Evaluating the SEE Sensitivity of a 45 nm SOI Multi-Core Processor Due to 14 MeV Neutrons

Pablo Ramos, Vanessa Vargas, Maud Baylac, Francesca Villa, Solenne Rey, Juan Antonio Clemente, Nacer-Eddine Zergainoh, Jean-Francois Méhaut, and Raoul Velazco

***Abstract*—The aim of this work is to evaluate the SEE sensitivity of a multi-core processor having implemented ECC and parity in their cache memories. Two different application scenarios are studied. The first one configures the multi-core in Asymmetric Multi-Processing mode running a memory-bound application, whereas the second one uses the Symmetric Multi-Processing mode running a CPU-bound application. The experiments were validated through radiation ground testing performed with 14 MeV neutrons on the Freescale P2041 multi-core manufactured in 45 nm SOI technology. A deep analysis of the observed errors in cache memories was carried-out in order to reveal vulnerabilities in the cache protection mechanisms. Critical zones like tag addresses were affected during the experiments. In addition, the results show that the sensitivity strongly depends on the application and the multi-processing mode used.**

***Index Terms*—Accelerated testing, AMP, multi-core, SEE, SEFI, SEU, SMP, soft error, SOI.**

I. Introduction

AVIONICS and spacecraft applications require determinism and robustness in their reactive embedded systems. The current technological trend in embedded systems is the use of multi-core processors in order to satisfy the growing demand of performance and reliability without a critical increase of power consumption. The inherent redundancy capability of multi-core architectures makes them ideal for implementing fault-tolerant mechanisms [1]. Moreover, these devices provide a great flexibility because they allow implementing different multi-processing modes and programming paradigms. Hence, avionics and spacecraft industries are interested in validating the use of multi-core and many-core devices for their applications [2], [3].

The continuous technology scaling in integrated circuits makes them more sensitive to the effects of natural radiation such as Single Event Effects (SEEs) [4]. For this reason, physical designers are continuously searching for new methods to improve manufacturing technologies to reduce SEE consequences. For instance, Silicon-On-Insulator (SOI) technology has been proved to be less sensitive than CMOS bulk technology [5].

On the other side, additional hardware implementations have been added to multi-core architectures for improving their reliability. Examples of these protection mechanisms are the implementation of Error Correcting Code (ECC) and parity in cache memories. Hamming codes are very useful to mitigate Single Event Upsets (SEUs) since they can detect double errors and correct single ones. Nevertheless, new hardware introduces an extra area with the corresponding increase in power comsumption and performance degradation [6].

The use of cache memories reduces the memory access time, thereby increasing substantially the performance of the system. However, enabling caches implies the increase of sensitive area and thus, the reduction of system reliability. Therefore, a trade-off between performance and reliability is needed depending on the application.

Given the significant importance that represents the use of multi-core and many-core processors for avionics and safety-critical applications, it is mandatory to evaluate their sensitivity to SEEs, and particularly to SEUs. The present work aims at evaluating the sensitivity to neutron radiation of a 45 nm SOI multi-core processor. Two main contributions are presented: The first one is the determination of the static cross section of a device implementing Error Correcting Code (ECC) and parity (that cannot be deactivated) in their caches. The second one is the evaluation of the neutron radiation sensitivity of the studied multi-core working in two different multi-processing modes running parallel aplications.

For achieving the mentioned contributions, static and dynamic tests were carried out with a neutron beam of 14 MeV to obtain the corresponding cross-sections. Concerning the dynamic tests, two operating modes were explored. The Asymmetric Multi-Processing Mode (AMP) without operating system (OS) was used in order to test specific sensitive hardware resources, such as cache memories and registers. On

Manuscript received September 29, 2015; revised December 23, 2015, February 05, 2016, and February 17, 2016; accepted February 29, 2016. Date of publication July 12, 2016; date of current version August 17, 2016. This work was supported in part by e2v, a Freescale partner, http://ww.e2v.com, by the Secretaría de Educación Superior Ciencia Tecnología e Innovación del Ecuador (SENESCYT) TIN2013-40968-P, by the Spanish Ministry of Education, Culture and Sports project TIN2013-40968-P, and by the "José Castillejo" mobility grant for professors and researchers.

P. Ramos and V. Vargas are with the Université Grenoble-Alpes & TIMA Labs, 38031 Grenoble, France, and also with the Universidad de las Fuerzas Armadas ESPE, DEEE, Sangolqui, Ecuador (e-mail: pframos@espe.edu.ec; vcvargas@espe.edu.ec).

M. Baylac, F. Villa, and S. Rey are with Laboratoire de Physique Subatomique et de Cosmologie LPSC, Université Grenoble-Alpes & CNRS/IN2P3, 38031 Grenoble, France (e-mail: maud.baylac@lpsc.in2p3.fr; francesca.villa@lpsc.in2p3.fr; solenne.rey@lpsc.in2p3.fr).

J. A. Clemente is with the Computer Architecture Department, Facultad de Informática, Universidad Complutense de Madrid (UCM), 28040 Madrid, Spain (e-mail: ja.clemente@fdi.ucm.es).

N.-E. Zergainoh and R. Velazco are with TIMA Labs, (Université Grenoble-Alpes & CNRS respectively), 38031 Grenoble, France (e-mail: nacer-eddine.zergainoh@imag.fr; raoul.velazco@imag.fr).

J.-F. Méhaut is with LIG Labs, Université Grenoble-Alpes & CNRS, 38031 Grenoble, France (e-mail: jean-francois.mehaut@imag.fr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNS.2016.2537643

the other hand, the Symmetric Multi-Processing Mode (SMP) was implemented to test the resources used by the embedded Linux SDK V1.6 OS. Experimental results show the relationship between the system reliability and the multi-processing mode used.

