## RIS team (Robust Integrated Systems)

### Themes
- Robust massively parallel single-chip architectures
- Power management from the OS down to silicon
- Fault tolerant and self-adaptive architectures
- 3D NOC Robust Architectures
- Design in Reliability face to aging, process variation and soft errors
- Evaluation of robustness and qualification: radiation testings, fault injection
- Architectures for Nanotechnologies

### Expertise

#### Fields of expertise
- Design for Reliability, Design for Test, Self-Repair, Fault-tolerance, Design for Soft-Error Mitigation: Methodologies, Tools and Architectures

#### Know-how
- Multilevel platforms for fault simulation and robustness automatic insertion at several abstraction levels; 3D integration solutions
- Test platform for radiation faults measurement; SEE error-rate prediction of circuits and systems

### Research keywords
- Fault tolerance, multi-core systems robustness, 3D circuits, aging, fault-injection

### Contact

<table>
<thead>
<tr>
<th>Michael NICOLAI DIS</th>
<th>Raoul VELAZCO</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNRS Research Director</td>
<td>CNRS Research Director</td>
</tr>
<tr>
<td>(+33) 4 76 57 46 96</td>
<td>(+33) 4 76 57 46 89</td>
</tr>
<tr>
<td><a href="mailto:michael.nicolaidis@univ-grenoble-alpes.fr">michael.nicolaidis@univ-grenoble-alpes.fr</a></td>
<td><a href="mailto:raoul.velazco@univ-grenoble-alpes.fr">raoul.velazco@univ-grenoble-alpes.fr</a></td>
</tr>
</tbody>
</table>
The Cells Framework: Overal Description

Keywords: Ultimate CMOS and post-CMOS technologies, high defect densities, reliability, yield, low-power, massively parallel single-chip tera-device computers

Members: T. Bonnoit, F. Bouesse, A. Charif, A. Coelho, P. Papavramidou, M. Nicolaidis, N.E. Zergainoh

Cooperation: STMicroelectronics, iRoC, Atmel

Contracts: ELESIS, RESIST

Ultimate-CMOS and post-CMOS technologies promise integrating trillions devices in a single die, leading to single-chip massively parallel architectures comprising thousands interconnected processors, and enabling the next computation turn. But the aggressive technology scaling that paves the way to the ultimate CMOS nodes has dramatic impact to: process, voltage and temperature (PVT) variations; sensitivity to electromagnetic interferences (EMI), to atmospheric radiation (neutrons and protons) and to alpha particles; and circuit aging. It also imposes stringent power dissipation constraints. The resulting high defect levels, heterogeneous behavior of identical circuits, accelerated circuit degradation over time, and extreme complexity, affect adversely fabrication yield and/or prevent fabricating reliable chips in ultimate CMOS and post-CMOS technologies. These issues are becoming the main show-stoppers in the path leading to these technologies.

The Cells framework addresses the severe issues related to the design of massively parallel tera-device processors affected by high defect densities, in which we severe issues have to be addressed such us:

- After fabrication, all processing and routing nodes may be affected by some temporary faults such as delay faults, or clock skews.
- Fabrication faults altering persistently the circuit behavior may massively affect one or more regular blocks (RAMs, FIFOs, buses) in a large fraction of nodes. Such faults may also very frequently occur during product life.
- Fabrication faults altering persistently the behavior of irregular blocks (thus difficult to repair) may affect a significant portion of nodes. Such faults may also frequently occur during circuit life (e.g. every few days), and thus during application execution.
- New timing faults induced by circuit aging, as well as soft errors (SEUs and transients) may frequently occur during circuit life (and thus during application execution).
- Circuit degradation is continuous and requires continuous self-regulation of circuit parameters (clock-frequency, voltage levels, body bias), to maintain operational each processor node.

Clearly, no existing solution can cope with such massively defective systems, which invalidate even massive redundancy schemes (e.g. duplication, TMR), as all replicated parts may be defective. Such schemes also induce high area and power penalties. Some approaches targeting the design of reliable single-chip massively parallel processors avoids massive redundancy by using self-tests (hardware implemented or software implemented to detect failures and create routing tables that are used subsequently to avoid failed processing nodes or failed routes. However, such approaches could not cope with the issues affecting ultimate CMOS and post CMOS technologies as:

- In highly defective technologies, the vast majority of nodes (processing elements and routers) may include one or another kind of faults (e.g. timing faults produced by process, voltage and temperature variations, EMI, or aging). Thus, declaring defective the nodes affected by any kind of faults will quickly waste the computational resources of the chip.
- Achieving high fault coverage for timing faults is very difficult. Thus, many of these faults may escape fabrication test and also periodic self-tests and produce run-time errors.
- Faults occurring during application execution can not be covered by self-tests.
- In this project we develop a comprehensive approach enabling using in efficient and reliable manner all parts able to perform useful computations.

The Cells framework (On-Chip Self-healing Tera-Device Processors) [1-2] comprises several techniques spanning at all levels of the system: circuit, processor/architectural, array/routing, task-scheduling/allocation. Innovations are introduced at all levels of this framework, including its overall architecture, its particular components, and the way the cooperation of these components is architectured to optimize the outcome. Developments concerning some of these components are presented in the next sections.

