AXI Lite Redundant On-Chip Bus Interconnect for High Reliability Systems

Jesús Lázaro, Armando Astarloa, Aitzol Zuloaga, José Ángel Araujo, and Jaime Jiménez

***Abstract*—Nowadays, system-on-chips have become critical since they support more and more safe applications due to their flexibility. However, they are susceptible to single-event upsets because the memory cell size has significantly shrunk. This article presents a triple redundant on-chip interconnect bus that provides low-speed peripherals with high reliability. In addition to correcting single errors and detecting duplicated ones, the proposed circuit offers zero latency and is transparent for both the embedded processor and the peripherals. These characteristics make it suitable for hard real-time applications. At the same time, the impact on area and power consumption is minimal.**

***Index Terms*—AXI, FPGA, redundancy, SoC.**

I. Introduction

NOWADAYS, static random access memory (SRAM) field programmable gate arrays (FPGAs) are widely used in many applications, such as autonomous vehicles, signal processing, industry 4.0, and embedded applications [1], thanks to their growth in capabilities.

Due to the nature of SRAM-based FPGAs, they are susceptible to single-event upset (SEU) induced by high-energy particles [2], [3], which limits their usage in safety and mission-critical applications. In SRAM-based FPGAs, the programmability is controlled by the SRAM cell-based configuration memory. With every advance in reducing voltage and dimensions of the SRAM cell, its capacitance is decreased. This decrement increases the vulnerability of the cell to particles of lower energy. When an event occurs, the actual behavior of the circuit may get altered until the FPGA is programmed again [4].

The most common SEU hardening techniques include triple modular redundancy (TMR) [5], partial TMR [6], and dynamic partial reconfiguration [7]. The TMR implementation uses three identical logic blocks performing the same task in parallel and compares the outputs by a majority voter. In safety-critical applications, TMR techniques are essential.

TMR may be applied with different granularities. The finest granularity of TMR is achieved when voting is applied for all registers in the design [8]. A more coarse granularity can be achieved if triplication and voting are applied through the device based on larger modules and their respective outputs [5]. The robustness increases when voting is performed over smaller modules, but so does the implementation cost in terms of FPGA utilization. Furthermore, a fine granularity reduces the highest achievable system speed because voters are added to the critical path [9].

In this article, the reliability of FPGA systems is increased by introducing coarse-grained TMR in AXI4-Lite on-chip peripherals. In addition, intellectual property (IP) core—AXILiteRedundant—has a minimal area, power impact, and maximum speed. At the same time, the IP core reduces the processor's overhead by presenting a single peripheral structure rather than the triple system produced by the TMR. This characteristic allows more straightforward software to process the incoming data, allowing hard real-time operations.

The scope of this research project includes implementing the AXILiteRedundant architecture in a ready-to-use HDL-described IP. It will be packaged in the format specified by the Silicon vendors [10], allowing the engineers to integrate this block into the system-on-chip (SoC) design seamlessly. The implementation includes the basic software driver to configure and manage the IP.

Redundancy can be exploited through an on-chip building block in SoC designs for critical systems, where the failure rate shall be maintained into shallow values. This IP enhances the reliability of the user IP that wraps. An example of a typical user IP that can be rugged using this wrap is an IP that implements the standard communication protocol for satellite platforms, SpaceWire [11]. This wrap and the redundant implementation for a given IP shall be combined with other mechanisms, such as lock-step CPU operation or SRAM memory scrubbing, to build a complete and robust SoC solution. The primary sector that benefits from these innovations is Space [12]. However, the demand for robust SoC platforms is increasing in other sectors, such as transportation, defense, and medical.

The rest of the article is organized as follows. Section II presents work done previously in the field. Section III describes the proposed solution. Section IV describes the obtained

Manuscript received 29 September 2022; revised 25 January 2023; accepted 12 April 2023. This work was supported in part by the Basque University System IT1440-22, in part by the Department of Education and, within PILAR ZE-2020/00022 and COMMUTE ZE-2021/00931 projects, in part by the Hazitek program, both of the Basque Government; in part by the latter also by the Ministerio de Ciencia e Innovación of Spain through the Centro para el Desarrollo Tecnológico Industrial (CDTI) within the project IDI-20201264 and IDI-20220543, and in part by the Fondo Europeo de Desarrollo Regional 2014-2020 (FEDER funds) . Associate Editor: A. Hamou-Lhadj. *(Corresponding author: Jesús Lázaro.)*

The authors are with the University of the Basque Country, 48940 Bilbao, Spain (e-mail: jesus.lazaro@ehu.eus; armando.astarloa@ehu.eus; aitzol.zuloaga@ehu.eus; joseangel.araujo@ehu.eus; jaime.jimenez@ehu.eus).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TR.2023.3267436.