A preliminary version of this work was presented in [7]. The present work provides a detailed description of the adopted approach, the results regarding the errors not detected by the studied multi-core architecture, and a deep analysis of the experimental results including the probable causes of the observed errors.

The remainder of the paper is organized as follows: Section II presents the related work. Section III describes the adopted approach that has been used to evaluate the intrinsic sensitivity of the multi-core and its dynamic response. Section IV details the experimental setup. Section V presents and analyzes the results issued from neutron ground testing. Finally, Section VI concludes the paper and provides some directions for future work.

II. Related Work

Several interesting works dealing with the sensitivity of electronic components can be found in the literature. Reference [8] summarizes the sensitivity to SEEs induced by neutrons of different integrated circuits (i.e., SRAMs, microprocessors and FPGAs) applicable to avionics. However, there are very few works available regarding multi-core and many-core processors sensitivity.

In [9] is presented a significant work that establishes a dynamic cross-section model for a multi-core server based on quad-core processors in 45 nm bulk CMOS technology. Also, it provides a fault handling comparison between Windows 5.2 and Linux 5.1 operating systems.

Reference [10] presents the radiation sensitivity evaluation of a modern Graphic Processing Units (GPUs) designed in 28 nm technology node, and composed by an array of streaming multiprocessors which share the L2 cache memory. It also provides a hardening strategy based on Duplication with Comparison.

Authors of [11] propose to disable the cache memories of high-end processsors in safety-critical applications in order to gain in reliability in spite of the increase of the execution time. An accurate analysis of the effects of soft errors in the instruction and data caches is also presented.

The work presented in [12] demonstrates that, by enabling L1 cache, it is possible to improve the performance of the system without compromising the reliability. A generic metric (Mean Workload Between Failures) taking into account both cross section and exposure time was introduced to evaluate the reliability of a embedded processor devoted to execute safe-critical applications.

III. Adopted Approach

A. Static Sensitivity

To estimate the intrinsic sensitivity of the accessible memory cells of the multi-core processor, the neutron static cross-section σ_{STATIC} was obtained. Typically, the method used to obtain σ_{STATIC} consists in writing a predefined pattern in the memory locations and registers, and checking it along the radiation experiments to detect errors.

However, since the cache memories on the target device implement protection mechanisms that cannot be deactivated, this method is not suitable as it is. This can be explained due to the fact that single-bit errors in a word are not visible while reading memory locations because whenever they occur, they are corrected by either the ECC or the cache invalidation mechanisms. It was thus necessary to use a complementary technique based on *machine-check error report* for logging data that have been corrupted during the radiation experiments.

In processors including *machine-check error report*, it is possible to enable an interrupt routine for reporting errors. The information about the errors is saved in some special-purpose registers of the device. By reading these registers, one can know the type of error ocurred, address, data, as well as the obtained and calculated ECCs.

For this test, the multi-core processor was configured in AMP mode without OS in order to have independence in the execution of each core when performing the self-testing of their cache memories. In addition, the L1, L2 and L3 caches, as well as the *machine-check error report* of each core must be enabled. In order to simplify the interpretation of the results due to cache coherence mechanisms, the self-testing application was configured so that each core reads from and writes to different sections of the main memory. Each section has the same size as the L2 cache. In the particular case of the L3 cache, only the core 0 was configured to use it, preventing other cores to access it.

B. Dynamic Response

For evaluating the reliability of the target device when an application is running, the dynamic cross section (σ_{DYN}) has been obtained. The motivation is to observe to what extent the dynamic chip response depends on the application and the multi-processing paradigm implemented. Thus, the SMP and AMP multi-processing modes were both adopted in this experiment.

In SMP mode, a single OS that runs on all the cores is responsible for achieving parallelism in the application. It dynamically distributes the tasks among the cores, manages the organization of task completion, and controls the shared resources. In AMP mode, the cores run independently of each other, with or without OS. Also, they have their own private memory space, although there is a common infrastructure for inter-core communications. Hence, AMP mode is very useful when working with embedded systems [13]. Fig. 1 depicts these two Multi-Processing modes.

Concerning dynamic tests, two different scenarios were considered. On the one hand, a memory-bound application was implemented when the processor operates in AMP mode without OS to evaluate the sensitivity of memory resources. On the other hand, a CPU-bound application was implemented when the processor operates in SMP mode in order to maximize the use of CPU resources and scheduling. In both cases, errors detected by the application and by the *machine check-error report* were considered in order to evaluate the sensitivity of the

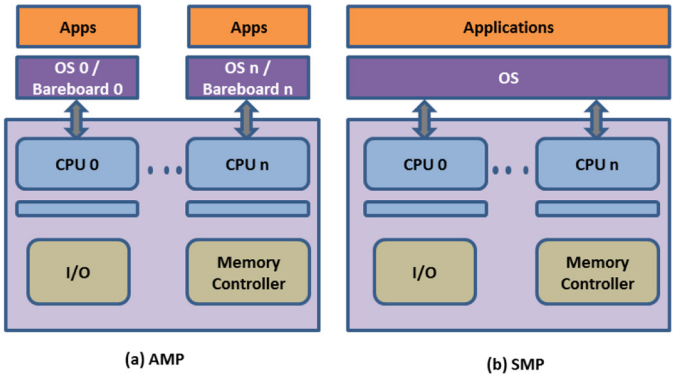

Fig. 1. Schemes of AMP and SMP processing modes.

target device. In the SMP scenario, it was necessary to modify the original *traps* code in the kernel and in the u-boot of the Linux OS. The *traps* code is the code that is executed by the system when an exception or a fault occurs. In this approach, this section of code was modified to log all the events detected by the *machine-check error report*. Also, in the case of a L2 cache error detection, the L2 error registers values were logged to obtain more details about the error. It is important to note that, the original *traps* code logs the recoverable and unrecoverable conditions that cause a machine-check exception. If the condition is recoverable by the machine check, then it returns to the previous state and its operation is resumed.

