

Article

Evaluation by Neutron Radiation of the NMR-MPar Fault-Tolerance Approach Applied to Applications Running on a 28-nm Many-Core Processor

Vanessa Vargas ^{1,*},[†] , Pablo Ramos ¹ and Raoul Velazco ²¹ DEEE, Universidad de las Fuerzas Armadas ESPE, Sangolquí 170501, Ecuador; pframos@espe.edu.ec² Laboratoire TIMA, Université Grenoble-Alpes, 38400 Grenoble, France; raoul.velazco@univ-grenoble-alpes.fr

* Correspondence: vcvargas@espe.edu.ec; Tel.: +593-2-398-9400

[†] Current address: Departamento de Eléctrica, Electrónica y Telecomunicaciones, Universidad de las Fuerzas Armadas ESPE, Av. General Rumiñahui s/n, P.O. Box 171-5-231, Sangolquí 170501, Ecuador.

Received: 27 September 2018; Accepted: 31 October 2018; Published: 8 November 2018



Abstract: Currently, there is a special interest in validating the use of Commercial-Off-The-Shelf (COTS) multi/many-core processors for critical applications thanks to their high performance, low power consumption and affordability. However, the continuous shrinking of transistor geometry and the increasing complexity of these devices dramatically affect their sensitivity to natural radiation, and thus diminish their reliability. One of the most common effects produced by natural radiation is the Single Event Upset which is the bit-flip of a memory content producing unexpected results at application-level. For this reason, manufacturers and users implement hardware and software error-mitigation techniques on multi/many-core processors. In this context, the present work aims at evaluating a new fault-tolerance approach based on N-Modular redundancy (NMR) and partitioning called NMR-MPar by means of 14 MeV neutron radiation ground testing in order to emulate the effects of high-energy neutrons present at avionics altitudes. For evaluation purposes, a case-study is implemented on the 28 nm CMOS KALRAY MPPA-256 many-core processor running two complementary benchmarks applications: a distributed Matrix Multiplication and the Travel Salesman Problem. Radiation experiments were conducted in GENEPI2 particle-accelerator. The correctness of the results of the application when an error is detected confirms the approach's effectiveness and boosts their usage on avionics applications.

Keywords: radiation ground testing; many-core processor; single event effect; single event upset; partitioning; redundancy

1. Introduction

Nowadays, computing systems based on COTS multi/many-core processors are the key solution for achieving high level of parallelism combined with reduced power of consumption within a reasonable degree of reliability. However, the miniaturization of these devices has increased their sensitivity to the effects of natural radiation. These effects are produced by the interaction of a single energetic particle with a semiconductor material, and are called Single Event Effects (SEE). SEEs can be transient or permanent. Transient effects may cause dysfunctions in electronic devices, performance degradation and even partial or total destruction of the device [1]. Among them, the single event upset (SEU) is the most representative, since it may produce the modification of the content of a memory cell with unpredictable consequences at application level.

Having more sensitive devices affects dramatically the reliability of computing systems. The high demand of reliability for several applications implies that device manufacturers implement complex

error-detection and correction circuits. Mitigation error techniques applied to these devices allow reducing significantly the soft-error rate. Nevertheless, these implementations lead to an overhead that causes unpredictable slowdowns to the system [2]. For this reason, it is not convenient to protect all sensitive areas even if a physical protection is feasible. Furthermore, device vendors search to decrease the cost of designing and testing circuits. Therefore, it is essential to improve the reliability of the system by exploiting the intrinsic characteristics of multi/many-core processors.

Benefitting of the multiplicity of cores, the literature proposes several approaches to improve the reliability of these devices through redundancy and (few works) by using partitioning. The most relevant works related to these advanced devices are summarized as follows.

Concerning fault-tolerance through redundancy, some authors [3–7] propose redundant multithreading techniques to detect and recover from faults on multi-core processors configured on Symmetric Multi-Processing (SMP) mode. The general trend is based on running identical copies of a process as independent threads. Nonetheless, in these works, details about the validation of the proposed techniques are not provided. Reference [8] proposes a Process-Level Redundancy with low overhead within a unique OS. This approach is evaluated by SEU fault-injection on randomly selected instruction on redundant processes.

Other authors improve the reliability by using N-Modular Redundancy at different levels. Reference [9] proposes a scheduler that applies a Triple Modular Redundancy (TMR) on multi-threading applications running on many-core architectures. The scheduler detects error and isolates faulty cores. The proposal is evaluated at simulation level by injecting one fault per run on a single cluster of 16 cores.

In [10], a framework using an adaptive NMR system is presented for improving the reliability of COTS many-core processors. It selects the best number of replicas to maximize the system reliability based on the voter reliability. This analysis is only theoretical and has not been evaluated. Reference [11] proposes a Soft NMR that improves the robustness of classical NMR systems by using error statistics. Its effectiveness is illustrated by an example in image coding.

Lastly, some authors use the conjunction of several mitigation techniques. For instance, reference [12] proposes the implementation of a kernel-level checkpoint rollback, a process and thread level redundancy and a heartbeat on a system based on the RHDB Maestro processor to mitigate hardware and software errors. Nonetheless, there is no evaluation of the effectiveness of their proposal.

Regarding proposals of improving multi/many-core reliability through fault-tolerance techniques by partitioning, it can be found that temporal partitioning has been proposed as a solution in mixed-criticality systems to improve multi-core processor reliability [13,14]. Also, a concept of parallel Software Partition (pSWP) is proposed for many-cores to guarantee time isolation. This approach is evaluated in a simulator compatible with PowerPC ISA (Instruction Set Architecture) [15].

In addition to temporal isolation, other authors propose also the use of spatial isolation. Reference [16] presents a methodology and tools needed to implement multiple partitions with rigorous temporal and spatial isolation. This approach supports mixed-criticality based on the XtratumM virtualization layer. Reference [17] uses the bare-board hypervisor Xtratum and duplex execution at task level on COTS device to propose a hybrid approach based on partitioning and redundancy. The authors have implemented their proposal on dual-core. However, there is no evaluation of the proposal. This work is probably the most related work to our fault-tolerance approach (called NMR-MPar approach) which is based on N-Modular redundancy and partitioning concepts. It has been presented and evaluated through software fault-injection in reference [18]. Results have shown its effectiveness to improve reliability.

It is well-known that authors use simulation or fault-injection for evaluating their fault-tolerance approach effectiveness. However, for validation purposes in avionics and spacecraft applications, it is important to evaluate the approach by exposing the system to natural radiation. Real-life tests are the most trustworthy way to study the effects of radiation on electronic circuits and to measure the soft-error rate of a given device, since it is tested in the radiation environment where it is intended

to work (terrestrial, atmosphere or space) [19]. It consists of gathering as many devices as possible with the aim of increasing the number of observed SEEs. However, the necessity of a huge number of devices and a long exposure time to obtain statistically satisfactory results boosts the use of other methods. Consequently, accelerated radiation ground tests are widely used to obtain significant data in a short exposure time to test the reliability of electronic devices. In the literature, there are a few works that have evaluated the SEEs sensitivity of multi/many-core processors by radiation. The most relevant are summarized as follows:

Reference [20] presents the SEE radiation results of the 49-core Maestro ITC microprocessor. Maestro is a Radiation Hardened by Design (RHBD) device based on the Tiler TILE64 processor intended to be used in space applications. Experimental tests have been conducted at the Texas A/M University's (TAMU) cyclotron facility using 15 and 25 MeV ions. Results have demonstrated that the L1 and L2 cache memories are handled by an effective Error Detection and Correction (EDAC) mechanism.

Reference [21] establishes a dynamic cross-section model for a multi-core server based on quad-core processors built-in 45 nm bulk CMOS technology. For evaluation purposes, authors have exposed the server to 14 MeV neutrons. In [22] the radiation sensitivity evaluation of a modern Graphic Processing Units (GPUs) designed in 28 nm technology is presented. Radiation tests have been conducted at Los Alamos facility using 14 MeV energy neutrons. It also provides a hardening strategy based on Duplication with Comparison.

Reference [23] evaluates the dynamic sensitivity of different scenarios running on the Freescale P2041 multi-core. This work illustrates that a 45 nm Silicon-On-Isolator (SOI) quad-core processor is about four times less sensitive to SEE than its CMOS (Complementary Metal-Oxide Semiconductor) counterpart when the devices are evaluated under 14 MeV neutron energy. From the results, it can be seen that the dynamic sensitivity of the device strongly depends on the multi-processing mode used. It has been demonstrated that the Asymmetric Multi-Processing (AMP) bare-board mode is more reliable than SMP mode. A considerable vulnerability of the Operating System (OS), especially concerning system crashes and exceptions, is observed. This vulnerability has put in evidence that the implementation of redundancy at user level is not enough to overcome dependability issues. Therefore, it is clear that there is an imperative necessity of using the partitioning concept to guarantee both reliability and availability of the system.