Digital Object Identifier 10.1109/TR.2023.3267436

results. Section V compares it with some similar ones. Finally, Section VI concludes the article.

II. Previous Work

Redundancy in SoC can take multiple approaches and exists at different levels. The most common ones are: a) system architecture, b) microprocessor, c) communications, d) input/output (I/O), and e) FPGA.

System architecture level redundancy is a common approach in SoC [2], [13], [14]. Key areas where it is used include flight control, industry, and satellite design. The main approaches are component redundancy with TMR and redundant intrachip communication buses.

Microprocessor level redundancy [15], [16], [17] uses multiple processors inside the same SoC. The most common approach is lockstep. In other words, several processors run the same program with the same input data. Newer approaches include using heterogeneous SoCs —systems with more than one type of processor— to mitigate potential hidden errors in the processor's design.

Communications level redundancy [18], [19] uses multiple communication channels to connect to external elements.

I/O level redundancy [20], [21], [22] is related to the electrical level redundancy of the signals. Common approaches are routing the same signal to multiple inputs or joining the signals from multiple outputs with some circuitry to avoid short circuits.

FPGA-level redundancy engulfs a variety of issues. FPGAs are incredibly flexible, which allows internal redundancy [23], [24]. They also enable some previous redundancy schemes [15].

At the same time, FPGAs suffer from specific issues such as SEUs due to numerous SRAM cells. In order to mitigate this issue, several approaches, such as scrubbing, have been proposed [25], [26], [27].

This article presents a new approach to enhancing reliability in SoC. It is based on including redundancy of internal buses for low-speed peripherals. The literature has several examples in this area of research.

Bertozzi et al. [23] proposed a redundant bus coding to increase redundancy in power-constrained systems. They tested known algorithms such as Hamming and cyclic redundancy check (CRC). One interesting point is their focus on power, which leads to the conclusion that retransmission is, in many cases, more efficient. However, this approach is not valid for time-critical systems. In addition, it is not transparent to the processor, requiring extensive recoding.

Benevenuti et al. [24] deal with the redundancy of advanced eXtensible interface (AXI) stream interfaces. Their solution includes using redundant bus inputs to every IP Core. Each IP Core can efficiently decide the correct one by knowing the kind of information it is receiving. The main problem of this approach is that it requires recoding of the IP Core to have triple internal redundancy.

III. Solution Description

Since redundancy is an overall requirement, it must be addressed at the IP and system levels. Therefore, our solution is divided into two main elements: a) `AXILiteRedundant` IP Core, and b) redundant system.

A. AXILiteRedundant

This IP core is in charge of triplicating elements from the slave interface (IF) to the master IFs. It also decides the values of the signals that go from the multiple masters IF to the slave IF. To do so, it will vote for the correct value. Fig. 1 shows the diagram of the IP Core. The slave IF is the one connected (directly or indirectly) to the processor. The master IF is the one connected to the external world. Since we are dealing with redundancy, there are three master IF.

The core manages five channels per interface, as the AXI specification proposes. The following modifications to the AXI standard have been applied to implement the redundancy mechanism.

1) *Read Address:* This bus is triplicated to all the master IF. Return ready signals are majority voted.
2) *Read Data:* Ready signal is triplicated to all masters. Data and valid are majority voted.
3) *Write Address:* This bus is triplicated to all the master IF. Return ready signals are majority voted.
4) *Write Data:* This bus is triplicated to all masters. Return ready signals are majority voted.
5) *Write Response:* Ready signal is triplicated to all masters. The response is majority voted.

The voting scheme uses a three-way majority voting [28]. This circuit outputs the most abundant value. In case of all three values are different, the circuit also outputs an error signal. The process of triplicating and voting is combinational. This ensures 0 latency in clock cycles. Since the process is highly optimized, it is also very efficient and adds a minor frequency penalty, as shown in the result section. Fig. 2 shows the Karnaugh table of the circuit.

The IP core is also in charge of feeding some information to the processor. It will state if all the masters are providing the same values. If not, and can be corrected (two masters provide the same information), it will do so and inform the error. If it cannot be corrected (errors in multiple bits of a bus but in different masters), it will inform the error, and the output value is irrelevant. This process is done independently for every channel in the interface.