IV. Experimental Setup

A. Neutron Radiation Facility

The radiation ground tests were conducted at the GENEPI2 (*GEnerator of NEutron Pulsed and Intense*) facility located at the LPSC (*Laboratoire de Physique Subatomique et Cosmologie*) in Grenoble, France [14]. This accelerator was originally developed for nuclear physics experiments, and recently it has been used to irradiate integrated circuits from different technologies.

From the target, neutrons are emitted in all directions. The Device Under Test (DUT) is set facing directly the target at a distance determined to adjust the neutron flux. While the DUT is fully exposed to neutrons, a dedicated neutron shielding can be used to protect the readout electronic platform.

Neutrons are produced with an average energy of 14 MeV. For the radiation campaigns, it was considered, to first approximation, that only neutrons emitted fully forward will impact the DUT. In this case, the neutron energy is maximal at 15 MeV. Reference [15] discusses the relevance of using 14 MeV neutron test to characterize the SEU sensitivity of digital devices.

Neutron production is monitored throughout the experiments to determine the neutron dose for each irradiation. An online Si detector, located within the beam pipe 1 meter upstream of the target, collects the recoil particles backscattered from the target during the fusion reaction.

Early 2015, a fresh T target was installed, generating a maximum neutron flux of $4.5 \times 10^7 n \cdot cm^{-2} \cdot s^{-1}$. For the radiation tests presented in this work, the flux was limited to $2 \times 10^5 n \cdot cm^{-2} \cdot s^{-1}$.

B. Device Under Test

The target device was a Freescale P2041 multi-core processor which is inside the P2041RDB design board [16]. The multi-core is based on four e500mc cores built on Power Architectures technology and manufactured in 45 nm SOI technology. This quad-core can operate up to 1.5 GHz and includes a three-level cache hierarchy. The e500mc core is a 32-bit superscalar processor that includes independent on-chip 32 KB L1 caches for instruction and data, and a unified 128 KB backside L2 cache. Additionally, the P2041 includes a 1024 KB L3 cache shared among the four cores. Table I gives details about the sensitive areas of the multi-core processor that were targeted during the radiation campaigns.

The e500mc processor implements an L1 instruction and data cache with automatic cache invalidation when a parity error is detected. Both the L2 backside cache and the L3 shared frontside cache are protected with configurable ECC or parity for the data array, and parity for the tag array. This architecture corrects single bit errors and detects multiple-bit ones [17]. Fig. 2 depicts the memory architecture of the studied multi-core processor.

For the experimental tests, the cores were configured in *write shadow mode*. In this mode, all modified data in the L1 cache is written through into the L2 cache. This ensures that, if data or parity tags are corrupted in the L1 cache, it can be invalidated and repopulated with the valid data from the rest of the memory hierarchy [17]. For logging all the SEEs occurred during the radiation experiments, the machine-check error interrupt and the *Cache Error Checking* bits were enabled.

C. Tested Applications

In the first scenario, for evaluating the dynamic response of the studied multi-core processor, a memory-bound 80×80 Matrix Multiplication (MM) algorithm was implemented. In this case, the device was configured in AMP mode, where each core executes independently the same matrix multiplication ($C = A \times B$) and compares its results with a predefined value in order to identify errors. The size of the matrix was selected in order to maintain a trade-off between the amount of memory used and the execution time. The matrices A, B and C were located in consecutive memory vectors. Matrix A was filled up with 1's and B was filled up with 2's, thus the expected result was 160 for all the elements of matrix C. The matrices were filled up with fixed values in order to simplify the data analysis since a known value helps to identify which bit or bits have been changed during the test. In this way, MBUs (Multiple Bit Upsets) and MCUs (Multiple Cell Upsets) can be easily detected. It is important to note that the results of the experiment are totally independent of the input values, no matter the particle produces a bit flip in a fixed or random value.

In the second scenario, the CPU-bound application *Travelling Salesman Problem* (TSP) for 16 cities was implemented. Its execution was distributed among all the cores. The multi-core processor was configured in SMP mode in which the OS manages the resources in order to maximize the processing capacity of the cores. The parallel implementation of

TABLE I
SENSITIVE AREAS OF THE P2041 MULTI-CORE PROCESSOR

Sensitive zone	*Location*	*Capacity*	*Description*
L1	Cores 0, 1, 2, 3	32 KB / D and 32 KB / I per core	Data / Instruction Cache
L2	Cores 0, 1, 2, 3	128 KB per core	Backside Unified Cache
L3	Multi-core	1024 KB per chip	Frontside cache
GPR	Cores 0, 1, 2, 3	32 registers of 32 bits	General purpose register
FPR	Cores 0, 1, 2, 3	32 registers of 64 bits	Floating point register

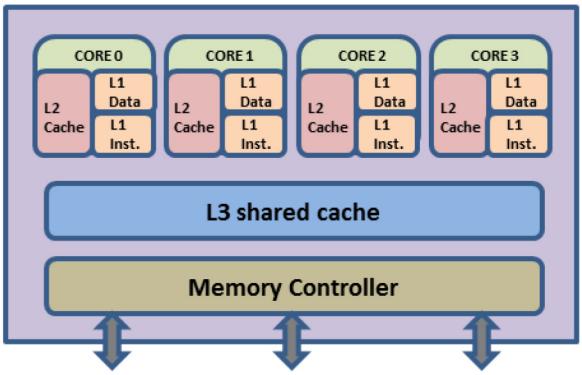

Fig. 2. Scheme of the memory architecture of the multi-core processor.

the *TSP* application makes this benchmark intrinsically fault-tolerant since, if one core is stopped by any reason, another core could find the correct result. The source code of this parallel implementation was an adapted version of the one used in [18]. An application based on Linux *PTRACE-Process trace* functions was implemented to monitor parallel applications and their related processes.