In [24], the 14 MeV neutron sensitivity of the KALRAY Multi-Purpose Processing Array (MPPA-256) many-core processor is presented. Results suggest that Error Correcting Codes (ECC) and interleaving implemented in Shared Memories (SMEMs) of the compute clusters are very effective to mitigate SEUs consequences since all detected events of this type were corrected. Furthermore, the evaluation of the device's dynamic response shows that by enabling the cache memories, it is possible to increase the performance of the application without compromising the reliability.

The present work aims at evaluating the effectiveness of the NMR-MPar fault-tolerance approach implemented in a many-core processor operating in a harsh radiation environment. For that purpose, two distributed parallel applications were implemented on the KALRAY MPPA-256 many-core processor: a *cpu-bound* and a *memory-bound* application that assess complementary chip resources. The evaluation was carried-out under 14 MeV neutron radiations at GENEPI2 particle accelerator. Results are extrapolated to avionic altitudes to appraise the reliability improvement. The current paper is an extension of the first author's PhD thesis [25].

In this work, radiation experiments were performed with 14 MeV neutrons in order to obtain in an easy and cost-efficient manner the sensitivity of the many-core processor for avionics applications. The availability of the neutron facility and the possibility of performing simultaneous experiments were other important reasons to choose 14 MeV. Nonetheless, the atmospheric neutron spectrum includes different neutron energies. At avionic altitudes the particle fluxes above several hundreds of MeV are not negligible and their impact on the system error rate can be significant [26]. However, the radiation tests under 14 MeV neutrons are largely used because it has been proved that due to the

constant decrease of the critical charge, it appeared that 14 MeV cross-section got close to the saturated cross section for most advanced unhardened devices [27]. In addition, authors of reference [28] have demonstrated that based on a bounded sensitivity of SRAM 90-nm memories obtained from 14MeV neutron test, it has been possible to deduce the neutron and proton saturated cross section within a very well approximated when applying a margin of 2 to the results.

The remainder of the paper is organized as follows: Section 2 details the materials and methods including the case study used in this work and the experimental setup. In Section 3 the experimental results are presented. Section 4 discusses the results. Finally, Section 5 concludes the paper.

2. Materials and Methods

2.1. Target Device

The selected device is the KALRAY MPPA-256 many-core processor due to its advanced CMOS 28 nm manufacturing technology and its architecture, which is similar to the Shen Wei SW26010 (260 cores) many-core processor, which is the base processor of the second TOP500 super computer list (June 2018). In addition, the MPPA many-core is focus of interest of embedded community to study the possibility of its use in critical real-time embedded systems [29,30]. For instance, CAPACITES is a project that gathers French academics and industrial partners to analyse the possibility of using MPPA many-core processors for critical embedded systems.

The MPPA-256 considered in this work is the second version called Bostan. This processor is manufactured in TSMC CMOS 28HP technology. The processor operates between 100 MHz and 600 MHz, for a typical power ranging between 15 W and 25 W. Its peak floating-point performances at 600 MHz are 634 GFLOPS and 316 GFLOPS for single and double-precision respectively. It integrates 256 Processing Engines (PEs) cores and 32 Resource Managers (RMs) cores distributed in a clustered architecture. All cores are based on the same VLIW 32-bit/64-bit architecture. The MPPA-256 comprises two Input Output cluster (IO) for external communication and minimal processing, and sixteen Compute Clusters (CC) exclusively for processing. The communication intra-cluster is achieved by buses while the communication inter-cluster is performed by a wormhole switching network-on-chip (NoC) with 32 nodes and a 2D torus topology [29]. An overview of the many-core processor is illustrated in Figure 1.

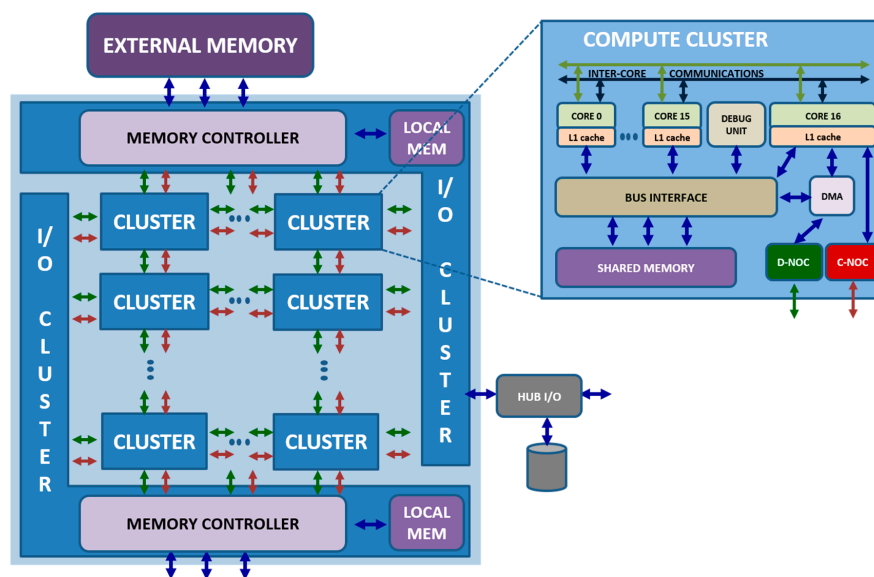


Figure 1. Many-core processor components. MPPA-256 has 2 IO clusters and 16 Compute Clusters.

The VLIW core includes General Purpose Registers (GPRs) and System Function Registers (SFRs). Outside the processor, the MPPA comprises different types of specific registers for controlling DMA,

Data-NoC (D-NoC), Control-NoC (C-NoC), cluster power controller, trace, debug, and the different inputs/outputs (IO). The VLIW core implements separate instruction and data cache. There is no hardware cache coherency mechanism between cores nor between data and instruction cache. However, to enforce memory coherency, several software mechanisms are available to programmers.

The SMEM is composed of 16 independent memory banks of 16,384 64-bit words, for a total capacity of 2 MB. Each memory bank is associated with a dedicated request arbiter that serves 20 bus masters: the D-NoC Rx (receive) interface, the D-NoC Tx (transmit) DMA engine, the Debug System Unit (DSU), the RM core, and 16 PE cores. Figure 2 illustrates the compute cluster buses. The 16 memory banks are arranged in two sides of eight banks, called left side and right side. The connections between the memory bus masters are replicated in order to provide independent access to the two sides. The private paths of the 16 PE cores are connected to the 16 memory bank arbiters. Other bus masters (D-NoC Rx, D-NoC Tx, DSU, RM) have their own private path also connected to the 16 memory bank arbiters [31].

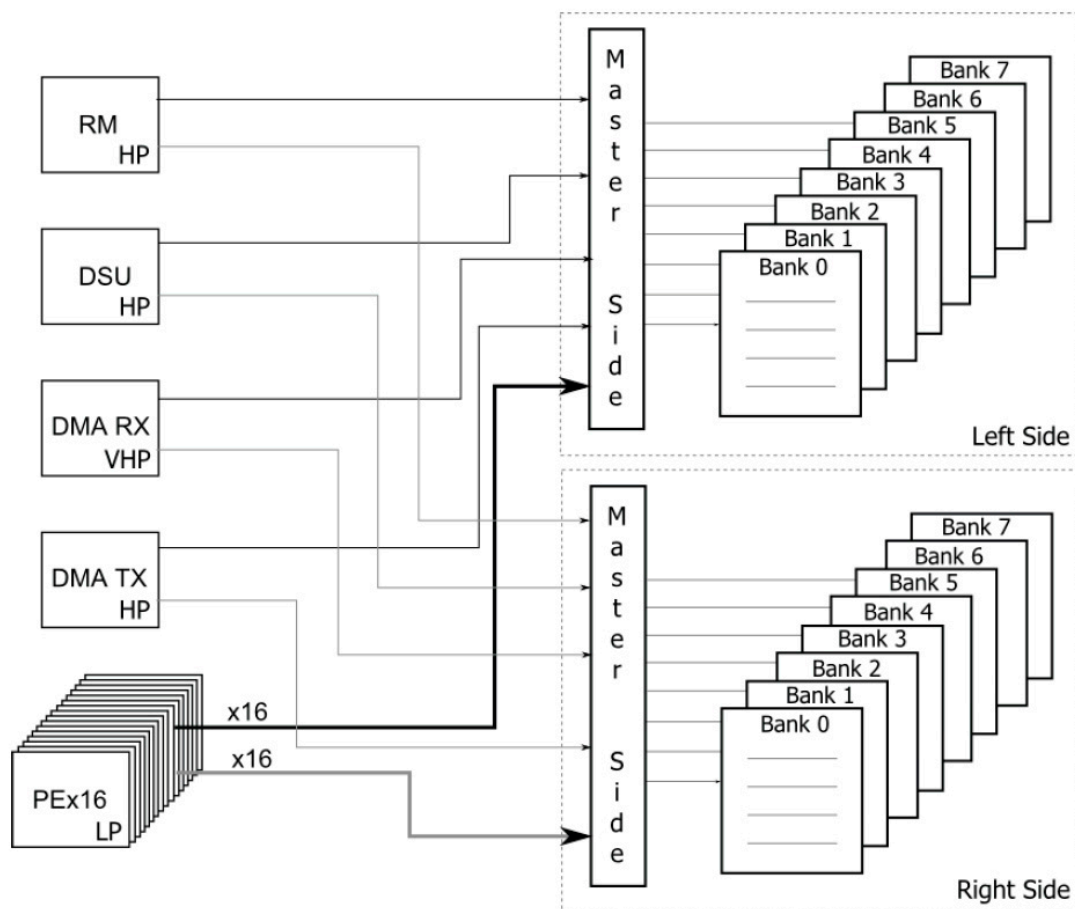


Figure 2. Compute cluster bus masters [31].