B. System

The system is composed of several IP cores. In our example, apart from the minimal system required to implement a bare SoC device, we have added a firewall IP core. The main elements of the system are (see Fig. 3) as follows:

1) `Zynq`: It is the processor present in the system [29]. In our example is in charge of accessing the peripherals. The processor has a single peripheral from the software point of view, even if they are triplicated.
2) `AXiLiteRedundant`: It is in charge of triplicating the incoming AXI transactions so that the processor sees a single peripheral.
3) `Firewall`: A bridge between two portions of an AXI memory-mapped network protects one portion from issues

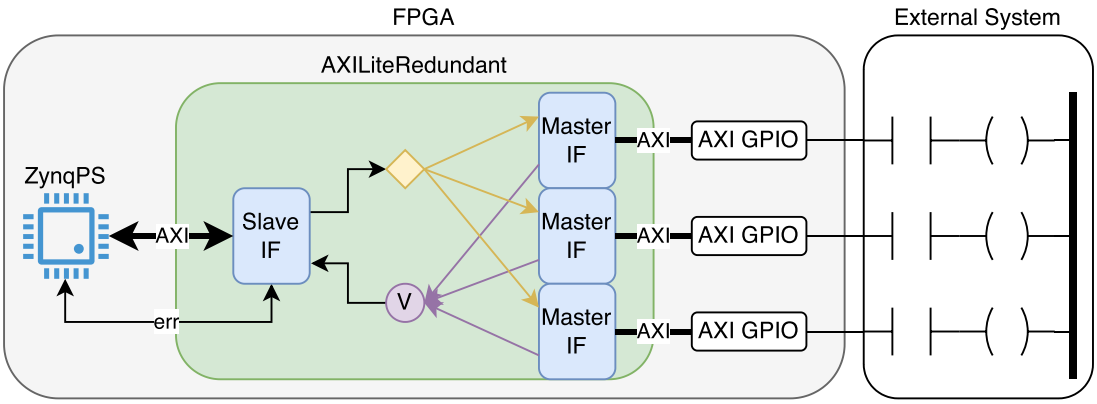

Fig. 1. Simplified diagram of the IP core. Slave to master is triplicated. Master to slave is majority-voted. Error backchannel (*err*) is also depicted. It will inform the processor of any issues.

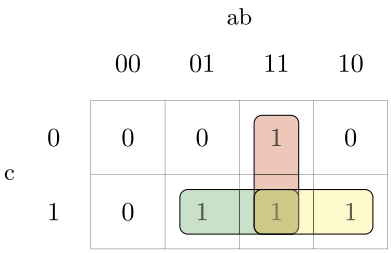

Fig. 2. Tree-way majority voting Karnaugh table.

caused by the opposite portion, such as protocol violations or timeout hangs [30]. In our case, it has been added as an extra protection for a case, where one of the cores has been compromised and is not working correctly. There are other ways of implementing this extra security, for example, using a `Firewall` per peripheral core.

4) `Interconnect`: It is the block responsible for connecting several AXI networks. Even if only a single slave is connected to the Zynq processor, this block is required because of the different natures of AXI on both sides. In this case, AXILite downstream and AXI 3 upstream [31].
5) `GPIO`: They are the peripheral blocks [32]. In order to have redundancy, each I/O is connected so that there is a physical majority vote [33].

IV. Results

In order to verify the system, we used a simulation environment. This simplified system allows for verifying the correctness of the IP Core and can generate errors to test the resilience of the whole architecture. The simulation system is depicted in Fig. 4.

The key elements, apart from those described in the previous section, are as follows:

1) `AXI Traffic Generator`: Generates traffic over the AXI4. It generates a wide variety of AXI transactions based on the core programming and selected mode of operation [34].
2) `AXI Verification IP`: Checks that all transactions comply with the AXI standard. It is useful when creating new IP to guarantee that it will not cause any issues on the communication channel [35].

The result of the simulation is depicted in Fig. 5. The I/O has been simulated in reading and writing, including high impedance. The process that can be seen is:

1) W_0 : Configuration of the I/O as output—write "0."
2) W_1 : Write all "1" to the outputs.
3) W_0 : Configuration of the I/O as input—write "1."
4) R_0 : Read the input.

The read input process is done four times with different results:

1) All inputs have the same value, so there is no error.
2) GPIO 3 has a different value, so there is a recoverable error.
3) GPIO 1 has a different value, so there is a recoverable error.
4) All GPIO have different values, so there is an unrecoverable error.