V. EXPERIMENTAL RESULTS

A. Static Cross Section

A first radiation campaign was carried out for obtaining the multi-core static cross-section. The device under test was placed facing the center of the target perpendicularly to the beam axis at a distance of $\sim 19.1\ cm$. The neutron beam energy was $14\ MeV$ with a flux of $\sim 1.96 \times 10^5 n \cdot cm^{-2} \cdot s^{-1}$ at a $500\ Hz$ frequency. The cross section is defined as:

$$\sigma = \frac{Number\ of\ Upsets}{Fluence} \tag{1}$$

58 SEEs were detected within 2 hours of exposure time. Among them, 46 were SEUs and 12 Single Event Functional Interrupts (SEFIs). There were no errors in general and floating point registers. Table II summarizes the results of this campaign.

In this experiment, all the SEEs were considered errors no matter they were detected by the *machine-check error report* or by the self-testing application. Then, the obtained static cross-section is:

$$\sigma_{STATIC} = \frac{58}{1.41 \times 10^9} = 4.11 \times 10^{-8} \frac{cm^2}{device} \tag{2}$$

Due to the scarcity of experimental data (58 SEEs), it is compulsory to add uncertainty margins to these results. For numerous events (typically > 100), the Poisson distribution can be used to calculate such margins. However, in this situation the most accurate and universal way to calculate the uncertainty margins consists in using the relationship between the cumulative distribution functions of the Poisson and chi-squared distributions as described in [19], [20]. Therefore, the following equation has been applied:

$$\frac{1}{2}\chi^2\left(\frac{\alpha}{2}, 2N_{err}\right) < \mu < \frac{1}{2}\chi^2\left(1-\frac{\alpha}{2}, 2(N_{err}+1)\right) \tag{3}$$

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

For a 95% confidence interval ($\alpha = 0.05$), the lower and upper limits for the static cross section are:

$$3.12 \times 10^{-8} \frac{cm^2}{device} < \sigma_{STATIC} < 5.32 \times 10^{-8} \frac{cm^2}{device} \tag{4}$$

Since the accessible registers and memory cells of the multi-core processor represent about 1.47×10^7 bits, the confidence interval for the static cross section per bit is estimated as $[2.12 - 3.62] \times 10^{-15} cm^2/bit$. Reference [9] provides the estimation of the bit cross section for a 45 nm CMOS technology processor ($1 \times 10^{-14} cm^2/bit$) for neutrons with the same energy. From these results, it can be seen that 45 nm SOI technology is between three and five times less sensitive to SEEs than its CMOS counterpart.

Errors in L1, L2 and L3 caches, both in data arrays and cache tags were detected by the *machine-check error report*. In addition, it was observed a SEFI (depicted in Table II as *"Other errors"*) that provoked a system hang simultaneously in all the cores. This event lead to several errors logged by the self-testing application running on the processors that showed data different from the original word ($0 \times 55AA55AA$) written in the memory. From these errors, two types of patterns were identified.

TABLE II
RESULTS OF THE STATIC RADIATION CAMPAIGN

SEE Type	*Type of error*	*Occurrences*	*Consequences*
SEU	L1 Data parity	9	None
SEU	L2 Single-bit ECC	29	None
SEFI	L2 Tag parity	5	Hang
SEU	Multiple L2 errors	1	None
SEU	L3 Single-bit ECC	7	None
SEFI	L3 Multiple-bit ECC	6	Hang
SEFI	Other errors	1	Hang
Total		58	

TABLE III
MAIN MEMORY SPACE AND DETAILS OF THE FIRST PATTERN OF ERRORS DURING THE STATIC TEST

Core	*Start Addr*	*End Addr*	*Range 1*	*Range 2*	*Range 3*
0	0x10000	0x30000	0x16548 - 0x1657c	0x14cc8 - 0x14cfc	0x16ac8 - 0x16afc
1	0x40000	0x60000	0x46048 - 0x4607c	0x463c8 - 0x463fc	0x46908 - 0x4693c
2	0x70000	0x90000	0x76048 - 0x7607c	0x76388 - 0x763bc	0x76908 - 0x7693c
3	0x100000	0x120000	0x106048 - 0x10607c	0x1063c8 - 0x1063fc	0x106908 - 0x10693c

TABLE IV
DETAILS OF SECOND PATTERN OF ERRORS DURING THE STATIC TEST

Core	*No. Ocurrence*	*1st Word Addr*	*2nd Word Addr*	*3rd Word Addr*	*4th Word Addr*
0	1	0x10000	0x10004	0x10008	0x10024
	2	0x16000	0x16004	0x16008	0x16024
	3	0x16200	0x16204	0x16208	0x16224
	4	0x16600	0x16604	0x16608	0x16624
1	5	0x42000	0x42004	0x42008	0x42024
	6	0x46380	0x46384	0x46388	0x463a4
	7	0x46440	0x46444	0x46448	0x46464
2	8	0x72000	0x72004	0x72008	0x72024
	9	0x763c0	0x763c4	0x763c8	0x763e4
	10	0x76440	0x76444	0x76448	0x76464
	11	0x7c000	0x7c004	0x7c008	0x7c024
3	12	0x102000	0x102004	0x102008	0x102024
	13	0x106380	0x106384	0x106388	0x1063a4
	14	0x106440	0x106444	0x106448	0x106464

The first one consists of a set of fourteen words with consecutive addresses containing $0 \times DEADBEEF$ as data. Table III presents the main memory space used by each core (Columns 2 and 3) and the address ranges where this pattern was replicated. The second pattern constitutes scattered clusters of errors of four words each. In each of them, the first word contained $0 \times DEADBEEF$, the second one 0×20200044, the third one 0×00130000 and the last one 0×00006000. Table IV summarizes the replications of this pattern, as well as the involved addresses.

