2017 International Nuclear Atlantic Conference - INAC 2017
Belo Horizonte, MG, Brazil, October 22-27, 2017
ASSOCIAÇÃO BRASILEIRA DE ENERGIA NUCLEAR – ABEN

Agência Brasileira do ISBN
ISBN 978-85-99141-07-6
9 788599 141076

HARDWARE RESILIENCE: A WAY TO ACHIEVE RELIABILITY AND SAFETY IN NEW NUCLEAR REACTORS I&C SYSTEMS

Marcos S. Farias[1], Nadia Nedjah[2] and Paulo Victor R. de Carvalho[1]

[1] Serviço de Instrumentação - Divisão de Engenharia Nuclear
Instituto de Engenharia Nuclear - CNEN
Rua Hélio de Almeida, 75 – Ilha do Fundão
21941-906 Rio de Janeiro, RJ
msantana@ien.gov.br; paulov@ien.gov.br

[2] Departamento de Engenharia de Sistemas e Telecomunicações
Universidade do Estado do Rio de Janeiro
nadia@eng.uerj.br

ABSTRACT

The idea that systems have a property called ‘resilience’ has emerged in the last decade [1]. In this paper we intend to bring the idea of resilient systems for the hardware applied in safety-critical systems, such as the new nuclear reactor instrumentation and control (I&C) systems. The new systems (based in hardware description language (HDL) programmable devices) have been developed in response to the obsolescence of old analog technologies and current microprocessor-based digital technologies. Although HDL programmable devices have been widely used in various other industries for decades, they are still very new in nuclear reactors systems, which can be seen as a challenge and risk in the safety analyses and licensing efforts for utilities and designers. The goal of this work is to develop and test hardware architectures to tolerate the occurrence of faults, including multiple faults, minimizing the impact of the recovery process on system availability. Basic concepts of resilience in complex systems, as “return to equilibrium”, “robustness” and “extra adaptive capacity” were analyzed from the point of view of hardware architectures, leading to linkages between concepts and methods for resilience using an approach that increases reliability and simplifies the licensing process of systems based in HDL programmable devices in nuclear plants.

1. INTRODUCTION

Circumventing risks posed to life, property and environment, of which one depends on, has always been a matter of concern for human beings, already since the times when nature was the only source of threats. Technologies developed throughout history have been deeply helpful to decrease natural hazards and they have brought major upsides for humanity in several areas. Nonetheless, those very same technologies, which have already set the path for such a wide array of achievements, also have threats associated with its application. Therefore, the control and supervision systems which have been developed by focusing on several applications, must also take into account the protection aspect of it and decrease the level of risk that any given activity may pose. This is particularly important for a whole class

of applications called Safety-Critical Applications, in which issues within the controlled processes are considered to be a risk to the human being, to properties and to the environment.

A vast set of applications fit the criterion of a critical classification in terms of safety. Applications which deal with nuclear radiation, such as nuclear power plants, are a proper example of such, as well as chemical industry plants, medical equipment for the monitoring and treatment of patients, passenger transport systems and space and military endeavours.

The number of applications which happen to be classified as critical have a tendency to escalate, taking into consideration the increase in the number of systems which are automatically controlled by relying on digital processing and also the negative consequences of failures which take place in these systems. The control and supervision systems for these applications are, in its vast majority, based on programmable processors, such as programmable logic controllers (PLCs), microcontrollers or even PCs for general use. Along with these complex hardware elements we find the software, which grants these same systems with a major flexibility to perform in countless applications.

The goal of this work is focused on how one can establish hardware architectures to integrate the most recent programmable logic devices (FPGA) in systems that can be used during critical applications, suiting the requirements of standards, such as the instrumentation for nuclear reactors, and granting the already developed architectures with a degree of reliability and fault tolerance based on the concept of resilience in hardware-based systems.

The rest of this paper is organized as follows: Section 2 presents the need to use programmable devices, such as FPGA, in new Safety-Critical Systems. In Section 3 we introduce the concept of hardware resilience. Then, in Section 4, we present hardware architectures simulated in Labview graphical environment. Last, in Section 5, we draw some conclusions and point out some directions for future work.

2. FPGA-BASED SYSTEMS IN SAFETY-CRITICAL APPICATIONS

The flexibility found in processor and software-based systems pose a major challenge for critical applications, as the complexity involved in guaranteeing that these systems keep their reliability is something massive. A reliable system must be endowed with the ability to provide its functions as previously specified, with availability (in other words, having its functions active whenever they are requested), with safety (in other words, without failures) and with a protection against intrusion.