The main components of the many-core processor are covered by error protection mechanisms except the instruction and data cache memories of the VLIW core that are protected by parity. The SMEM implementation interleaves bits of eight adjacent 64-bit words which allows localized errors to spread as multiple single ECC (SECC) errors. SECC errors are detected and corrected on the fly. The NoC router queues (512 of 32-bit flits each) are also protected by ECC. Note that SECC errors are silently corrected while Double ECC (DECC) errors are detected and signaled. The registers do not implement any protection mechanism.

The MPPA-256 many-core is embedded in a development platform (MPPA Developer) containing the MPPA ACCESSCORE SDK version 2.5 for developing, optimizing and evaluating applications.

The SDK includes three programming models for developing an application: POSIX, Kalray OpenCL and Lowlevel. For running a POSIX, the manufacturer has implemented a proprietary OS called NodeOS. Since, the MPPA-256 is a coprocessor, a CPU host (HOST) is needed to manage it. The communication between the HOST and the MPPA-256 is achieved using specific drivers provided by the manufacturer. The HOST is based on an Intel core I7 CPU operating at 3.6 GHz and running a Linux OS. MPPA DEVELOPER includes a PCIe board MPPA-256 Bostan version and a PCIe board for debug and probe [31]. This board implements a module for controlling current and voltage aiming at mitigating latch-up events. Table 1 summarizes the main sensitive areas of the target many-core processor.

Table 1. Sensitive zones of the MPPA-256 many-core processor.

Sensitive Zone	Location	Capacity	Description
SMEM	Computing Cluster	2 MB per cluster	Static Shared Memory
SMEM	I/O Cluster	4 MB per cluster	Static Shared Memory
IC-CC	VLIW core	8 KB per core	Instruction Cache
DC-CC	VLIW core	8 KB per core	Separated Data cache
IC-IO	IO VLIW core	32 KB per core	Instruction Cache
DC-IO	I/O cluster	128 KB per I/O	Shared Data cache
GPR	VLIW Core	64 registers of 32 bits per core	General Purpose Registers
SFR	VLIW Core	51 registers of 32 bits per core	System Function Registers
Shared resources	On-chip	Information Not Available	NoC, Buses

The MPPA-256 implements ECC and interleaving in its shared memories, and parity in the processor core's caches memories. Evaluation under neutron radiation of previous work on the same many-core processor has proved the effectiveness of the protection mechanisms implemented on its memories. Reference [24] shows that all the events observed during radiation tests on SMEM and cache memories were corrected during static and dynamic tests. Furthermore, results show the pertinence of enabling the cache memories of this device for improving its performance with minimum reliability consequences. However, it is fundamental to configure by software means the invalidation of data and instruction cache memories when parity errors are detected.

In case of detection of a cache parity error when the Node-OS runs on the computing cluster, the manufacturer has established by default a TRAP EXCEPTION that halts the related core. It was desired to change the handling trap code in order to invalidate the cache memory and to continue with the execution of the related process. This configuration was tested in a previous work when the intrinsic sensitivity to SEE of the MPPA was assessed [24]. Since Node-OS is not an open source code, the manufacturer was asked to do this change. However, the manufacturer did not agree to do it, and only gave some ideas about how to do it with a wrap code.

2.2. Benchmark Applications

This work proposes the use of two types of distributed applications: a CPU-bound and a memory-bound. The selected CPU-bound application is the Traveling Salesman Problem (TSP), a Non-deterministic Polynomial (NP) hard problem very used for evaluating computing system optimization [32]. This application aims at finding the shortest possible route to visit n cities, visiting each city exactly once and returning to the departure city. To solve the problem there are several proposals. This work uses a brute force exact algorithm based on a simple heuristic. The implemented version of TSP on the MPPA by authors of [33] was used as a basis. Figure 3 illustrates it. The light dark format in the illustration of the solution means that this possible route was not completely explored. It was discarded when the algorithm arrive to city 4, because the path distance until this point (65) is larger than other complete solution (62).

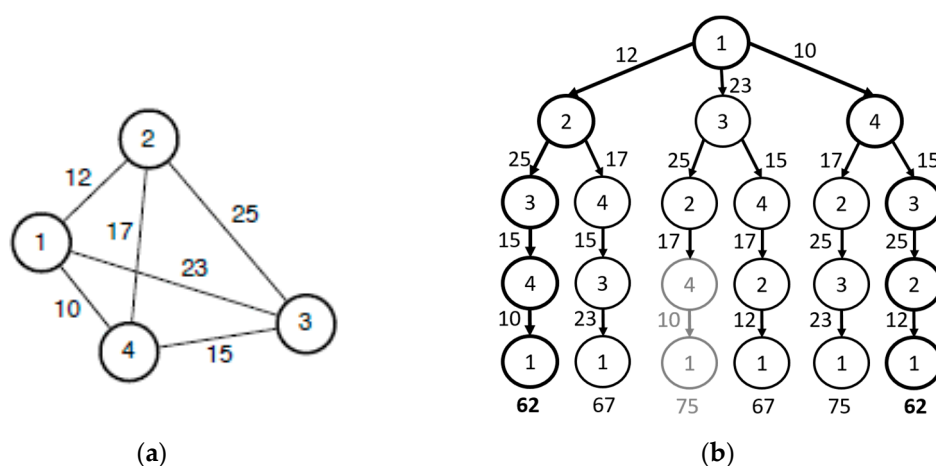


Figure 3. (a) Illustration of the TSP Problem; (b) Illustration of the solution.

On the other hand, the Matrix Multiplication (MM) was chosen as memory-bound application. The MM is widely used for solving scientific problems related to linear algebra, such as systems of equations, calculus of structures and determinants among others. Concerning avionic applications, MM is used for image processing, filtering, adaptive control, and navigation and tracking. In addition, the parallelism of MM is one of the most fundamental problems in distributed and High-Performance Computing. There are many approaches proposed to optimize performance. The present work implements the approach divide and conquer. The selected application is a collaborative 256×256 matrix multiplication. This computation is iterated to sum it 8192 times as stated in Equation (1).

$$C = \sum_1^{8192} A \times B \quad (1)$$

A , B and C are single precision floating-point matrices. Due to data are typed single precision (4 bytes per matrix element), each matrix occupies 4×256 bytes. The size of the matrix was chosen so that data remain in the local SMEM memory.

Both applications were configured with 4 CCs in order to maximize the use of resources. Table 2 summarizes the characteristics of both applications. The present work evaluates under neutron radiation the same case study implemented and evaluated by fault-injection in our previous work [18]. During neutron radiation tests, all sensitive areas summarized in Table 1 were assessed, but not the whole capacity of each area. Since the tests are dynamics, the area evaluated depends on the specific type of application and the related used resources.

Table 2. Benchmark applications characteristics.

Characteristic	TSP	MM
Application Type	CPU-bound	Memory-bound
Data Type	Integer	Single Floating point
Processing Data	Common data	Independent data
Type of work during execution	Collaborative amount cores	Independent per core
Work load	Interactive	Batch
Distribution of work per processing element core	Jobs by using queues of work	1/16 of the cluster result
Use of buses and NoC	Continuous use to request new tasks and to broadcast the newly found minimum distance and path.	Maximizes their use at the beginning and at the end of the computation.
Validation of result	Verification of the response includes a valid path.	A standard CRC-32 set by IEEE 802.3 was used

Table 2. Cont.

Characteristic	TSP	MM
Behaviour under SEU	If SEU affects a data, there is a high possibility to find the correct result. If one core halts, the others can find the correct result.	If SEU affects a data, there is a high possibility to have erroneous result. If one core halts, the result will be erroneous.
Application Use	To evaluate computing system optimization, routing of traffic, planning, logistics and manufacturing.	To solve scientific problems related to linear algebra. Concerning avionic applications, MM is used for image processing, filtering, adaptive control, and navigation and tracking

2.3. Fault-Tolerance Approach

This work evaluates the fault-tolerance approach called NMR-MPar under neutron radiation. This approach uses redundancy and partitioning as basic concepts to improve the reliability of applications running on multi-core and many-core processors. NMR-MPar takes advantage of the multiplicity of cores to implement redundancy techniques that allow masking faults. Complementary, partitioning protects against not authorized access and data modification by temporal and spatial isolation of each partition. The proposed approach was presented in a previous work [18].