Due to the fact that errors have occurred simultaneously and the observed pattern is repeated among the cores, it is presumed that a particle perturbed a shared resource of the chip. Because of the nature of these errors, it is suggested that the affected resource was a register belonging to the *CoreNet Coherency Fabric (CCF)*, which is the connectivity infrastructure of the multi-core system.

B. Sensitivity of the P2041 in AMP Scenario

A second radiation test campaign was carried out to obtain the dynamic cross section in AMP scenario (σ_{DYN_AMP}). The device was again set at a distance of $\sim 19.1\ cm$ from the target. The neutron energy was $14\ Mev$ with a flux of $\sim 1.96 \times 10^5 n \cdot cm^{-2} \cdot s^{-1}$ at a $500\ Hz$ frequency. Two tests, each one lasting 2 hours were performed.

Table V shows that L1, L2 and L3 caches were all perturbed by neutrons. The *Load Instruction* and *Instruction fetch* errors are the most critical ones since they provoked processor hang. Half of the observed L2 *Tag parity* errors lead to processor hang. L1 *Data cache parity* errors are not critical since L1 cache is invalidated when parity fails. Finally, L2 and L3 *Single-bit* errors are not critical as the ECC corrects them.

TABLE V
RESULTS OF RADIATION EXPERIMENTS FOR THE P2041 WORKING IN AMP SCENARIO

SEE Type	*Type of error*	*Test 1*	*Test 2*	*Consequences*
SEFI	Load Instruction	1	0	Hang
SEU	L1 Data parity	19	17	None
SEU	L2 Single-bit ECC	9	20	None
SEFI	L2 Tag parity	0	4	Hang
SEU		3	1	None
SEU	Multiple L2 errors	3	1	None
SEU	L3 Single-bit ECC	3	2	None
SEFI	Instruction fetch	0	1	Hang
MBU	Other errors	6	0	App. result error
Total		44	46	

Briefly, there were one SEFI in Test 1 and five SEFIs in Test 2 that caused system hangs. In addition, six events in Test 1 provoked errors in the results of the application, but they were not detected by the multi-core *machine-check error report*. This puts in evidence that errors were produced by Multiple Bit Upsets (MBUs) involving not only data, but also parity information. A deeper analysis has allowed to identify the origin and multiplicity of these events. Four of them were clusters of errors whereas the other two were single data errors.

1) Clusters of Errors: Three clusters of errors occurred in Core 2, and one in Core 1. All of them were very closely related and they were detected in the same read cycle. Each cluster involves exactly 16 consecutive positions of the resulting matrix. Each matrix element was an integer value (4 bytes). In all cases, an incorrect result of "2" was observed instead of the expected "160".

Considering that:

- The e500mc processor features a set associative L1 cache memory organized as 64 sets of 8 blocks with 64 bytes in each cache line.
- The L2 cache memory is organized as 256 sets of 8 blocks of 64-byte cache lines [16].
- The number of consecutive corrupted addresses exactly matches the size of the cache line in the processor architecture.
- The physical addresses involved in each cluster correspond to a cache block.

Then, it is clear that the cluster of errors was produced by an upset affecting the cache *address tag*. It could be explained as follows: Upon reading the involved addresses which have Line $TagT$ stored in Set S, the cache hardware retrieves incorrect data instead of fetching the correct values from the main memory because a tag belonging to this set S was corrupted and became that precise tag T. Fig. 3 depicts the clusters of errors observed in Core 2 assuming that the particle affected the L1 cache. Two of them had line tag $0 \times 403DD$ and the other one $0 \times 403DE$.

The persistence of 2's in these errors indicates that the cache had already been filled up with the contents of matrix B. Taking into account the data address mapping shown in Fig. 3(a), any

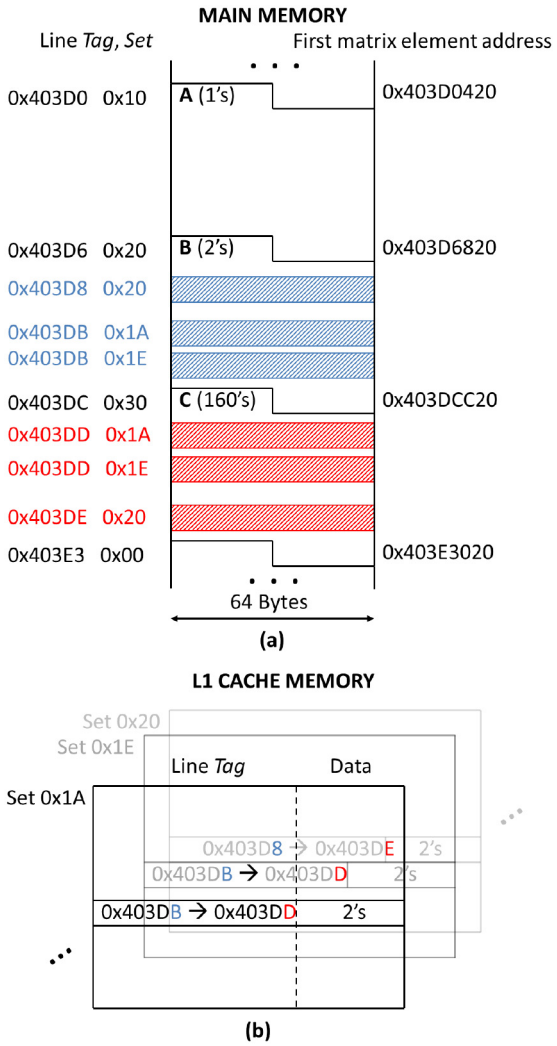

Fig. 3. Clusters of errors attributed to undetected tag errors. Note that in (a), the main memory is depicted by blocks of 64 Bytes. A main memory address (36 bits) comprises cache tag (24 bits), set (6 bits) and word position (6 bits). Also, note that the matrices A, B and C do not start exactly at the beginning of an address line that the cache refers to.

line tag comprised in the interval $(0 \times 403D6 - 0 \times 403DC)$ (matrix B) could have become the cluster error line tag. Comparing the tags of the clusters of errors with each one of the tags in the previous interval, it was possible to detect a MBU affecting bits b_1 and b_2 due to their physical adjacency. For the three cases the tags had to be changed (from $0 \times 403DB$ to $0 \times 403DD$ and from $0 \times 403D8$ to $0 \times 403DE$). These errors were not detected by the parity protection mechanisms since parity bit remains the same. Note that the L1 cache implements only one parity bit per tag.

