due to shared multiplexers may interconnect the branches [28].

TABLE 3. Speed and Power consumption of the different unprotected and protected architectures.

	Speed (MHz)	Power (mW)
Unprotected	183	10
Inverse 5 stages	169	15
Inverse 6 stages	146	15
Inverse 7 stages	122	13
Inverse 8 stages	115	13
Inverse 9 stages	100	10
Trigonometric	149	13

Regarding the timing, the trigonometric protection introduces a latency of 2 clock cycles, the inverse polymorphic CORDIC introduces 4 clock cycles in all the six different implementations, and the DMR introduces 1 clock cycle. A summary of speed and power consumption is included in Table 3. It is important to remark that other solutions for correction such as SEM IP are discarded as they require milliseconds to detect errors in the FPGA configuration, while our proposed solutions detect the errors in tens of nanoseconds, being a real-time solution and allowing faster reconfiguration [31].

Inspired by the work of Fujimori and Watanabe in [32] and [33], we used as an upper bound a CORDIC with 40-bit precision, which is used as an address generator for holographic memory systems in radiation environments such as the Fukushima Daiichi nuclear power plant. The conclusions that we obtained after implementing the unprotected and the protected versions of the architecture are similar, with a detection of more than 84% with our proposed method and an area overhead of less than 50% (1249 LUTs for the unprotected version and 1891 LUTs for the protected one).

VI. CONCLUSION

In this paper, two ad-hoc protection techniques have been presented as an alternative to the classic DMR scheme. An ABFT technique based on the trigonometric properties of the CORDIC and a structural-based technique. Regarding the latter, its detection rate depends highly on the number of stages used. It ranges from 72%, for 4 stages to 81.4% for 10 stages, but it provides more precision in the computation of the synthesized signals in absence of error. The area overhead ranges from 33.7/0.5 (% LUTS/FFs) to 85.4/2.6 (% LUTS/FFs). On the other hand, the trigonometric protection has a better trade-off detection rate against area overhead. It exhibits a detection rate of 88% with an overhead of 49/8 (LUTS/FFs %). The latency introduced by the inverse CORDIC protection is equal to 4 clock cycles whereas the latency introduced by the trigonometric

protection is of 2 clock cycles, one to register the outputs of the multipliers and another one to register the output of the adder (see Fig. 4). The figures of merit, area and detection rate, of the trigonometric protection are better than those of the inverse CORDIC protection. However, the inverse CORDIC one can be used as a polymorphic structure with double functionality. In environments with low radiation it can be used as an extended resolution CORDIC and in environments with high radiation it can be split into a lower resolution CORDIC and an inverse CORDIC for error detection purposes.

It can be concluded that all the protection techniques proposed introduce an area overhead lower than that of the DMR and in the case of the inverse polymorphic CORDIC it even adds extra functionality to the CORDIC. Despite the fact that the error detection rate is lower than DMR, high error detection rates can still be obtained. As stated in the Introduction, applications that do not require a 100% protection can benefit from the overhead reduction provided by these techniques, especially if an adaptive approach is followed based on the FPGA reconfiguration capability.

REFERENCES

[1] J. E. Volder, "The CORDIC trigonometric computing technique," *IRE Trans. Electron. Comput.*, vol. EC-8, no. 3, pp. 330–334, Sep. 1959.

[2] J. S. Walther, "A unified algorithm for elementary functions," in *Proc. Spring Joint Comput. Conf. AFIPS (Spring)*, May 1971, pp. 379–385.

[3] P. K. Meher, J. Valls, T.-B. Juang, K. Sridharan, and K. Maharatna, "50 years of CORDIC: Algorithms, architectures, and applications," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 56, no. 9, pp. 1893–1907, Sep. 2009.

[4] P. J. Katkar and Y. S. Angal, "Realization of cordic algorithm in DDS: Novel approch towards digital modulators in MATLAB and VHDL," in *Proc. Int. Conf. Inf. Process. (ICIP)*, Pune, India, Dec. 2015, pp. 355–359.

[5] L. Vachhani, K. Sridharan and P. K. Meher, "A novel CORDIC-based array architecture for the multidimensional discrete Hartley transform," *IEEE Trans. Ind. Electron.*, vol. 56, no. 12, pp. 4915–4929, Dec. 2009.

[6] S. Yu and E. E. Swartzlander, "A scaled DCT architecture with the CORDIC algorithm," *IEEE Trans. Signal Process.*, vol. 50, no. 1, pp. 160–167, Jan. 2002.

[7] J.-I. Guo, C.-M. Liu, and C.-W. Jen, "A novel CORDIC-based array architecture for the multidimensional discrete hartley transform," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 42, no. 5, pp. 349–355, May 1995.

[8] J. R. Cavallaro, M. P. Keleher, R. H. Price, and G. S. Thomas, "VLSI implementation of a CORDIC SVD processor," in *Proc. 8th Univ./Government/Ind. Microelectron. Symp.*, Westborough, MA, USA, 1989, pp. 256–260.

[9] C. Yong Kang and E. E. Swartzlander, "Digit-pipelined direct digital frequency synthesis based on differential CORDIC," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 53, no. 5, pp. 1035–1044, May 2006.