NMR-MPar proposes a physical resource distribution to each partition in order to minimize the propagation of faults producing dysfunctions in other resources or cores. Each partition can be setup as a mono or multi-core running on different multiprocessing modes: Asymmetric Multi-Processing or Symmetric Multi-Processing mode with different programming models: bare-metal, OpenMP, POSIX, etc. Consequently, there is a considerable versatility on the system configuration that is enhanced by the number of cores comprised in the device. It is important to note that each partition can be configured in bare-metal or with an independent OS. Furthermore, it is proposed (for critical functions) that N partitions with the same configuration participate of an N-Modular Redundancy system. Thus, several partitions execute the same application and the results are used by a voter to build a fault tolerant system. The voter system can include one or more cores of the device, or an external one if desired.

For evaluation purposes, a case study with different scenarios that exploit massive parallelism is assessed. Distributed algorithms of TSP and MM applications were implemented on the MPPA-256 many-core processor. The selection of both benchmark applications was done to verify the approach effectiveness when testing complementary on-chip resources. The case study implements a Quad Modular Redundancy system profiting from the device architecture, in which the resources of each IO cluster are shared by a couple of redundant modules. Therefore, if one IO cluster suffers a dysfunction, the redundant system continues working as a Double Modular Redundancy.

Each module of the redundant system runs the application in an independent partition. Each of these partitions is composed of one RM core belonging to the IO cluster and four Compute Clusters (CC). That means that a module processes an instance of the redundant system. At the end of the execution, each instance stores the results in both SMEMs IO Cluster memory. The system has two voters, each one is located in a different IO cluster. Before voting, the obtained results are validated by means of the verification of a valid and complete path in the case of TSP and by using a Cyclic Redundancy Check (CRC) method in the case of MM. By using only validated data, each voter applies majority voting to select the correct result. The following voting criteria is used:

- The correct response can be determined if three or four results are equal.
- Also, it is possible to establish the correct response if two results are equal and the other two are different between them.
- If there are two pairs of two equal results, the voter logs an error.

After voting, the correct result is sent to the other IO cluster. The voter that completes the operation first logs the correct results. The voter and the coordinator of the inter-cluster communications cores

are not part of the partitions that run the application. Both are independent mono-core partitions. Therefore, the system is composed by four multi-core partitions (P0 to P3) and four mono-core partitions (P4 to P7). Figure 4 summarizes the implemented case study.

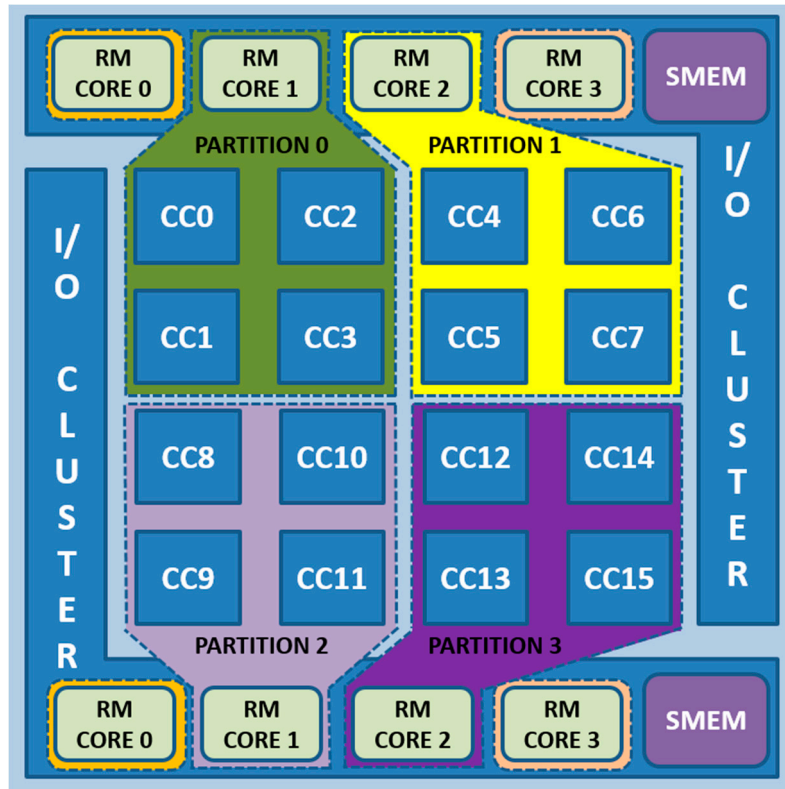


Figure 4. Configuration of the proposed case study for evaluating NMR-MPar approach. RM0 of each IO Cluster coordinates the inter-cluster communications. RM1 and RM2 cores are the masters of each instance and RM3 is the voter.

Further information about the details of the implemented case study can be found in reference [18]. As it was stated for application operating in harsh radiation environment such as avionic or spacecraft domain, fault-injection evaluation can be considered as a preliminary evaluation. However, it is compulsory the evaluation under radiation for approval purposes.

2.4. Evaluation of the Radiation Effects

Technology advances affect the device sensitivity to radiation effects, and consequently, its error rate and reliability. Particle accelerators are commonly used to characterize integrated circuits to radiation at ground level and to obtain significant results in a short period of time. A particle accelerator is a machine that uses electric fields to accelerate elementary particles such as heavy-ions, protons, electrons to very high energies producing a beam of charged particles. The more the particles interact with the device, the more SEE can be obtained. The drawbacks of this evaluation strategy are: particle beam spectrum is not really that one of the natural radiations, there are few facilities around the world, high cost in experiment setup and tests. Radiation experiments were carried-out at GENEPI2 (GEnerator of NEutron Pulsed and Intense) particle accelerator facility located in Grenoble, France.

The sensitivity of a device exposed to ionizing radiation is expressed in terms of the cross-section (σ). It is an effective area that quantifies the intrinsic probability that an ionizing particle crossing 1 cm² area produces an SEE. Equation (2) defines the cross-section as number of events produced by a particle fluence per unit area (cm²).

$$\sigma = \frac{N_{ev}}{\Phi} \quad (2)$$

where N_{ev} is the number of detected events and Φ is the particle fluence, which is the particle flux (φ) integrated on a certain period of time. For semiconductor memories where the capacity is known, σ can be expressed in cm^2/bit or $\text{cm}^2/\text{device}$.

In this work, the term cross-section (σ) is related to SEU events and provides the average number of particles needed to cause a bit-flip in a memory cell. To analyze the results, it is important to consider the consequences of bit-flips in memory cells (“soft error”) that can be classified as follows:

- *Masked fault*: there is no apparent effect on system; all the system and application processes are terminated in a predefined way and the results are correct.
- *Application erroneous result*: the result of the application is not the expected one.
- *Timeout*: when the program does not respond after a duration equal to the worst-case execution time
- *Exception*: the application triggers an exception routine.

From the above list, the most critical one is the “*Application erroneous result*” because it is not detected by the system and can cause unpredictable consequences. *Timeouts* and *exceptions* are not so critical, since the system accounts for a problem with the execution of the application. Therefore, it could take the corresponding action, such as running the application again. In spite of the slowdown of the system and the reduction of performance, reliability is not affected.

Reliability is the probability of having no failure in a semiconductor device within a given period of time. The reliability is function of the failure rate. For electronic devices, the failure rate is considered as a constant. When failure rate (λ) is constant, the following equation is applied.

$$R(t) = e^{-\lambda t} \quad (3)$$

Being $\lambda = \sigma * \varphi$, to obtain the failure rate produced by SEE, where φ is the particle flux in a given environment.

The Soft Error-Rate (SER) is the rate at which soft errors appear in a device or system for a given environment. When the operating environment is known, the SER can be expressed in Failure in Time (FIT) or in Mean Time between Failure (MTBF). One FIT is equal to a failure per billion hours. The sensitivity of semiconductor memories is often given in FIT/Mb or FIT/device. The FIT value can be predicted by simulation or is obtained experimentally in radiation facilities. The cross-section of a device can be used to calculate the SER as follows:

$$FIT \text{ value} = \sigma \times \varphi \times 10^9 \quad (4)$$

2.5. Experimental Setup

For radiation experiments, the PCI-e board containing the MPPA-256 many-core processor was located inside the armored chamber. The rest of the DEVELOPER platform was located outside the chamber. To power up the PCIE-e board a secondary power supply was used. The connection between both, the DEVELOPER and the board was done via a JTAG port.