Thus, in the authors' opinion, a particle modified two consecutive bits (MBU) belonging to three different tags (Multiple Cell Upset with multiplicity of three). Moreover, when decoding the corrupted addresses, it was possible to determine that the cache lines in Sets $0 \times 1A$, $0 \times 1E$ and 0×20 were affected. The fact that even and quasi-consecutive sets in cache were involved, gives clues about the possible 3-D implementation of the caches.

Finally, the cluster of errors observed in Core 1 appears in the line tag $0 \times 203DD$ set $0 \times 1B$. Following the previous analysis it is possible to verify that the particle has also changed the bits b_1 and b_2 of the line tag $0 \times 203DB$ set $0 \times 1B$ becoming the line tag $0 \times 203DD$. This perturbation in the cache was not detected since the parity remains the same. This cluster of errors may have been produced by a MBU, or it was probably related to the clusters of errors occured in Core 2 due to their similarities, in which case the mentioned MCU would have multiplicity of 4.

2) Single Errors: Two separated matrix-result data were corrupted from "160" to "162" in Core 2, at addresses $0 \times 403DF380$ and $0 \times 403DF480$ respectively. Since the same bit (b_1) was corrupted in both addresses and the difference between them is 0×100, it is very likely that they constitute an MCU. Also, this distance suggests that memory interleaving probably involves memory blocks of 256 addresses. These events were not detected by the parity protection which indicates that the parity bit was corrupted as well. Note that the L1 data cache implements one-bit-per-byte parity checking.

To conclude, the occurrences of application errors and hangs are evidences that the ECC and Tag parity mechanisms are not enough to guarantee the immunity of the cache memories.

The errors obtained during tests 1 and 2 described in Table V were added in order to have the total number of errors occurred within 4 hours of irradiation. The device was exposed to a fluence of $2.82 \times 10^9 n \cdot cm^{-2}$. In this experiment the total number of SEEs was 90 and, among them 12 produced errors (*application errors and hangs*). Applying Equations (1) and (3) for a confidence level of 0.95, the dynamic cross-section in AMP scenario without OS is:

$$2.17 \times 10^{-9} \frac{cm^2}{device} < \sigma_{DYN\_AMP} < 7.33 \times 10^{-9} \frac{cm^2}{device} \quad (5)$$

C. Sensitivity of the P2041 in SMP Scenario

Three additional radiation campaigns were carried out to calculate the dynamic cross section in SMP scenario (σ_{DYN_SMP}). In the first campaign the code was loaded from the NOR flash memory provided in the design board. During the test, the application definitely stopped due to a fatal crash in the OS after only 22 minutes. When a reboot of the system was performed, the image of the OS could not be loaded since the NOR flash memory was corrupted as well. Once, the OS image was restored, the test continues but a fatal crash in the OS occurred again after 28 minutes, giving a total test time of 50 minutes. That is the reason why for the other two campaigns the OS image was loaded from a hard disk. The second and the third campaign lasted one and four hours respectively. Table VI shows the characteristics of the radiation test campaigns.

Table VII summarizes the obtained results for the three tests. The fault classification was done based on the OS fault-handling messages and the monitor application. Faults with multiple indications were scored at the most critical level. The order of the rows in this table depends on the criticity of the fault, being the last one the most critical. It is important to

TABLE VI
TEST CAMPAIGNS CHARACTERISTICS FOR SMP SCENARIO

Test Campaign	*Flux* $[n \cdot cm^{-2} \cdot s^{-1}]$	*Time [min]*	*Fluence* $[n \cdot cm^{-2}]$
Test 1	$\sim 1.96 \times 10^5$	50	6.00×10^8
Test 2	$\sim 1.62 \times 10^5$	60	5.83×10^8
Test 3	$\sim 1.45 \times 10^5$	240	20.88×10^8

TABLE VII
RESULTS OF RADIATION EXPERIMENTS FOR THE P2041 WORKING IN SMP SCENARIO

SEE Type	*Type of OS fault*	*Test1*	*Test2*	*Test3*	*Consequences*
SEU	Machine Check exception - Cache	6 0	10 0	53 1	None Unreliable sys.
SEU	Machine check exception - Code lost	0 1	0 2	1 0	None Timeout
SEU	Other error messages	1 0	0 0	0 1	Timeout Unreliable sys.
SEU	Abnormal process termination	6 2	3 3	11 22	Timeout Unreliable sys.
SEFI	System hang	3	3	17	System crash
SEFI	Automatic system restart	1	1	8	System crash
	Total	20	22	114	

note that the results include the messages obtained during the execution of the application and during the idle time.

Most of the *Machine Check Exceptions (MCEs)* were produced by errors affecting the cache memories. When the condition exception was recoverable by the system, there were no consequences neither in the application nor in the system. However, there was one case in which a system process of the scheduler was affected. A *Machine check exception - Code lost* occurs when the *MCE routine* has lost the raised error code that has provoked the exception; consequently there is no possibility to determine the source of the error.