Developing architectures for critical systems closer to hardware, with less reliance on processors, software and often operating systems, has proven to be a good choice for the development of reliable systems in new projects [2] [3]. Some upsides associated with this choice are related to an easier verification and validation procedure, concerning the architectures that are closer to the hardware, a much more streamlined way to promote the diversity of technology at critical systems which require redundancy, aside from ensuring a cost reduction when a comparison is established with those prompted by a quick obsolescence of systems with processors and software.

The hardware which happens to be deployed in control systems permeates a broad spectrum of digital services, which range from the general purpose processor to the application-specific integrated circuits, the ASICs. Among the processors, supplied with highly flexible functions by software, and the ASICs, with great performance, but lacking any sort of functional flexibility (they cannot be reprogrammed after having been manufactured), one can find several architectures of programmable logic devices (PLDs) which provide the advantages related to being closer to the hardware and allow their own reprogramming.

The PLDs are part of a family comprised of programmable devices using the hardware description language (HDL), large-scale integrated circuits in which the internal hardware leaves the factory without having any function previously set, being totally configured according to the need of the final application for which it is intended to.

The most renowned among the devices of this family are the FPGA (Field-Programmable Gate Array), which have registered a tremendous evolution in recent years in order to be capable of providing architectures with top-notch performance and a big functional capacity. Figure 1 shows designs options for digital systems.

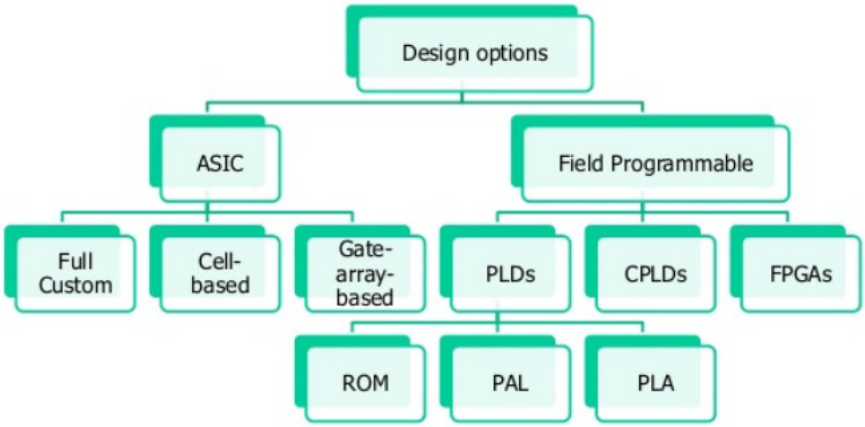

Figure 1: Design options for digital systems.

2.1. The choice for FPGA devices

The pursuit of increasing the reliability in systems for critical applications is something permanent, being regarded as a matter of concern by designers of several fields, such as the instrumentation for nuclear reactors, chemical industries, air and land-based transport, space missions, banking systems and communication networks in general, among several others. This concern is augmented when a new device technology becomes part of the options available to carry out those systems, as it happens with programmable logic devices such as the FPGA.

An array of factors prompts the industry to rely on FPGA for control and monitoring systems in brand new projects, which include critical systems. The FPGA devices may embody a simpler and easier to verify the hardware architecture, one that happens to be less expensive than processor-based systems. Aside from this, they are less vulnerable to obsolescence, taking into account the possibility of portability of the hardware, described among different manufacturers.

The route towards a wider use of the FPGA poses a challenge for the research conducted in hardware architecture modelling, which ensures the reliability of the critical system, particularly in what concerns fault tolerance issues in the hardware developed. Some works present ways to mitigate failures in hardware by using redundancy [4][5], which is the most common passive method, in addition to active methods such as the replacement of resources allocated in advance [6], or a dynamic recovery [7], in which the replacement of hardware resources is deployed right from the moment when the failure is detected.

Given the progresses in component technology, there has been an increase in the capacity and performance of the FPGA, which also includes internal processors that constitute the Systems-on-Chip (SoCs) within the device itself, demanding a team of experts in software and hardware (digital integrated circuits), more wide-ranging than the one which usually constitutes a prerequisite for processor-based systems, as well as brand new methods to keep the fault tolerance within an appropriate range for critical applications in hardware.

Safety-critical applications, such as the instrumentation of nuclear reactors, are endowed with deeply recent rules and guidelines, related specifically to the use of FPGA devices [8], thus posing a brand new challenge for designers related to the licensing issues of systems that happen to be developed with this technology. Adopting a more recent technology constitutes a problem for the development teams of critical systems, particularly for applications of this kind, as they are still lacking a vast operating experience or a significant set of data capable of being presented to the regulatory bodies.