In addition to increasing reliability, the IP design has focused on minimizing FPGA resources and energy consumption. Moreover, simultaneously, without introducing a significative penalty on timing results. Table I shows the area and energy results. As can be seen, the area required to implement `AXILiteRedundant` is $0.42\,\%$ of the look-up tables (LUTs) present in the `xc7z020` of the Zedboard. From the energy results, it is clear that energy consumption is low compared to the rest of the IPs in the system. For example, the IP core requires a fourth of the energy of the `GPIO` core.

From the timing point of view, as stated before, the IP is designed to introduce 0 clock latency. This simplifies the design and allows it to be used in real-time systems. From the frequency point of view, the overall system can work at speeds over 166 MHz. This shows that the overall performance is not affected by the IP core.

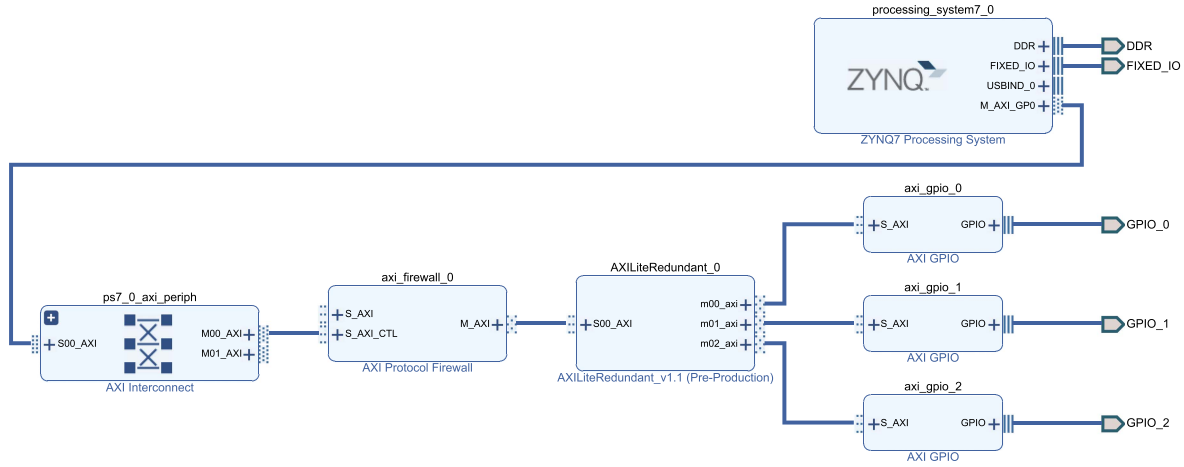

Fig. 3. Overall system. The PS is connected to the redundant system through an AXI Interconnect and the proposed core.

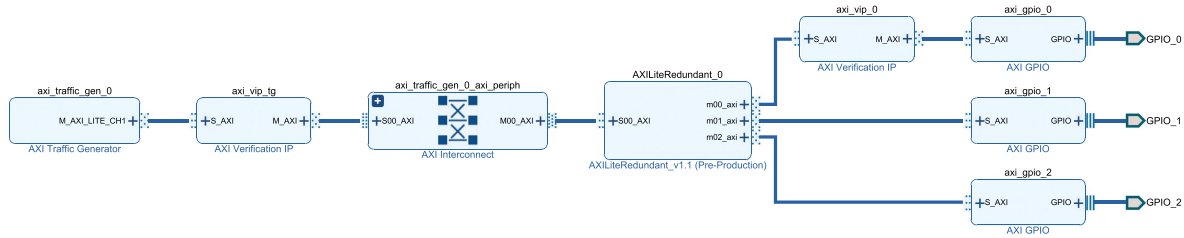

Fig. 4. Simulation system. A traffic generator replaces the processor system. A couple of AXI verification cores are added to check for erroneous transactions. The AXI infrastructure is simplified compared to Fig. 3 to more easily check the correct functioning of the system.