An *Abnormal process termination* occurs when the monitor detects a timeout in the application, or when the *MCE* logs an exception in kernel code which causes an unreliable system condition. A *System hang* is produced when the system shell do not respond to any command, or when the *MCE* logs a message showing that a rebooting is needed. In the most critical level, the *MCE* logs an *Automatic system restart* message. Finally, there were errors that did not come either from the *MCE* or the monitor application. They were classified as *Other error messages* and have caused, in one case an application timeout, and in the other case a killing of a system process.

To estimate the dynamic cross-section, only faults that led to *unreliable system* condition, application *timeouts and system crashes* were considered. The total number of events for the three tests was 156, and among them 86 produced errors. The total fluence was $3.27 \times 10^9 n \cdot cm^{-2}$. Applying Equations (1) and (3) for a confidence level of 0.95, the dynamic cross section in SMP scenario is:

$$2.10 \times 10^{-8} \frac{cm^2}{device} < \sigma_{DYN\_SMP} < 3.25 \times 10^{-8} \frac{cm^2}{device} \quad (6)$$

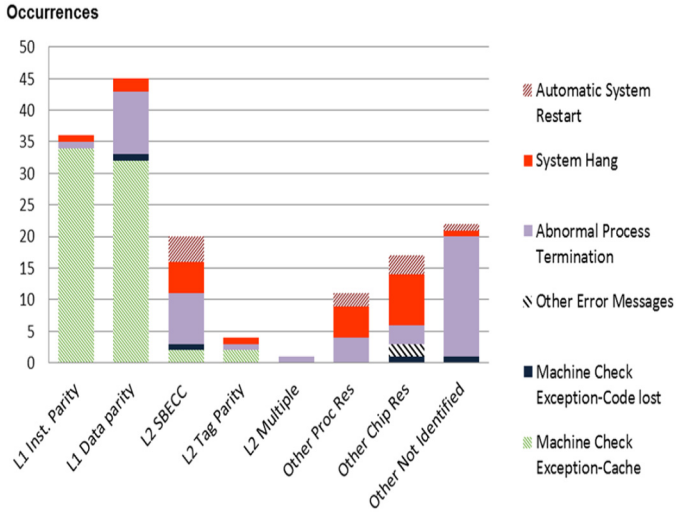

Fig. 4. Relationship between error source and OS fault.

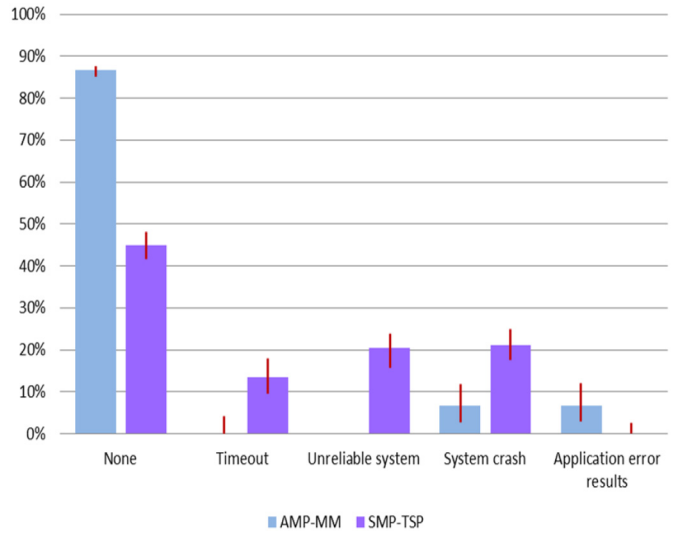

Fig. 5. SEE consequences according to the scenario implemented. The confidence intervals are shown by means of the red lines.

The *machine-check-error report* and the *traps* implemented in the OS allowed determining the source of the errors of these faults. Fig. 4 illustrates the relationship between the OS faults and the hardware source of the error. It is important to note that the dynamic cross section is strongly dependent on the characteristics of the tested scenario. The obtained results show that around of 70% of the errors affects the system while the other 30% affects the application itself. It can be explained, since this scenario maximizes the use of CPU resources and scheduling, and the TSP implementation has an intrinsic fault-tolerant capability.

D. Comparison of SEEs Consequences in the Two Dynamic Scenarios

The SEEs consequences issued from the two dynamic scenarios are shown in Fig. 5. In the AMP-memory-bound 86.67% of the events had no consequences, 6.67% provoked errors in the user application, and 6.67% caused system hangs. On

TABLE VIII
CONFIDENCE INTERVALS OF SEEs CONSEQUENCES

Consequence Type	*AMP-MM*	*SMP-TSP*
None	[62 - 97]	[55 - 88]
Timeout	[0 - 4]	[13 - 32]
Unreliable system	N/A	[22 - 45]
System crash	[2 - 13]	[23 - 46]
App. result error	[2 - 6]	[0 - 4]
Total	[72 - 110]	[132 - 182]

the other hand, in the SMP-CPU-bound scenario 44.87% of the observed events had no consequences on the system or application, 13.46% of the events provoked timeouts of the user application, 20.51% of the events caused an unreliable condition in system and 21.15% crashed the system. Table VIII summarizes the uncertainty margins of the SEEs consequences for the two dynamic scenarios with a 0.95 confidence level.

A comparison of both dynamic cross sections (σ_{DYN_AMP}) and (σ_{DYN_SMP}) shows that the dynamic response of the device depends not only on the application but also on the adopted multi-processing mode. Moreover, the obtained results revealed that errors may occur in SMP mode, even if the OS is in idle mode.

In the literature, there is a work that compares the performance of the SMP and AMP modes both with operating systems for a dual-core giving as a conclusion that SMP outperforms the AMP mode [21]. Inferring this affirmation to the present work, it is possible to suggest the existence of a trade-off between reliability and performance according to the multi-processsing mode selected.