3. HARDWARE RESILIENCE CONCEPTS

In computer systems, resilience has been defined as the persistence of reliability when it has to cope with changes [9]. A concept that happens to be more comprehensive than the fault tolerance mechanism, which is effective when the fault is within its own spectrum of effectiveness. Abiding by this context, in order to be resilient, the software must have, beforehand, reliability, an attribute that has to be kept even when facing changes in the system's disruption.

In order to keep the reliability of critical systems, whilst accommodating the strict standards and relying on the most recent devices, even when facing the everlasting additions of features in these SoCs, this work approaches an extension to the reliability and fault tolerance issue, based on the concept of resilience in hardware-based systems.

In this section we will examine basic concepts of resilience applied to hardware, indicating architectures that can satisfy these concepts.

3.1. Resilience and Robustness

Some of the earliest explorations of resilience confounded these two labels, and this confound continues to add noise to work on resilience [10]. Robustness is linked to increased ability to absorb disturbances. But, as verified by Alderson and Doyle [11], robustness is always of the

form: System X has property Y that is robust in sense Z to perturbation W. In other words, robust control works, and only works, for cases where the disturbances are well-modeled.

In hardware, robustness can be seen as redundancy. Triple Modular Redundancy (TMR) has become the most common practice because of its straightforward implementation and reliable results. The TMR fault mitigation scheme uses three identical logic circuits performing the same task in parallel with corresponding outputs being compared through majority voters [4]. TMR implementations in FPGA have the drawback that a fault may affect more than one module and cause the crash of the system. TMR is capable to mask any faults affecting the output of only one replica at any time. Figure 2 shows a triple-redundant module with a single voter at the output.

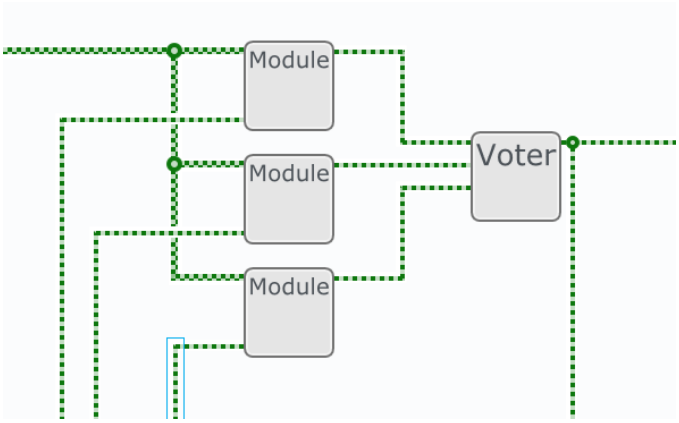

Figure 2: First approach - TMR.

3.2. Resilience and Return to Equilibrium

Wood [1] conceptualizes that *return to equilibrium*, or rebound, considers responses to specific disruptions, but much more importantly the disrupting events represent surprises, that is, the event is a surprise when it falls outside the scope of variations and disturbances that the system in question is capable of handling.

There are techniques, like S*crubbing,* available in some FPGA devices, to make the hardware *return to equilibrium* from a faulty state in its configuration memory. Configuration memory is the largest FPGA memory element and is used to store the user design [12]. Numerous approaches can be taken with respect to scrubbing, from simply reprogramming the FPGA to partial reconfiguration. These techniques are heavily dependent on the tools of the FPGA manufacturers, which can be inconvenient for critical applications.

For transient faults, a way to achieve a *return to equilibrium* in hardware is to use temporal redundancy. For this it is necessary to identify a failed module and repeat the execution.
If the fault is transient, repeating the execution will make it disappear. Comparators should be used in conjunction with, at least, two redundant modules to be able to identify a failed module. Figure 3 shows this approach. It is necessary to use registers so that the output does not pass a wrong value to the next stage while comparing the value of the redundant outputs.

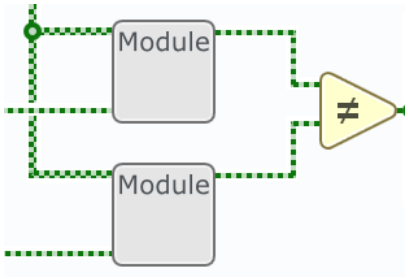

Figure 3: Second approach – redundancy with comparison.