The device under test was decapsulated and placed facing the center of the target perpendicular to the beam axis at a distance of 28 ± 0.5 cm. The DUT fan was placed laterally to cool-down the device rather than being placed on the device. It was done for not interfere the neutron flux. Consequently, the computer cluster frequency was set to 100 MHz to reduce power dissipation. In previous work [24], the experimental results have proved that frequency has no impact in the occurrence of SEUs. In that work, the MPPA-256 processor was tested at 100, 200 and 300 MHz.

The bias voltage of the device was set to 0.9 V. The neutron beam energy was 14 MeV with an estimated flux of $1.23 \times 10^5 \text{ n}\cdot\text{cm}^{-2}\cdot\text{s}^{-1}$ at 500 Hz frequency with an error of $\pm 0.1 \times 10^5 \text{ n}\cdot\text{cm}^{-2}\cdot\text{s}^{-1}$. For protecting the rest of the PCI-e board, a 5-cm thickness polypropylene block was used. In the presented experiments, the power supply was controlled by means of a current-voltage controlling module implemented in the platform (PCIe board MPPA).

For these experiments, there was no additional anti-latch-up circuit since current and voltage levels are automatically controlled by the Bostan PCIe board. Figure 5 illustrates the experimental set-up of the neutron radiation experiments at the GENEPI2 facility. The protocol used for the experimental tests is described by the JEDEC STANDARD, JESD89A [34].

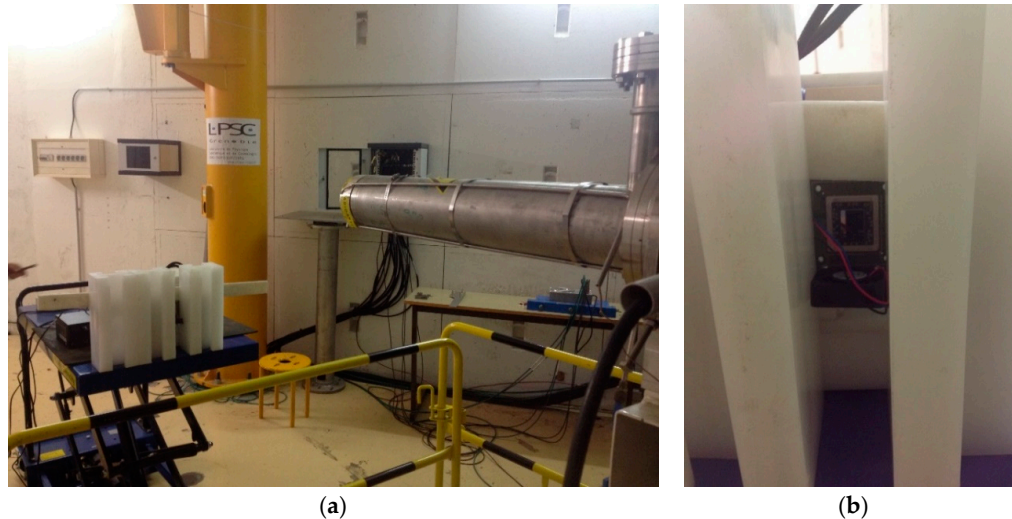


Figure 5. (a) Experimental Set-up at GENEPI2; (b) MPPA target device.

2.6. General Considerations

In order to evaluate the efficiency of the NMR-MPar approach under different scenarios, the MM and TSP application were considered due to their complementary application nature. In addition, in the case of TSP, it was considered two different problem sizes: a 16-city and a 17-city problem, in order to analyze the impact of the application exposure time in the results. Thus, three different configurations were assessed.

- The first one implements TSP solving for 16 cities.
- The second one runs TSP solving for 17 cities.
- The third one considers 256×256 MM.

Furthermore, it was necessary to compare the obtained results with those obtained from running the application without redundancy exposed to same radiation conditions. Therefore, six scenarios were tested. It is important to note that the scenarios that implement the NMR-MPar approach occupy all the computing clusters (CC0 to CC15), while the application without redundancy runs only on CC0 to CC3. That means, NMR-MPar scenarios have a sensitive zone four times bigger than the scenarios without redundancy. Table 3 summarizes the main characteristics of each scenario.

Table 3. Benchmark characteristics for each scenario.

Application	w/o Redundancy		Used Resources	NMR-MPar		Used Resources
	Standard Exec. Time			Standard Exec. Time		
	[Gcycles]	(s)		[Gcycles]	(s)	
TSP-16	9.51	100.3	4 CCs + 2 RMs of IO	11.29	128.6	16 CCs + 8 RMs of IOs
TSP-17	58.58	594.6		60.7	626.2	
MM	5.1	69.6		7.22	104.4	

Each scenario was evaluated in different radiation test campaigns. During the test campaigns, for preventing the propagation of errors between successive executions, at the end of each run the HOST resets the platform and reloads the code in the many-core processor.

3. Results

From previous radiation experiments on the same target device in the same particle accelerator, it is possible to see that all single ECC errors produced in the SMEMs were corrected by the ECC, while Register Trap errors remain uncorrected. In addition, by invalidating the cache memory when a parity error is detected, and continuing with the process execution, it was decreased the number of system exceptions [24]. It is important to note that this invalidation was possible thanks to working in bare-metal mode that allows user to program a custom trap handling. In bare-metal, no OS is used, then the programmer uses the Board Support Package (BSP) functions provided by the manufacturer to access hardware resources. There is no abstraction layer from hardware architecture. All the configurations and the distribution of the tasks must be programmed. Therefore, the programmer has the control of each function.

As it was stated in Section 2.1, it was required to implement a user *wrap code* to handling traps produced by cache parity errors on Compute Cluster running *NodeOS*. Since cache memories are not accessible directly by user, the only way to test the effectiveness of the implemented *wrap code* is by exposing the system to radiation to produce parity errors. Unfortunately, from the results it can be seen that the default *trap-handling-code* was not overwritten. Then, when a parity error in cache memories is produced, the system halts producing an *exception*. Consequently, the NMR-MPar scenarios will have more *exceptions* than the scenarios without redundancy because of their larger sensitive area (four times). In spite of the impossibility to change the *trap-code*, authors have decided to enable cache memories because applications with cache disable increases until six to ten times the execution time. This situation was not acceptable due to costs per hour and availability of the radiation facility.

3.1. Travel Salesman Problem Evaluation

For evaluating the SEU sensitivity of TSP application solving 16 cities, two radiation campaigns were performed. The first one was carried out to assess the NMR-MPar approach, and the second one to evaluate the application without redundancy. In the first campaign, three tests with a total fluence of $2.85 \times 10^8 \text{ n}\cdot\text{cm}^{-2}$ were executed giving a total of 18 executions. Results of this dynamic radiation campaign are summed-up in Table 4.

Table 4. Evaluation details of TSP NMR-MPar for 16 cities.

Test	Execution Number	Number of Instances with			IO Trap	Result
		Correct Answer	Trap	No Answer		
1	1	3	0	1	NO	Masked error
	2	3	0	1	NO	Masked error
	3	4	0	0	NO	Correct Answer
	4	3	1	0	NO	Masked error
	5	4	0	0	NO	Correct Answer
	6	4	0	0	NO	Correct Answer
	7	4	0	0	NO	Correct Answer
	8	4	0	0	NO	Correct Answer
2	1	3	0	1	NO	Masked error
	2	2	0	2	NO	Masked error
3	1	3	0	1	NO	Masked error
	2	3	0	1	NO	Masked error
	3	1	0	3	NO	Masked error
	4	0	1	3	NO	Exception
	5	0	1	3	NO	Exception
	6	3	0	1	NO	Masked error
	7	4	0	0	NO	Correct Answer
	8	4	0	0	NO	Correct Answer
TOTAL	18	52	3	17	0	

Results show that only in five executions, the four instances of the NMR-MPar produce a *correct answer*, nine executions *masked error* by the approach and 2 gives an *exception*. During this campaign, there were no *IO Traps*. A Trap is produced when a core halted in one of the Compute Cluster, giving as result that the referred CC halts and the concerning instance does not continues the execution producing an exception. For instance, if Process Element 2 of Compute Cluster 11 halts, it produces that CC 11 halts. As CC 11 is comprised by partition P2, thus the instance running in P2 gives an exception.

On the other hand, the second campaign that evaluates the application without redundancy, was composed of two tests with a total fluence of $1.85 \times 10^8 \text{ n}\cdot\text{cm}^{-2}$ running a total of 15 executions. Among them, one resulted in a *timeout* and three in *exceptions*. The others were *correct answers*. To obtain the SEE sensitivity in terms of cross-section, Equation (2) was used. Table 5 summarizes the obtained results for both radiation campaigns.

Table 5. SEE Sensitivity of the TSP solving a 16 cities problem.