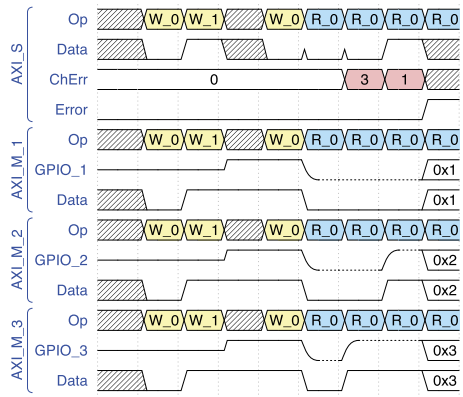

Fig. 5. Simulation result. Initially, W_0 and W_1 configure the general purpose input/output (GPIO) as output and write the value "1." Next, the GPIO is configured as an input by writing into W_0, and four consecutive reads (R_0) are performed. In the first one, all three cores respond with the same value. Thus, there is no error. In the second read, the third GPIO is erroneous. In the third read, the first GPIO is wrong, while in the last one, all three values are different, leading to an unrecoverable error ($\texttt{Error} =$ "1").

TABLE I
RESOURCE UTILIZATION OF DIFFERENT CORES IN THE SYSTEM.

Name	Slice LUTs	Slice Registers	LUTs % of total	Power mW
axi_firewall_0	626	810	1.18	4.35
ps7_0_axi_periph	543	692	1.02	6.66
AXILiteRedundant	223	81	0.42	0.48
axi_gpio_2	131	382	0.25	1.65
axi_gpio_1	131	382	0.25	1.66
axi_gpio_0	131	382	0.25	1.74
axi_gpio_result	96	318	0.18	2.90
processing_system7_0	24	0	0.05	1530.38
rst_ps7_0_100M	19	40	0.04	0.23

The proposed core requires less than 0.42% of xc7z020 present in the Zedboard. The power results are those provided by Vivado postimplementation.

V. COMPARISON

Due to the specific nature of the IP, it is not easy to compare it with other IPs. Other approaches are specific to other interface definitions and, thus, difficult to compare with the proposed IP.

Benevenuti et al. [24] proposed a redundant system for a single processing core using AXI Stream. The main difference is that the system has a single peripheral with a triplicated input interface and a single output interface. The input voter is included inside the IP core, thus requiring redesigning the IP. The input data comes from three different direct memory accesses (DMAs) that transfer the information from the processor to the peripheral. The output is transferred to the direct memory accesss (DMAs) that copy the data to the three different memory sections where the processor can get it. The processor must be aware of the redundancy to vote the results. The article does not provide any area, power, or timing information.

Bertozzi et al. [23] proposed a resilient IF not by redundancy but by including parity bits in the bus. This system is transparent

TABLE II
COMPARISON OF THE PROPOSED PAPER WITH OTHERS PRESENT IN THE LITERATURE.

Paper	Red.	CPU indep.	IP indep.	Area (gates)	Energy (mW)	Timing (ns)
[24]	IF	✗	✗	—	—	—
[23]	IF	✓	✓	20,826	0.62	4.85
This Paper	IF + IP	✓	✓	4376	0.48	2.50

The table shows the redundant elements, whether the CPU and the IP must be aware of the redundancy to function, and the impact on the area, energy, and timing.

for the processor and the peripheral. This approach does not use redundant peripherals and can only cope with errors in the bus and internal to the FPGA. The article provides comprehensive power information depending on the redundancy algorithm (hamming, CRC, etc.). The proposed algorithms require very little power but may have high latency. The latency is $0.88\,\text{ns}$ to $4.56\,\text{ns}$ and is different for encoding and decoding. Area usage is 2.7×10^3 gates to 11.0×10^3 gates. The energy consumption is $1.2\,\mu\text{W}$ to $617\,\mu\text{W}$.

Table II shows the comparison between the proposed system and the other ones. The proposed system is the most efficient in terms of area and has the lowest latency. In terms of energy, it is better than the most power-hungry version provided by Bertozzi et al., although it requires more energy than the most basic ones presented in the article. Furthermore, it is the sole one that can handle both IF and peripheral errors. It also possesses critical characteristics, such as using unmodified IP cores and software.

VI. CONCLUSION

This article presents aTMR voter for the AXI4-Lite Standard. As FPGA systems are more and more usual in safe applications, at the same time that susceptible to SEUs, they must be hardened using redundancy. Our core protects the interconnection as well as the peripheral IPs. In addition to correcting single errors and detecting duplicated ones, the resulting IP requires very few FPGA resources for the implementation. The increment in power consumption is negligible, and it does not impact the latency due to the combinational nature of the IP core. This proposal is transparent both for the peripheral IPs and the CPU, allowing the usage of standard peripheral IPs and eliminating any overhead to the processing software. Further work in this area includes extending the IP range to other standard interfaces, such as AXI Stream. A redundant AXI Stream IP would be useful for systems that require high bandwidth and low latency for data processing and for multichip systems.