3.3. Resilience and "Extra Adaptive Capacity"

The idea of a system that extends performance, or brings an extra adaptive capacity to withstand failures when surprise events challenge its limits [1], it's a motivating factor of many researches. When applied to hardware, it brings us closer to the idea of self-healing hardware.

Thinking about the hardware we are trying to preserve from failures in the two previous examples, a surprise event that challenges its limits would be when a failure occurs in two redundant modules and become permanent. The hardware architecture to mitigate faults of this type would have to use the techniques previously presented to mask (TMR) and detect (comparison) the faulted module, replacing this faulting module with a spare module (or reconfiguring the module) before a second module fails.

Figure 4 shows this approach. It is necessary to include a circuit (state machine) that checks the results of the comparators and identifies the module with error. In this operation, the first fault detected is considered as a transient and so proper output is selected. A second, repeated error in the same location is acknowledged as permanent.

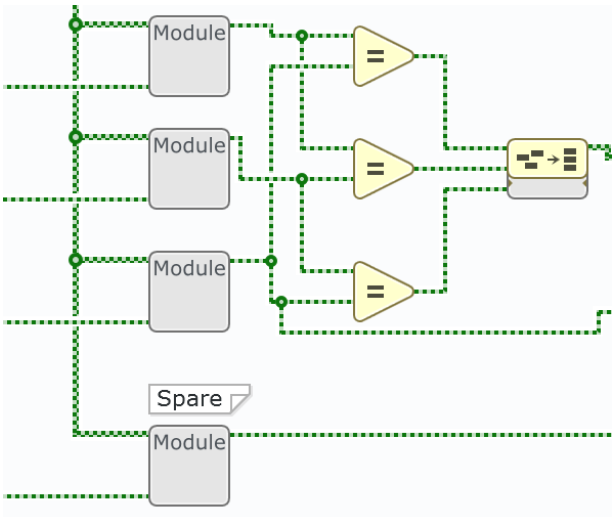

Figure 4: Third approach – TMR and comparison

4. SIMULATION OF HARDWARE ARCHITECTURES IN LABVIEW

The circuit showed at Figure 5 was selected as the module where it is intended to mitigate faults using the approaches described in Section 3. This simple combinational circuit represents one of the possible logics of shutdown (TRIP) belonging to a reactor protection system.

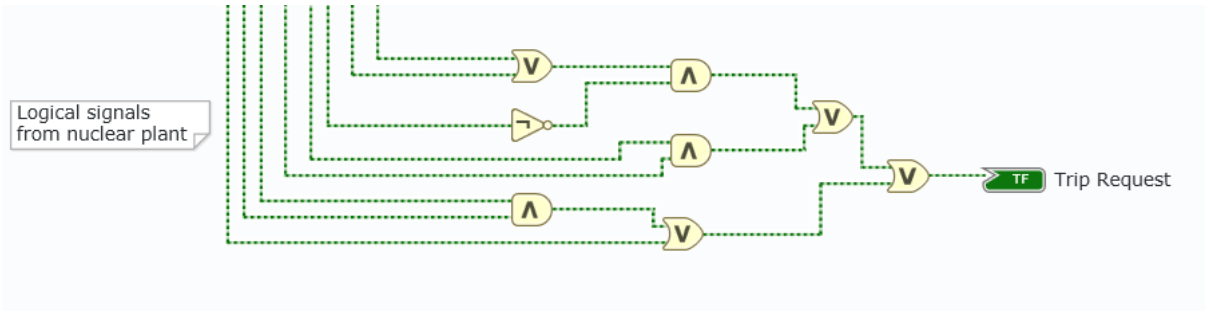

Figure 5: Design selected for digital systems.

FPGA resources are all vulnerable to radiation-induced upsets. SEU (Single Event Upset) can be defined as a radiation-induced upset that causes the state of a memory cell to change, from 1 to 0 or from 0 to 1. This is also informally known as a *bit-flip*. SETs (Single Event Transient) are transient glitches on transmission lines or in combinatorial logic. Depending on the duration and amplitude of these glitches, they may lead to errors. SEUs and SETs are unpredictable and random by nature [13].

Using functions in Labview, the failure injection in the module is simulated by randomly changing the value of the bit in its output. These changes are transient, simulating SEU or SET. It was also randomly maintained by one or two processing cycles to simulate multiple errors in the same module. The injection of a fault is done in the same module and at the same time in all three hardware architectures. Permanent changes, simulating a permanent hardware failure, will be evaluated later.

Table 1: Simulation results.