TSP—16 cities	Correct Answer	Masked Error	Erroneous Result	Time Out	Exceptions	Total Errors
w/o redundancy	11	0	0	1	3	4
NMR-MPar	7	9	0	0	2	2

Similarly, for evaluating the TSP for 17 cities, other two radiation campaigns were performed. Results of the third campaign that evaluate the application implementing the NMR-MPar approach are presented in Table 6. It was exposed to a fluence of $1.15 \times 10^9 \text{ n}\cdot\text{cm}^{-2}$ comprises 15 executions. From the results, it is possible to observe the presence of two *IO Traps* that produce *timeouts* in all the running instances. There were no *correct answers* without using the approach, the six *correct answers* were provided by masking the error thanks to NMR-MPar.

Table 6. Evaluation details of TSP NMR-MPar for 17 cities.

Test	Execution Number	Number of Instances with			IO Trap	Result
		Correct Answer	Trap	No Answer		
1	1	1	0	3	NO	Masked error
	2	0	1	3	NO	Exception
2	1	2	0	2	NO	Masked error
	2	0	1	3	NO	Exception
	3	1	0	3	NO	Masked error
	4	1	0	3	NO	Masked error
3	1	2	1	1	NO	Masked error
	2	0	0	4	YES	Exception
	3	1	1	2	NO	Exception
	4	0	0	4	YES	Exception
	5	1	1	2	NO	Exception
	6	0	1	3	NO	Exception
	7	0	0	4	NO	Timeout
	8	2	1	1	NO	Masked error
	9	0	0	4	NO	Timeout
TOTAL	15	11	7	42	2	

A fourth radiation campaign was run to evaluate the application without redundancy. This campaign was composed of two tests with a total fluence of $1.02 \times 10^9 \text{ n}\cdot\text{cm}^{-2}$ running a total of 14 executions. In this scenario, one *application erroneous result* was produced. As explained, this is the worst consequence of a bit-flip since the system cannot detect the error producing an unpredictable situation. There were also five *timeouts*. Regarding *exceptions*, they were produced by one *IO Trap* and three *Traps* in Computing Clusters. Results for both scenarios solving the 17 cities problem are summarized in Table 7. The total number of errors groups the erroneous result, timeouts and exceptions produced during the tests.

Table 7. SEE Sensitivity of the TSP solving the 17 cities problem.

Scenario	Correct Answer	Masked Error	Erroneous Result	Time Out	Exceptions	Total Errors
w/o redundancy	4	0	1	5	4	10
NMR-MPar	0	6	0	2	7	9

3.2. Matrix Multiplication Evaluation

Two radiation campaigns were carried out to evaluate the MM application. The fifth campaign assesses the scenario implementing NMR-MPar. The total fluence was about $7.23 \times 10^8 \text{ n}\cdot\text{cm}^{-2}$. During the test, three instances result in *exceptions* and 21 in *timeouts*. From them, two *timeouts* occurred in the same execution. Also, one *timeout* and one *exception* occurred in the same execution. All the *exceptions* were produced by Compute Clusters since there were no *IO traps*. Although 22 executions were affected by errors, all of them were masked by the approach and the correct result was found. Finally, the sixth campaign received a total fluence of $6.53 \times 10^8 \text{ n}\cdot\text{cm}^{-2}$ running 83 executions of the MM application without redundancy. One of them produces an *application erroneous result*. There were also ten *timeouts* and four *exceptions*. All the *exceptions* in both scenarios were produced by *traps* in the Compute Clusters. Results of both campaigns are summarized in Table 8.

Table 8. SEE Sensitivity of the MM.

Scenario	Correct Answer	Masked Error	Erroneous Result	Time Out	Exceptions	Total Errors
w/o redundancy	62	0	1	10	4	15
NMR-MPar	34	22	0	0	0	0

4. Discussion

Soft error consequences of the six scenarios are illustrated in Figure 6. From the results, it is possible to observe a different behavior of the NMR-MPar depending on the application. For the MM, all the errors were *masked*, while for the TSP, there remain to be some *timeouts* and *exceptions* that cannot be masked. This can be explained by the fact that parity errors in cache memories were not masked by the approach since the memory was not invalidated. As it was aforementioned, the default code for *handling-trap-code* was not overwritten by the user-programmed *wrap-code*, so when a parity error is detected, the operating system produces a *timeout* or an *exception*. Since the execution time of TSP is larger than the MM, and the use of cache memories is different because of the nature of each application, TSP is more sensible to parity errors in cache memories. Results show more *exceptions* and *timeouts* for the TSP scenarios solving the 17 cities problem caused by the longer exposure time of the application which is around seven times the other applications (TSP-16 cities and MM).

Furthermore, an increase in the number of exceptions when using NMR-MPar in the TSP-17 cities is observed. This can be explained by both a longer exposure time and an increase in the sensitive zone, so the possibility of having parity errors in cache memories has increased. Therefore, in order to decrease the number of timeouts and exceptions of the application, it is necessary that the *handling-trap-code* considers the invalidation of cache memories when a parity error is detected. This politic of invalidation should be considered by manufacturers in proprietary systems or by programmers in open systems. In all the cases, by using the approach, the application reliability was increased. The most dangerous consequences “application erroneous result” were masked, while timeouts and exceptions were detected by the system. Timeouts were produced when an IO Trap was executed. The possibility of having IO Traps depends on the use of the Resource Manager cores of the IO Cluster. By minimizing its use, the system reliability could be improved.

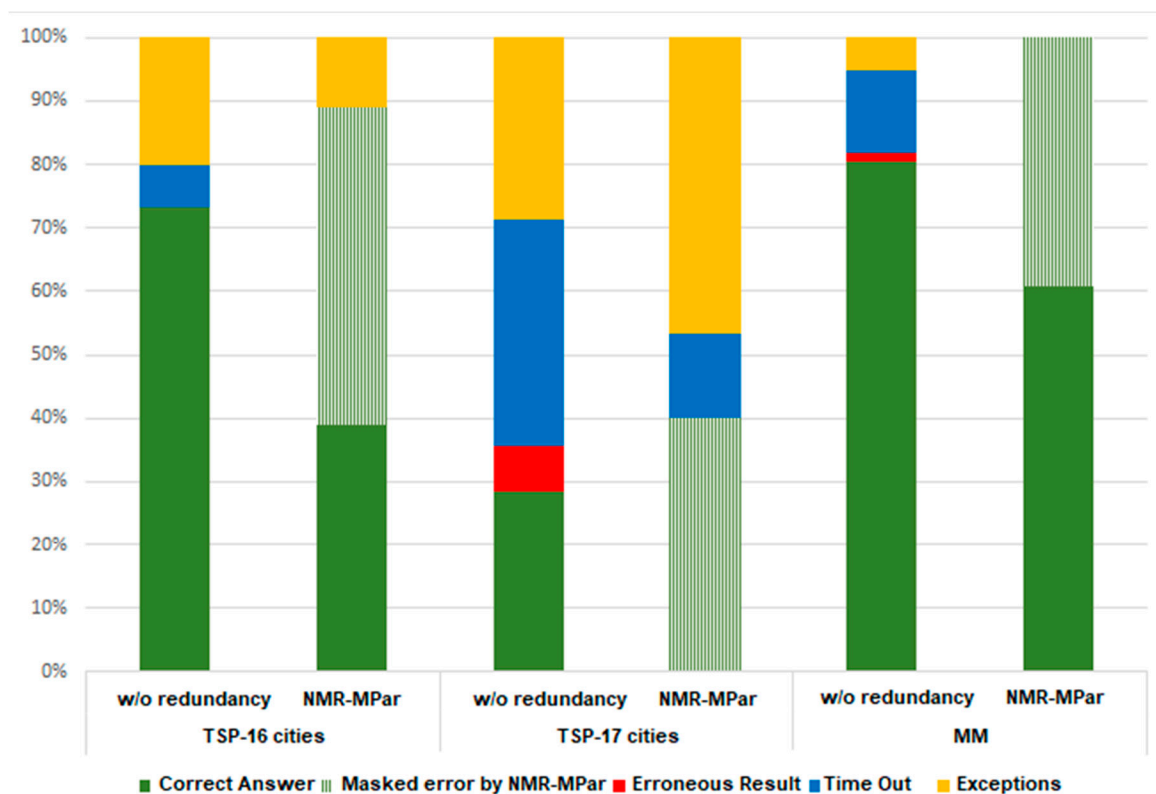


Figure 6. Comparison of the consequences of the evaluated applications.

To obtain the SEE sensitivity in terms of cross-sections, Equation (2) was applied. Two cases are proposed to define the number of events (Nev). Case 1 considers only *application erroneous result* and *timeouts* since *exceptions* can be easily handled by applying an adequate *handling-trap-code*. Case 2 considers *application erroneous results*, *timeouts* and *exceptions*.