REFERENCES

[1] G. Lentaris, I. Stratakos, I. Stamoulias, D. Soudris, M. Lourakis, and X. Zabulis, "High-performance vision-based navigation on SoC FPGA for spacecraft proximity operations," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 4, pp. 1188–1202, Apr. 2020, doi: 10.1109/TCSVT.2019.2900802.

[2] X. Zhang and X. Zhao, "Architecture design of distributed redundant flight control computer based on time-triggered buses for UAVs," *IEEE Sensors J.*, vol. 21, no. 3, pp. 3944–3954, Feb. 2021, doi: 10.1109/JSEN.2020.3025551.

[3] Q. Shao, S. Yang, and X. Gou, "Formal analysis of multiple-cell upset failure based on common cause failure theory," *IEEE Trans. Rel.*, vol. 70, no. 4, pp. 1495–1509, Dec. 2021, doi: 10.1109/TR.2020.3010937.

[4] A. Stoddard, A. Gruwell, P. Zabriskie, and M. J. Wirthlin, "A hybrid approach to FPGA configuration scrubbing," *IEEE Trans. Nucl. Sci.*, vol. 64, no. 1, pp. 497–503, Jan. 2017, doi: 10.1109/TNS.2016.2636666.

[5] L. A. C. Benites et al., "Reliability calculation with respect to functional failures induced by radiation in TMR arm cortex-m0 soft-core embedded into SRAM-based FPGA," *IEEE Trans. Nucl. Sci.*, vol. 66, no. 7, pp. 1433–1440, Jul. 2019, doi: 10.1109/TNS.2019.2921796.

[6] A. J. Sanchez-Clemente, L. Entrena, and M. Garcia-Valderas, "Partial TMR in FPGAs using approximate logic circuits," *IEEE Trans. Nucl. Sci.*, vol. 63, no. 4, pp. 2233–2240, Aug. 2016, doi: 10.1109/TNS.2016.2541700.

[7] X. Wei, Y. Z. Xie, Y. Xie, and H. Chen, "Dynamic partial reconfiguration scheme for fault-tolerant FFT processor based on FPGA," *J. Eng.*, vol. 2019, no. 21, pp. 7424–7427, Oct. 2019, doi: 10.1049/joe.2019.0353.

[8] Xilinx, "Triple module redundancy design techniques for virtex FPGAs - XAPP197," Xilinx, Application Note, Jul. 2006. [Online]. Available: https://docs.xilinx.com/v/u/en-US/xapp197

[9] U. Kretzschmar, A. Astarloa, J. Lazaro, M. Garay, and J. D. Ser, "Robustness of different TMR granularities in shared wishbone architectures on SRAM FPGA," in *Proc. IEEE Int. Conf. Reconfigurable Comput. FPGAs*, 2012, pp. 1–6, doi: 10.1109/ReConFig.2012.6416785.

[10] AMD-Xilinx Inc., "Vivado design suite user guide: Creating and packaging custom IP," 2022. [Online]. Available: https://docs.xilinx.com/r/en-US/ug1118-vivado-creating-packaging-custom-ip

[11] System-on-Chip Engineering S.L., "SpaceWire IP," 2023. [Online]. Available: https://soc-e.com/spacewire-ip-core/

[12] F. Siegle, T. Vladimirova, J. Ilstad, and O. Emam, "Mitigation of radiation effects in SRAM-based FPGAs for space applications," *ACM Comput. Surv.*, vol. 47, no. 2, pp. 1–34, Jan. 2015, doi: 10.1145/2671181.

[13] M. S. Farias, N. Nedjah, and P. V. R. de Carvalho, "Active redundant hardware architecture for increased reliability in FPGA-based nuclear reactors critical systems," *Microprocessors Microsystems*, vol. 90, Apr. 2022, Art. no. 104495, doi: 10.1016/j.micpro.2022.104495.

[14] K. Kazi et al., "Design and validation architecture of the dream chaser fault tolerant flight computer," in *Proc. IEEE Space Comput. Conf.*, 2019, pp. 67–76, doi: 10.1109/SpaceComp.2019.00013.

[15] C. Rodrigues, I. Marques, S. Pinto, T. Gomes, and A. Tavares, "Towards a heterogeneous fault-tolerance architecture based on arm and RISC-v processors," in *Proc. IEEE 45th Annu. Conf. Ind. Electron. Soc.*, 2019, pp. 3112–3117, doi: 10.1109/IECON.2019.8926844.