	First approach	Second approach	Third approach
Number of injected faults	17963	12041	17963
Output changed	85	214	78
Percentage	0,473%	1,77726%	0,434%

Table 1 shows a summary of the simulation results. After some time running the program it is possible to verify several failure events and the amount of those that was mitigated in the architectures. As expected, the third approach, which masks and detects the failed module, has proved more resilient to mitigate failures. However, the improvement is not so significant in relation to the first approach, TMR only. The great advantage of this approach will be to mitigate permanents failures in a module. This approach obviously will cause area overhead in the FPGA.

The changes in the output are caused by failures in two modules at the same time, which is not expected for the architectures to mitigate. Parts of a system which are capable of causing a failure for the whole system, called *single points of failure,* as the voter, were not considered in this study. These should also have redundancy to minimize single point of failure.

4.1. Permanent Hardware Failures

Only the architecture of the third approach (fig. 4) has the possibility of mitigating permanent failure occurred in a module by replacing it with a spare, thus maintaining the same triple redundancy. Maintaining hardware performance even with boundary-defying changes is one of the important characteristics when thinking about resilient systems. The resilient capacity of the system can still be high if we use techniques of reconfiguration of the module, such as scrubbing, in addition to the spare module.

3. CONCLUSIONS

This work presented the proposal of hardware architecture models which can integrate the features of new FPGA devices on systems that may be used in critical applications, as nuclear reactor protection system, adding an adaptive dimension to the concept of fault tolerance in hardware, described as resilience in hardware.

The simulations have shown that these hardware architectures work efficiently to mitigate module failures, allowing subsequent studies to use these approaches to assess the resilience of critical systems using FPGA, evaluating benefits in relation to the occupied area overhead in these devices.

Some studies are of the opinion that hardware-based systems, without an operating system, will keep increasing their influence in critical applications. The continuation of this work will contribute to enabling the adoption of new FPGA devices by hardware development teams for critical systems, as nuclear instrumentation reactors, benefitting the issues related to the licensing of these systems with regulatory bodies.

ACKNOWLEDGMENTS

We gratefully acknowledge support from the Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro (FAPERJ) and Comissão Nacional de Energia Nuclear (CNEN).

REFERENCES

1. Woods, D, “Four Concepts for resilience and the Implications for the Future of Resilience Engineering,” *Reliability Engineering and System Safety*, (2015) http://dx.doi.org/10.1016/j.ress.2015.03.018
2. Korea Atomic Energy Research Institute (KAERI), *Survey of the CPLD/FPGA Technology for Application to NPP Digital I&C System*, Tech. Rep. (2009).

3. K. O'Neill, G. R. Newell and S. K. Odiga, "Protecting flight critical systems against security threats in commercial air transportation," *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, Sacramento, CA, pp. 1-7 (2016).
4. C. Carmichael. *Triple Module Redundancy Design Techniques for Virtex FPGAs*. Xilinx Application Notes, XAPP197 (2001).
5. X. Wang, K. E. Holbert, and L. T. Clark. "Single event upset mitigation techniques for fpgas utilized in nuclear power plant digital instrumentation and control," *Nuclear Engineering and Design*, 241 (2011).
6. Cheatham, J. A., Emmert, J. M., and Baugart, S. "A survey of fault tolerant methodologies for FPGAs," *ACM Trans. Des. Autom. Electron. Syst. 11*, 2, 501–533 (2006)
7. Emmert, J. M., Stroud, C. E., and Abramovici, M. "Online fault tolerance for FPGA logic blocks," *IEEE Trans. VLSI Syst. 15*, 2, 216–226 (2007).
8. International Electrotechnical Commission, *IEC 62566 - Nuclear Power Plants Instrumentation and Control Important to Safety Development of HDL-programmed Integrated Circuits for Systems Performing Category A*. International Electrotechnical Commission (2012).
9. Laprie, J.C., From Dependability to Resilience. *In International Conference on Dependable Systems and Networks (DSN 2008),* Anchorage, AK, USA, volume 8. (2008).
10. Woods, D. D. "Essential Characteristics of Resilience for Organizations." In E. Hollnagel, D.D. Woods and N. Leveson, eds., *Resilience Engineering: Concepts and Precepts* (pp. 21-34). Ashgate, Aldershot, UK. (2006).
11. Alderson, D. L. and Doyle, J. C. "Contrasting views of complexity and their implications for network-centric infrastructures," *IEEE SMS Part A*, 40:839-852. (2010).
12. Nunes, J.L. "Improving FPGA resilience through Partial Dynamic Reconfiguration," *CoRR* abs/1608.06559. (2016).
13. Petersen, E., *Single Events Effects in Aerospace*, John Wiley & Sons (2011).