Due to the scarcity of data, confidence intervals with a 95% level of uncertainty were considered to obtain the cross-sections. For events less than 100, the most accurate and universal way to calculate the confidence intervals uses the relationship between the cumulative distribution functions of the Poisson and chi squared distributions as described in [35]. Therefore, the following equation has been applied:

$$\frac{1}{2}\chi^2\left(\frac{\alpha}{2}, 2N_{err}\right) < \mu < \frac{1}{2}\chi^2\left(1 - \frac{\alpha}{2}, 2(N_{err} + 1)\right) \quad (5)$$

where $\chi^2(p, n)$ is the quantile function of the chi-square distribution with n degrees of freedom, α is a parameter that defines the $100(1 - \alpha)$ percent confidence interval, and N_{err} is the number of detected errors. Table 9 summarizes the results.

Table 9. Dynamic cross-section for the studied applications.

Application	Scenario	Fluence (n·cm ⁻²)	Case 1 (Nev)	Case 2 (Nev)	$\sigma_{Case 1}^{Case 1}$ (10 ⁻⁹ cm ² ·dev ⁻¹)	$\sigma_{Case 2}^{Case 2}$ (10 ⁻⁹ cm ² ·dev ⁻¹)
TSP-16 cities	w/o redundancy	1.85×10^8	1	4	[0.14–30.10]	[5.90–55.40]
	NMR-MPar	2.8×10^8	0	2	[0.00–13.2]	[0.87–25.80]
TSP-17 cities	w/o redundancy	1.02×10^9	6	10	[2.16–12.8]	[4.70–18.00]
	NMR-MPar	1.15×10^9	2	9	[0.21–6.28]	[3.58–14.90]
MM	w/o redundancy	6.53×10^8	11	15	[8.41–30.10]	[12.90–37.90]
	NMR-MPar	7.23×10^8	0	0	[0.00–5.10]	[0.00–5.10]

The failure rate of the studied devices can be classified within the DO-178B Software Considerations in Airborne Systems and Equipment Certification. The DO-178B is a guideline used as

de facto standard for developing avionic software systems [36]. Table 10 shows the admitted failure rate per hour for aircraft applications [37]. For instance, level A applications are related to critical functions required to safely fly and land aircraft.

Table 10. Failure condition levels according DO-178B.

Level	Failure Condition	Objectives	With Independence	Failure Rate
A	Catastrophic	66	25	$P \leq 10^{-9}/h$
B	Hazardous	65	14	$P \leq 10^{-7}/h$
C	Major	57	2	$P \leq 10^{-5}/h$
D	Minor	28	2	$P > 10^{-5}/h$
E	No Effect	2	0	N/A

For evaluating the reliability of a system, the concept of Failure in Time (FIT) is required. It considers the neutron flux in New York City ($13 \text{ n}\cdot\text{cm}^{-2}\cdot\text{h}^{-1}$). In addition, to assess the applicability of the case study in the avionics domain, the failure rate at avionics altitude per hour (FR/h) was calculated for each scenario. For this purpose, the dynamic cross sections were extrapolated to avionic altitude (35,000 feet) where the neutron flux is about $2.99 \times 10^3 \text{ n}\cdot\text{cm}^{-2}\cdot\text{h}^{-1}$. Table 11 summarizes the obtained results. Column “DO-178B” refers to the level of failure condition that could be used by the corresponding application.

Table 11. Failure-rate of the evaluated scenarios.

Application	Scenario	CASE 1			CASE 2		
		FIT	FR/h $\times 10^{-5}$	DO-178B	FIT	FR/h $\times 10^{-5}$	DO-178B
TSP-16 cities	w/o redundancy	[1.8–391.3]	[0.04–9.01]	D/E	[76.7–720.2]	[1.77–16.6]	D/E
	NMR-MPar	[0.0–171.6]	[0.0–3.95]	All levels	[11.3–335.4]	[0.26–7.72]	D/E
TSP-17 cities	w/o redundancy	[28.1–166.4]	[0.65–3.83]	D/E	[61.1–234.0]	[1.41–5.39]	D/E
	NMR-MPar	[2.7–81.6]	[0.06–1.88]	C/D/E	[46.5–193.7]	[1.07–4.46]	D/E
MM	w/o redundancy	[109.3–391.3]	[2.52–9.01]	D/E	[167.7–492.7]	[3.86–11.3]	D/E
	NMR-MPar	[0.0–66.3]	[0.0–1.53]	All levels	[0.0–66.3]	[0.0–1.53]	All levels

From results, it is possible to observe that by applying the approach for the MM application, there is a considerable improvement for both cases. Hence, this application can be used inclusively at Level A. In the case of TSP for achieving better reliability, it is necessary to improve the handling of exceptions.

5. Conclusions

The NMR-MPar approach has been demonstrated to be effective to improve the reliability of a distributed application running on a many-core processor exposed to neutron radiation. For reducing the number of exceptions, it is possible to use a complementary Software Implemented Fault Tolerance (SIFT) techniques. The present work is an important step for the validation of NMR-MPar as a fault tolerance approach for multi/many-core processors intended to be used in avionic and spacecraft domains.

Previous radiation experiences with a bare-metal system configuration on the same target device have demonstrated that by enabling the cache memories, it is possible to increase the performance of the application without compromising the reliability of the device, since cache memories implement an effective parity protection [24]. The present work shows that this affirmation could be supported with an adequate handling trap of parity errors. The use of proprietary operating systems has been a significant constraint. Therefore, it is desirable the use of open source code.

On the other side, previous work also shows that non-correctable errors were originated in GPRs, since registers do not implement any protection mechanism [24]. The implementation of NMR-MPar has overcome this issue by masking this type of errors. These results encourage the use of COTS many-core processors for applications running on harsh radiation environment.

Regarding future directions, this work opens up several possibilities to continue the research topic. First, the implementation of the NMR-MPar on other platforms and system configurations to evaluate its broad applicability on systems based on multi/many-core processors. Second, the evaluation of the approach under heavy-ions radiation to validate its applicability to spacecraft applications. Third, the combination of this approach with other SIFT techniques such as compiler techniques and replication of instructions, registers and cache-lines to reduces exceptions.

Author Contributions: Conceptualization, V.V. and R.V.; Data curation, V.V. and P.R.; Formal analysis, V.V. and P.R.; Investigation, V.V.; Methodology, V.V.; Resources, R.V.; Validation, V.V. and P.R.; Writing—original draft, V.V. and P.R.; Writing—review and editing, R.V.

Funding: This work has been supported by the Ecuadorian government through the Secretaría de Educación Superior, Ciencia, Tecnología e Innovación (SENESCYT) and the Universidad de las Fuerzas Armadas ESPE, grant PIC-2017-EXT-004. Radiation experiments on GENEPi2 at LPSC (CNRS/UGA) has been performed within the “Characterisation Program” of the IRT nanoelec, co-funded by the French government in the frame of the so called “Programme d’Investissements d’Avenir” under the reference ANR-10-AIRT-05.

Acknowledgments: Authors thank Jean-Francois Méhaut from LIG laboratory for his valuable contribution during the proposal of the fault-tolerance approach, Maud Baylac, Francesca Villa and Solene Rey from the LPSC laboratory for their long-lasting cooperation in the execution of the radiation tests experiments, and Stéphane Gailhard from the Societé Kalray for his cooperation to solve the MPPA programming issues.

Conflicts of Interest: The authors declare no conflict of interest. The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

References

1. Gaillard, R. Single Event Effects: Mechanisms and Classification. In *Soft Errors in Modern Electronic Systems*; Springer: New York, NY, USA, 2011.
2. Mittal, S.; Vetter, J.S. A survey of techniques for modeling and improving reliability of computing systems. *IEEE Trans. Parallel Distrib. Syst.* **2016**, *27*, 1226–1238. [[CrossRef](#)]
3. Vijaykumar, T.; Pomeranz, I.; Cheng, K. Transient Fault Recovery using Simultaneous Multithreading. In Proceedings of the 29th Annual International Symposium on Computer Architecture, Anchorage, AK, USA, 25–29 May 2002.
4. Basile, C.; Kalbarczyk, Z.; Iyer, R.K. Active Replication of Multithreaded Applications. *IEEE Trans. Parallel Distrib. Syst.* **2006**, *17*, 448–465. [[CrossRef](#)]
5. Mushtaq, H.; Al-Ars, Z.; Bertels, K. Efficient Software-Based Fault Tolerance Approach on Multicore Platforms. In Proceedings of the Design, Automation & Test in Europe Conference, Grenoble, France, 18–22 March 2013.
6. Reinhardt, S.; Mukherjee, S. Transient fault detection via simultaneous multithreading. In Proceedings of the 27th International Symposium on Computer Architecture (IEEE Cat. No. RS00201), Vancouver, BC, Canada, 14 June 2000.
7. Mukherjee, S.; Kontz, M.; Reinhardt, S. Detailed design and evaluation of redundant multi-threading alternatives. In Proceedings of the 29th Annual International Symposium on Computer Architecture, Anchorage, AK, USA, 25–29 May 2002.
8. Shye, A.; Blomstedt, J.; Moseley, T.; Reddi, V.J.; Connors, D. PLR: A Software Approach to Transient Fault Tolerance for Multicore Architectures. *IEEE Trans. Dependable Secure Comput.* **2009**, *6*, 135–148. [[CrossRef](#)]
9. Bolchini, C.; Miele, A.; Sciuto, D. An adaptive approach for online fault management in many-core architectures. In Proceedings of the 2012 Design, Automation Testing Europe Conference Exhibition (DATE), Dresden, Germany, 12–16 March 2012.
10. Alhakeem, M.; Munk, P.; Lisicki, R.; Parzyjegl, H.; Parzyjegl, H.; Muehl, G. Framework for Adaptive Software-Based Reliability in COTS Many-Core Processors. In Proceedings of the ARCS2015—The 28th International Conference on Architecture of Computing Systems, Porto, Portugal, 24–27 March 2015.
11. Kim, E.; Shanbhag, N. Soft N-Modular Redundancy. *IEEE Trans. Comput.* **2012**, *61*, 323–336. [[CrossRef](#)]
12. Walters, J.; Kost, R.; Singh, K.; Suh, J.; Crago, S. Software-based fault tolerance for the Maestro many-core processor. In Proceedings of the 2011 Aerospace Conference, Big Sky, MT, USA, 5–12 March 2011.