[10] P. Fiala and R. Linhart, "High efficient carrier phase synchronization for SDR using CORDIC implemented on an FPGA," in *Proc. 23rd Telecommun. Forum Telfor (TELFOR)*, Belgrade, Serbia, Nov. 2015, pp. 512–515.

[11] F. Sobhanmanesh, S. Nooshabadi, and K. Kim, "A 212 Mb/s chip for 4 time 4 16-QAM V-BLAST decoder," in *Proc. 50th Midwest Symp. Circuits Syst.*, Montreal, QC, USA, Aug. 2007, pp. 1437–1440.

[12] M. R. Marcin, "Digital receiver phase meter for LISA," *IEEE Trans. Instrum. Meas.*, vol. 54, no. 6, pp. 2446–2453, Dec. 2005.

[13] G. Hechenblaikner, V. Wand, M. Kersten, K. Danzmann, A. García, G. Heinzel, M. Nofrarias, and F. Steier, "Digital laser frequency control and phase-stabilization loops in a high precision space-borne metrology system," *IEEE J. Quantum Electron.*, vol. 47, no. 5, pp. 651–660, May 2011.

[14] J. Johnson, W. Howes, M. Wirthlin, D. L. Mcmurtrey, M. Caffrey, P. Graham, and K. Morgan, "Using duplication with compare for on-line error detection in FPGA-based designs," in *Proc. IEEE Aerosp. Conf.*, Big Sky, MT, USA, Mar. 2008, pp. 1–11.

[15] H. Quinn, Z. Baker, T. Fairbanks, J. L. Tripp, and G. Duran, "Robust duplication with comparison methods in microcontrollers," *IEEE Trans. Nucl. Sci.*, vol. 64, no. 1, pp. 338–345, Jan. 2017.

[16] L. A. Aranda, P. Reviriego, and J. A. Maestro, "A comparison of dual modular redundancy and concurrent error detection in finite impulse response filters implemented in SRAM-based FPGAs through fault injection," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 65, no. 3, pp. 376–380, Mar. 2018.

[17] R. Gonzalez-Toral, P. Reviriego, J. A. Maestro, and Z. Gao, "A scheme to design concurrent error detection techniques for the fast Fourier transform implemented in SRAM-based FPGAs," *IEEE Trans. Comput.*, vol. 67, no. 7, pp. 1039–1045, Jul. 2018.

[18] P. Reviriego, O. Ruano, and J. A. Maestro, "Implementing concurrent error detection in Infinite-Impulse-Response filters," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 59, no. 9, pp. 583–586, Sep. 2012.

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

[20] M. M. Nisar and A. Chatterjee, "Guided probabilistic checksums for error control in low power digital-filters," in *Proc. 14th IEEE Int. On-Line Test. Symp.*, Jul. 2008, pp. 239–244.

[21] H.-G.-D. Stratigopoulos and Y. Makris, "Concurrent detection of erroneous responses in linear analog circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 25, no. 5, pp. 878–891, May 2006.

[22] S. Mandal, R. Paul, S. Sau, A. Chakrabarti, and S. Chattopadhyay, "A novel method for soft error mitigation in FPGA using modified matrix code," *IEEE Embedded Syst. Lett.*, vol. 8, no. 4, pp. 65–68, Dec. 2016.

[23] S. Mandal, R. Paul, S. Sau, A. Chakrabarti, and S. Chattopadhyay, "Efficient dynamic priority based soft error mitigation techniques for configuration memory of FPGA hardware," *Microprocessors Microsyst.*, vol. 51, pp. 313–330, Jun. 2017.

[24] H. Asadi and M. B. Tahoori, "Soft error modeling and remediation techniques in ASIC designs," *Microelectron. J.*, vol. 41, no. 8, pp. 506–522, Aug. 2010.

[25] K. Morgan, M. Caffrey, P. Graham, E. Johnson, B. Pratt, and M. Wirthlin, "SEU-induced persistent error propagation in FPGAs," *IEEE Trans. Nucl. Sci.*, vol. 52, no. 6, pp. 2438–2445, Dec. 2005.

[26] G. Asadi and M. B. Tahoori, "Soft error rate estimation and mitigation for SRAM-based FPGAs," in *Proc. ACM/SIGDA 13th Int. Symp. Field-Programmable Gate Arrays (FPGA)*. New York, NY, USA: ACM, 2005, pp. 149–160.

[27] M. Wirthlin, D. Lee, G. Swift, and H. Quinn, "A method and case study on identifying physically adjacent multiple-cell upsets using 28-nm, interleaved and SECDED-protected arrays," *IEEE Trans. Nucl. Sci.*, vol. 61, no. 6, pp. 3080–3087, Dec. 2014.

[28] L. Sterpone and M. Violante, "Analysis of the robustness of the TMR architecture in SRAM-based FPGAs," *IEEE Trans. Nucl. Sci.*, vol. 52, no. 5, pp. 1545–1549, Oct. 2005.