[16] J. Li, S. Zhang, and C. Bao, "DuckCore: A fault-tolerant processor core architecture based on the RISC-V ISA," *Electronics*, vol. 11, no. 1, Dec. 2021, Art. no. 122, doi: 10.3390/electronics11010122.

[17] H. Su, T. Lu, C. Feng, and L. Chen, "Triple module redundancy reliability framework design based on heterogeneous multicore processor," *Procedia Comput. Sci.*, vol. 183, pp. 504–511, 2021, doi: 10.1016/j.procs.2021.02.090.

[18] S. Hiergeist and G. Seifert, "Implementation of a SPI based redundancy network for SoC based UAV FCCs and achieving synchronization," in *Proc. IEEE/AIAA 37th Digit. Avionics Syst. Conf.*, 2018, pp. 1–10, doi: 10.1109/DASC.2018.8569858.

[19] A. Astarloa, J. Lazaro, U. Bidarte, A. Zuloaga, and M. Idirin, "System-on-chip implementation of reliable ethernet networks nodes," in *Proc. IEEE 39th Annu. Conf. Ind. Electron. Soc.*, 2013, pp. 2329–2334, doi: 10.1109/IECON.2013.6699494.

[20] W. R. Tonti, J. A. Mandelman, A. R. Bonaccio, C. L. Bertin, H. L. Kalter, and J. A. Fifield, "Redundant input/output driver circuit," U.S. Patent US6 177 809B1, Jan. 23, 2001. [Online]. Available: https://patents.google.com/patent/US6177809B1/en

[21] Y. Li, J. Xue, and Q. Gai, "Design for input and output card of triple redundant control system," in *Proc. IEEE Symp. Comput. Appl. Commun.*, 2014, pp. 102–104, doi: 10.1109/SCAC.2014.28.

[22] A. Astarloa, J. Lázaro, U. Bidarte, A. Zuloaga, and J. L. Martín, "An autonomous fault tolerant system for CAN communications," in *Trends in Applied Intelligent Systems*. Berlin, Germany: Springer, 2010, pp. 281–290, doi: 10.1007/978-3-642-13033-5_29.

[23] D. Bertozzi, L. Benini, and G. D. Micheli, "Error control schemes for on-chip communication links: The energy-reliability tradeoff," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 24, no. 6, pp. 818–831, Jun. 2005, doi: 10.1109/TCAD.2005.847907.

[24] F. Benevenuti and F. L. Kastensmidt, "Analyzing AXI streaming interface for hardware acceleration in AP-SoC under soft errors," in *Applied Reconfigurable Computing. Architectures, Tools, and Applications*. Berlin, Germany: Springer, 2018, pp. 243–254, doi: 10.1007/978-3-319-78890-6_20.

[25] L. Gantel, Q. Berthet, E. Amri, A. Karlov, and A. Upegui, "Fault-tolerant FPGA-based nanosatellite balancing high-performance and safety for cryptography application," *Electronics*, vol. 10, no. 17, Sep. 2021, Art. no. 2148, doi: 10.3390/electronics10172148.
[26] C. D. Sio, S. Azimi, and L. Sterpone, "On the analysis of radiation-induced failures in the AXI interconnect module," *Microelectronics Rel.*, vol. 114, Nov. 2020, Art. no. 113733, doi: 10.1016/j.microrel.2020.113733.
[27] J. Borcsok, W. Muller, E. Hahn, M. Schwarz, and M. Abdelawwad, "Safe-system-on-chip for functional safety," in *Proc. IEEE 18th Int. Multi-Conf. Syst., Signals Devices*, 2021, pp. 619–624, doi: 10.1109/SSD52085.2021.9429321.
[28] B. Parhami, "Voting algorithms," *IEEE Trans. Rel.*, vol. 43, no. 4, pp. 617–629, Dec. 1994, doi: 10.1109/24.370218.
[29] Xilinx, "Processing system 7-PG082," Xilinx, Product Guide, May 2017. [Online]. Available: https://docs.xilinx.com/v/u/en-US/pg082-processing-system7
[30] Xilinx, "AXI protocol firewall IP - PG293," Xilinx, Product Guide, Feb. 2022. [Online]. Available: https://docs.xilinx.com/v/u/en-US/pg293-axi-firewall
[31] Xilinx, "AXI interconnect - PG059," Xilinx, Product Guide, May 2022. [Online]. Available: https://docs.xilinx.com/r/en-US/pg059-axi-interconnect
[32] Xilinx, "AXI GPIO - PG144," Xilinx, Product Guide, Oct. 2016. [Online]. Available: https://docs.xilinx.com/v/u/en-US/pg144-axi-gpio
[33] A. Mukherjee and A. S. Dhar, "Triple transistor based triple modular redundancy with embedded voter circuit," *Microelectronics J.*, vol. 87, pp. 101–109, May 2019, doi: 10.1016/j.mejo.2019.03.014.
[34] Xilinx, "AXI traffic generator—PG125," Xilinx, Product Guide, Feb. 2019. [Online]. Available: https://xilinx.com/content/dam/xilinx/support/documents/ip_documentation/axi_traffic_gen/v3_0/pg125-axi-traffic-gen.pdf
[35] Xilinx, "AXI verification IP—PG267," Xilinx, Product Guide, Dec. 2021. [Online]. Available: https://docs.xilinx.com/r/en-US/pg267-axi-vip