13. Löfwenmark, A.; Nadjm-Tehrani, S. Challenges in Future Avionic Systems on Multi-Core Platforms. In Proceedings of the IEEE International Symposium on Software Reliability Engineering Workshops, Naples, Italy, 3–6 November 2014.
14. Mollison, M.; Erickson, J.; Anderson, J.; Baruah, S.; Scoredos, J. Mixed-Criticality Real-Time Scheduling for Multicore Systems. In Proceedings of the 10th IEEE International Conference on Computer and Information Technology, Bradford, UK, 29 June–1 July 2010.
15. Panic, M.; Quiñones, E.; Zavkov, P.; Hernandez, C.; Abella, J.; Cazorla, F. Parallel many-core avionics systems. In Proceedings of the International Conference on Embedded Software (EMSOFT), Jaypee Greens, India, 12–17 October 2014.
16. Trujillo, S.; Crespo, A.; Alonso, A.; Pérez, J. MultiPARTES: Multi-core partitioning and virtualization for easing the certification of mixed-criticality systems. *Microprocess. Microsyst.* **2014**, *38*, 921–932. [[CrossRef](#)]
17. Galizzi, J.; Pignol, M.; Masmano, M.; Munoz, M.; Coronel, J.; Parrain, T.; Combettes, P. Temporal Duplex-Triplex on COTS processors with XtratuM. In Proceedings of the 2016 Eurospace DASIA Conference, Tallinn, Estonia, 10–12 May 2016.
18. Vargas, V.; Ramos, P.; Méhaut, J.; Velazco, R. NMR-MPar: A fault-tolerance approach for multi-core and many-core processors. *Appl. Sci.* **2018**, *8*, 465. [[CrossRef](#)]
19. Peronnard, P.; Velazco, R.; Hubert, G. Real-Life SEU Experiments on 90 nm SRAMs in Atmospheric Environment: Measures Versus Predictions Done by Means of MUSCASEP3 Platform. *IEEE Trans. Nuclear Sci.* **2009**, *56*, 3450–3455. [[CrossRef](#)]
20. Guertin, S. *Initial SEE Test of Maestro*; Jet Propulsion Laboratory, National Aeronautics and Space Administration: Pasadena, CA, USA, 2012.
21. Stolt, S.; Normand, E. A Multicore Server SEE Cross Section Model. *IEEE Trans. Nuclear Sci.* **2012**, *59*, 2803–2810. [[CrossRef](#)]
22. Oliveira, D.; Rech, P.; Quinn, H.; Fairbanks, T.; Monroe, L.; Michalak, S.; Anderson-Cook, C.; Navaux, P.; Carro, L. Modern GPUs Radiation Sensitivity Evaluation and Mitigation Through Duplication with Comparison. *IEEE Trans. Nuclear Sci.* **2014**, *61*, 3115–3122. [[CrossRef](#)]
23. Ramos, P.; Vargas, V.; Baylac, M.; Villa, F.; Rey, S.; Clemente, J.; Zergainoh, N.; Méhaut, J.; Velazco, R. Evaluating the SEE sensitivity of a 45 nm SOI Multi-core Processor due to 14 MeV Neutrons. *IEEE Trans. Nuclear Sci.* **2016**, *63*, 2193–2200. [[CrossRef](#)]
24. Vargas, V.; Ramos, P.; Ray, V.; Jalier, C.; Stevens, R.; de Dinechin, B.D.; Baylac, M.; Villa, F.; Rey, S.; Zergainoh, N.E.; et al. Radiation Experiments on a 28 nm Single-Chip Manycore Processor and SEU error-rate prediction. *IEEE Trans. Nuclear Sci.* **2017**, *64*, 483–490. [[CrossRef](#)]
25. Vargas, V. Approche Logicielle pour Améliorer la Fiabilité D’applications Parallèles Implémentées sur des Processeurs Multi-Cœur et Many-Cœur. Available online: <https://tel.archives-ouvertes.fr/tel-01693054/> (accessed on 28 April 2017).
26. Uznanski, S.; Alia, R.G.; Blackmore, E.; Brugger, M.; Gaillard, R.; Mekki, J.; Todd, B.; Trinczek, M.; Villanueva, A.V. The Effect of Proton Energy on SEU Cross Section of a 16 Mbit TFT PMOS SRAM with DRAM Capacitors. *IEEE Trans. Nuclear Sci.* **2014**, *61*, 3074–3079. [[CrossRef](#)]
27. Baggio, J.; Lambert, D.; Ferlet-Cavrois, V.; Paillet, P.; Marcandella, C.; Duhamel, O. Single event up sets induced by 1–10MeV neutrons in static-RAMs using mono-energetic neutron sources. *IEEE Trans. Nuclear Sci.* **2007**, *54*, 2149–2155. [[CrossRef](#)]
28. Miller, F.; Weulersse, C.; Carrière, T.; Guibbaud, N.; Morand, S.; Gaillard, R. Investigation of 14 MeV Neutron Capabilities for SEU Hardness Evaluation. *IEEE Trans. Nuclear Sci.* **2016**, *4*, 2789–2796. [[CrossRef](#)]
29. Saidi, S.; Ernst, R.; Uhrig, S.; Theiling, H.; de Dinechin, B.D. The shift to multicores in real-time and safety-critical systems. In Proceedings of the Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS), Amsterdam, The Netherlands, 4–9 October 2015.
30. KALRAY; TIMA; MINATEC; AIRBUS; AVIATION; DASSAULT; INRIA. CAPACITES Calcul Parallèle pour Applications Critiques en Temps et Sécurité “Investissement d’Avenir” Program. Available online: <http://capacites.minalogic.net/en/> (accessed on 28 February 2018).
31. KALRAY. *MPPA-256 Boostan Cluster and I/O Subsystem Architecture*; KALRAY: Montbonnot, France, 2016.
32. Applegate, D.; Bixby, R.; Chvatal, V.; Cook, W. *The Traveling Salesman Problem: A Computational Study*; Princeton Series in Applied Mathematics; Princeton University Press: Princeton, NY, USA, 2007; pp. 49–53.

33. Francesquini, E.; Castro, M.; Penna, P.; Dupros, F.; Freitas, H.; Navaux, P.; Méhaut, J. On the energy efficiency and performance of irregular application executions on multicore, NUMA and manycore platforms. *J. Parallel Distrib. Comput.* **2015**, *76*, 32–48. [CrossRef]
34. JESD89A, Measurement and Reporting of Alpha Particle and Terrestrial Cosmic Ray-Induced Soft Errors in Semiconductor Devices. JEDEC Solid State Technology Association. Available online: <http://www.jedec.org/sites/default/files/docs/jesd89a.pdf> (accessed on 23 December 2017).
35. Autran, J.; Munteanu, D.; Roche, P.; Gasiot, G. Real-time soft-error rate measurements: A review. In *Microelectronics Reliability*; Elsevier: Oxford, UK, 2014; pp. 1455–1476.
36. Ferrel, T.; Ferrel, D. RTCA DO-178B/EUROCAE ED-12B. In *Digital Avionics Handbook*, 3rd ed.; CRC Press: Boca Raton, FL, USA, 2014; pp. 195–206.
37. Hilderaman, V.; Baghai, T. *Avionics Certification: A Complete Guide to DO-178B (SW), DO-178C (Update), DO-254(HW)*; Avionics Communication Inc.: Leesburg, VA, USA, 2013.



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).