[29] *Soft Error Mitigation Controller Logicore IP Product Guide (PG036)*. San Jose, CA, USA: Xilinx, Sep. 2015.

[30] L. A. Aranda, A. Sánchez-Macián, and J. A. Maestro, "ACME: A tool to improve configuration memory fault injection in SRAM-based FPGAs," *IEEE Access*, vol. 7, pp. 128153–128161, 2019.

[31] *Soft Error Mitigation Controller v4.1*. Xilinx Appl. Notes PG036. LogiCORE IP Product Guide, Apr. 2018.

[32] T. Fujimori and M. Watanabe, "A 400 MRAD radiation-hardened optoelectronic embedded system with a silver-halide holographic memory," in *Proc. NASA/ESA Conf. Adapt. Hardw. Syst. (AHS)*, Edinburgh, Scotland, Aug. 2018, pp. 218–224.

[33] T. Fujimori and M. Watanabe, "Holographic memory calculation FPGA accelerator for optically reconfigurable gate arrays," in *Proc. IEEE 15th Int. Conf Dependable, Autonomic Secure Comput., 15th Int. Conf. Pervas. Intell. Comput., 3rd Int. Conf Big Data Intell. Comput. Cyber Sci. Technol. Congr. (DASC/PiCom/DataCom/CyberSciTech)*, Orlando, FL, USA, Nov. 2017, pp. 620–625.

LUIS ALBERTO ARANDA received the B.Sc. degree in industrial engineering and the M.Sc. degree in robotics from the Universidad Carlos III de Madrid, Spain, in 2012 and 2015, respectively, and the Ph.D. degree (Hons.) in industrial engineering from the Universidad Antonio de Nebrija, Madrid, Spain, in 2018.

He worked as a Project Engineer with Zeus Creative Technologies, S.L., developing various computer vision projects, from 2013 to 2014. He was responsible for both hardware and software design and implementation. He is currently with the ARIES Research Center, Universidad Antonio de Nebrija. He is the author of several technical publications, both in journals and international conferences. His research interests include reconfigurable computing for space applications, computer vision, and robotics.

FRANCISCO GARCIA-HERRERO received the B.Sc. degree in telecommunication engineering from the Escuela Politecnica Superior de Gandia, Spain, in 2008, and the M.S. and Ph.D. degrees in electrical engineering from the Universitat Politècnica de València, Spain, in 2010 and 2013, respectively.

He has worked as a Lecturer and a Researcher at several universities, including the European University Miguel de Cervantes and the Universitat Politècnica de València. He is currently an Associate Professor and a Researcher with the Universidad Antonio de Nebrija. His research interests include hardware and algorithmic optimizations of error-control decoders and fault-tolerance electronics in communication and storage systems.

LUIS ESTEBAN received the B.Sc. degree in electronics engineering from the Universidad de Zaragoza, Spain, in 2003, the M.Sc. degree in electronics engineering from the Universidad Complutense de Madrid, Spain, in 2005, and the Ph.D. degree in electronics from the Universidad Politécnica de Madrid, Spain, in 2011.

He has worked as a Lecturer and Researcher at several universities and research centers such as the University of Liverpool, U.K., the University of California at Berkeley, USA, the Universidad Politécnica de Madrid, Spain, Ciemat, Spain, and CSIC, Spain. His research interest is focused on the implementation of digital signal processing and artificial intelligence algorithms in ASICs and FPGAs for space and scientific applications. Furthermore, the evaluation of the implementation issues in digital- and mixed-signal systems and the use of rapid prototyping techniques are also the fields that he is interested in pursuing.

ALFONSO SÁNCHEZ-MACIÁN received the M.Sc. and Ph.D. degrees in telecommunications engineering from the Universidad Politécnica de Madrid, Madrid, Spain, in 2000 and 2007, respectively.

He has worked as a Lecturer and a Researcher at several universities such as the Universidad Politécnica de Madrid, the IT Innovation Centre, University of Southampton, Southampton, U.K., and the Universidad Antonio de Nebrija, Madrid, where he is currently a part of the ARIES Research Center. He previously worked in numerous national and multinational companies as a project manager and senior consultant for IT projects. His current research interests include fault-tolerance and reliability, the performance evaluation of communication networks, and knowledge representation and reasoning in distributed systems.

JUAN ANTONIO MAESTRO (Senior Member, IEEE) received the M.Sc. degree in physics and the Ph.D. degree in computer engineering from the Universidad Complutense de Madrid, Madrid, Spain, in 1994 and 1999, respectively.

He has been the Director of the Electronic Design and Space Technology Research Group, Universidad Antonio de Nebrija, Madrid, since 2004, where he has also recently founded the ARIES Research Center (www.nebrija.es/aries), devoted to the aerospace research and innovation in electronic systems. His current activities are oriented to the space industry, with several projects on the protection of digital circuits against the effects of radiation, including microprocessors, memories, and auxiliary systems. He also collaborates with institutions such as the European Space Agency, Stanford University, University College Dublin, or the Harbin Institute of Technology, among others. He is the author of numerous technical publications in journals and international conferences. His areas of interest include computer architecture, digital design, fault-tolerance, reliability, small satellites, and space applications.