Jesús Lázaro revieved the M.Sc. and Ph.D. degrees in telecommunication engineering from the University of the Basque Country (UPV/EHU), Spain, in 2001 and 2005, respectively. He is a Full Professor. In 2010, he was a Research Visitor with the Configurable Computing Lab of Virginia Tech. He is a member of the Applied Electronics Research Team and co-founder and entrepreneur of System-on-Chip engineering S.L. (soc-e.com). He has participated in 48 competitive research projects supported by public institutions and in 49 private research contracts, in 21 of them as the leading Researcher. He is author or coauthor of four patents, 38 articles in international scientific magazines, and 76 papers in international conferences. His main areas of research are high-speed circuits based on reconfigurable devices and communications devices.

Armando Astarloa received the M.Sc. and Ph.D. degrees in electrical engineering from the University of the Basque Country (UPV/EHU), Spain, in 1999 and 2005, respectively.

After several years of developing his professional career as a Hardware Engineer in the private sector, in 2001 he joined the staff of the UPV/EHU in the Faculty of Engineering of Bilbao as a full-time Researcher and Lecturer. He is a founding member of the Research Applied Electronics Research Team, and his research topics include reliable electronic design for reconfigurable devices, industrial networking, and cybersecurity. He is the author of more than 50 scientific journals, 120 contributions to conferences, book chapters, patents, and he collaborates as a committee member in IEC working groups. In 2008, he enrolled as a visiting Researcher in the Institute of Microelectronics and Wireless Systems in Ireland. In 2010, he launched System-on-Chip engineering S.L. (SoC-e.com) business project, and he held the CEO position at this company from 2016 to 2020. In 2021, he achieved the Full Professor position at the UPV/EHU.

Aitzol Zuloaga received the M.Sc. degree in electronics engineering from the Simon Bolivar University, Venezuela, in 1985 and the Ph.D. degree in telecommunications engineering from the University of the Basque Country, Spain, in 2001.

From 1985 to 1995, he worked as a Project Manager in R&D Departments of digital telephony related companies. In 1995, he entered in the University of the Basque Country as Researcher, in 2000 he became Teaching Assistant and, in 2002 he became Associate Professor of electronics and telecommunications engineering of the School of Engineering of the University of the Basque Country. He is a member of the "Applied Electronics Research Team" and he has participated in more than 40 research projects supported by public institutions and private companies, in five of them as main Researcher. He is author or coauthor of more than 20 papers in international magazines and 60 communications in national and international conferences. He has been the Director of four Ph.D. thesis.

José Ángel Araujo received the M.S. degree in telecommunications engineering and advanced electronics systems, and the doctor degree from the University of the Basque Country (UPV/EHU), Bilbao, Spain, in 2000, 2012, and 2015, respectively.

From 2000 to 2005, he was in the engineering and production industry. In 2006, he joined UPV/EHU, as a Researcher and as a Lecturer and he is in the Electronics Technology Department. His areas of research are reconfigurable devices and communications devices.

Jaime Jiménez received M.S. and Ph.D. degrees in telecommunication engineering from the University of the Basque Country (UPV/EHU), Spain, in 1991 and 2005, respectively.

He has participated in 45 competitive research projects supported by public institutions and 39 private research contracts. He is author or coauthor of 26 articles in scientific international journals. His areas of research are high-speed circuits on reconfigurable devices and communications devices.