VI. CONCLUSIONS AND FUTURE WORK

This work has evaluated the sensitivity to 14 MeV neutrons of a 45 nm SOI P2041 multi-core processor. From the static test results, it can be seen that 45 nm SOI technology is between 3 and 5 times less sensitive to neutron radiation than its CMOS counterpart.

The dynamic AMP tests have demonstrated that in spite of the parity and ECC protection mechanisms, errors have been occured in the application results. A deeper analysis has allowed to determine that errors were caused by MBUs in the *address tags* and *data*.

Results presented in Section V-D suggest that the studied multi-core working in AMP scenario without OS is at least five times less sensitive to SEEs than working in SMP scenario. The main reason is that AMP scenario exploits fewer chip resources than SMP scenario.

In future work, a similar approach will be applied for other multi-core and many-core circuits with different memory architectures.

ACKNOWLEDGMENT

The authors thank F. J. Franco, their college from UCM, for his valuable collaboration in the analysis of the results.

REFERENCES

[1] R. Hyman, K. Bhattacharya, and N. Ranganathan, "Redundancy mining for soft error detection in multicore processors," *IEEE Trans. Comput.*, vol. 60, pp. 1114–1125, Aug. 2011.
[2] F. Certification Authorities Software Team (CAST)"Position paper cast-32 multi-core processors," https://www.faa.gov/aircraft/air_cert/design_approvals/air_software/cast/cast_papers/media/cast-32.pdfMay 2014.
[3] C. Villalpando, D. Rennels, R. Some, and M. Cabanas-Holmen, "Reliable multicore processors for NASA space missions," *Proc. Aerospace Conf.*, Mar. 2011, pp. 1–12.
[4] P. E. Dodd and L. W. Massengill, "Basic mechanisms and modeling of single-event upset in digital microelectronics," *IEEE Trans. Nucl. Sci.*, vol. 50, no. 3, pp. 583–602, Jun. 2003.
[5] G. Gasiot *et al.*, "SEU sensitivity of bulk and SOI technologies to 14-MeV neutrons," *IEEE Trans. Nucl. Sci.*, vol. 49, no. 6, pp. 3032–3037, Dec. 2002.
[6] G. H. Asadi, S. Vilas, M. B. Tahoori, and D. Kaeli, "Balancing performance and reliability in the memory hierarchy," *Proc. Int. Symp. Performance Analysis of Systems and Software*, Mar. 2005, pp. 269–279.
[7] P. Ramos *et al.*, "Sensitivity to neutron radiation of a 45 nm SOI multi-core processor," *Proc. Radiation Effects on Components and Systems Conf.*, Sep. 2015, pp. 1–4.
[8] E. Normand and L. Dominik, "Cross comparison guide for results of neutron SEE testing of microelectronics applicable to avionics," *Proc. Radiation Effects Data Workshop*, Jul. 2010, pp. 1–8.
[9] S. S. Stolt and E. Normand, "A multicore server SEE cross section model," *IEEE Trans. Nucl. Sci.*, vol. 59, no. 6, pp. 2803–2810, Dec. 2012.
[10] D. A. G. Oliveira *et al.*, "Modern GPUs radiation sensitivity evaluation and mitigation through duplication with comparison," *IEEE Trans. Nucl. Sci.*, vol. 61, no. 6, pp. 3115–3122, Dec. 2014.
[11] M. Rebaudengo, M. Reorda, and M. Violante, "An accurate analysis of the effects of soft errors in the instruction and data caches of a pipelined microprocessor," *Proc. Design, Automation and Test in Europe Conf. Exhib.*, 2003, pp. 602–607.
[12] T. Santini, P. Rech, G. Nazar, L. Carro, and F. Rech Wagner, "Reducing embedded software radiation-induced failures through cache memories," *Proc. Eur. Test Symp.*, May 2014, pp. 1–6.
[13] QNX Software Systems"Running AMP, SMP or BMP Mode for Multicore Embedded Systems,"http://cache.freescale.com/files/32 bit/doc/brochure/PWRARBYNDBITSRAS.pdf2012.
[14] F. Villa *et al.*, "Accelerator-based neutron irradiation of integrated circuits at GENEPI2 (france)," *Proc. Radiation Effects Data Workshop*, Jul. 2014, pp. 1–5.
[15] F. Miller, C. Weulersse, T. Carriere, N. Guibbaud, S. Morand, and R. Gaillard, "Investigation of 14 MeV neutron capabilities for SEU hardness evaluation," *IEEE Trans. Nucl. Sci.*, vol. 60, no. 4, pp. 2789–2796, Aug. 2013.
[16] Freescale"P2040/P2041 QorIQ integrated multicore communication processor family reference manual," http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=P20402013.
[17] Freescale"e500mc Core Reference Manual," http://cache.freescale.com/files/32 bit/doc/ref_manual/E500MCRM.pdf2013.
[18] E. Francesquini, M. Castro, P. Penna, F. Dupros, H. Freitas, P. Navaux, and J. F. Mehaut, "On the energy efficiency and performance of irregular application executions on multicore, NUMA and manycore platforms," *J. Parallel Distrib. Comput.*, vol. 76, pp. 32–48, Feb. 2015.
[19] J. Autran, P. Munteanu, P. Roche, and G. Gasiot, "Real-time soft- error rate measurements: A review," *Microelectron. Reliab.*, vol. 54, pp. 1455–1476, Feb. 2014.
[20] R. Velazco *et al.*, "Evidence of the robustness of a COTS soft-error free SRAM to neutron radiation," *IEEE Trans. Nucl. Sci.*, vol. 61, no. 6, pp. 3103–3108, Dec. 2014.
[21] N. De Witte, R. Vincke, S. Van Landschoot, E. Steegmans, and J. Boydens, "Comparing dual-core SMP/AMP performance on a telecom architecture," https://lirias.kuleuven.be/bitstream/123456789/416618/1/072_Paper-N_Witte.pdf Dec. 2013.