References

Publications on the Overall Cells Framework


Recent Publications on Cells' Components


Early Publications on Cells' Components


Circuit-level fault-tolerance in the Cells Framework

Keywords: PVT variability; aging; timing faults; soft errors; design for reliability, yield, and low-power; double-sampling schemes

Members: T. Bonnoit, F. Bouesse, M. Nicolaidis, N.E. Zergainoh

Cooperation: iRoC, Atmel, EADS

Contracts: ELESIS, RESIST

In massively parallel processor chips, a possible (and rather straightforward) approach may consist in exploiting the existence of large numbers of processing cores to implement fault tolerance, by using two processor cores to duplicated the execution of each task and comparing their outcome to detect errors (double modular redundancy-DMR), or by using three processor cores to triplicate the execution of each task and voting their outcome to correct errors (triple modular redundancy-TMR). These approaches have two major flaws: drastic decrease of processing power, and drastic increase of energy dissipation per task. Similarly, software implemented fault-tolerance drastically impacts performance and power, as it replicates the execution of each task. Hence, a breakthrough in fault-tolerant design is required for reducing drastically performance and energy dissipation penalties, and improve reliability.

Such a breakthrough was achieved by a scheme (referred hereafter as double-sampling) we proposed in reference [1] and evaluated in [2]. This scheme reduces drastically hardware and power costs, by avoiding both hardware replication and operations replication. Instead, to check the correctness of an output signal of a logic block; this scheme observes this signal at two different instants, by adding a redundant latch and driving it by means of a delayed clock signal (Figure 1a). This scheme was later extended in the RAZOR architecture [3] to also perform error correction: upon error detection Razor used the contents of the redundant latch to replace the contents of the functional flip-flop (Figure 1b). Reference [3] demonstrated that this scheme can achieve drastic power reduction, by reducing the supply voltage to very low levels and using RAZOR to detect and correct the timing errors induced by this reduction.

While the double-sampling scheme reduces drastically the area and power cost with respect to traditional fault-tolerant schemes, some area and power penalties are still introduced due to the redundant latches. Furthermore, it suffers from two drawbacks:

- To avoid false alarms and miss-corrections, the path delays of the combinational circuits signals checked by the schemes in figure 1, must exceed the delay $\delta$ of the delayed clock. To ensure this constraint, buffers should be needed in the paths with delays shorter than $\delta$, inducing no negligible area and power cost.
- Also, due to this issue, to avoid significant area and power penalties, we have to use moderate values for $\delta$, limiting the duration of detectable faults.

1. GRAAL Architecture

One goal of Cells is to explore new fault tolerant architectures that are exempt of the above problems. These problems are due to the fact that all the stages of a FF-based design compute at the same time. This leaves short stability time to the combinational circuit outputs that we could exploit for error checking. In flip-flops, when the master latch is transparent, the slave is on memorization and vice versa. Thus, if we
transform a flip-flop-based design (Figure 2a), into its equivalent latch-based design, by moving the slave latches of the FFs to the middle of the combinational circuits, (as shown in Figure 2b where Ck and Ckb are replaced by non-overlapping clocks Φ1 and Φ2), we obtain a design that can work at the same clock frequency as the original one. In addition, the master and slave latches operate at different phases. Then, the outputs of any combinational block (inputs of latches), are stable during the period in which its adjacent blocks are in computation. Thus, we can compare the outputs of the latches against their inputs to detect timing and transient faults of large duration, according to the GRAAL architecture (Figure 3) that we have introduced few years ago.

Our GRAAL architecture can be implemented in two ways. The one version uses redundant latches, which introduce non-negligible area and power penalties. Recently this first version of our GRAAL architecture was used by others in the so-called Bubble Razor implementation. The second version of GRAAL architecture is much more efficient as it does not use of redundant latches. Only a two-input XOR gate is added to each latch to compare its inputs against to its outputs. Thus, this version of GRAAL induces drastically lower area and power penalties with respect to any known error detection scheme.

Table 1 shows the evaluation results of the GRAAL architecture. These results are based on the implementation of GRAAL in the icyflex processor from CSEM [4][5], in ST Microelectronics 40nm technology. In order for the comparisons to be consistent, all implementations of icyflex, shown in table 1, were synthesized for the same clock frequency. Thus, the 150MHz clock frequency suggested by CSEM was selected for all these implementations.

Latch-based designs can be synthesized with or without time borrowing, and the synthesis was done by means of the Design Compiler [6], which supports latch-based design for both cases.

Latch-based designs can be synthesized with or without time borrowing, and the synthesis was done by means of the Design Compiler [6], which supports latch-based design for both cases. The maximum duration of detectable faults is achieved by GRAAL when no time borrowing is used. Thus, no-time borrowing has been first selected for achieving the highest possible reliability.

<table>
<thead>
<tr>
<th>icyflex2 Implementations</th>
<th>Reference Latch no-time Borrowing</th>
<th>Latch GRAAL A no-time Borrowing</th>
<th>Flip-Flop</th>
<th>Latch Time Borrowing</th>
<th>Latch GRAAL B Time Borrowing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cell Library</td>
<td>1.10V</td>
<td>1.10V</td>
<td>1.10V</td>
<td>0.90V</td>
<td>0.90V</td>
</tr>
<tr>
<td>Area</td>
<td>100%</td>
<td>103.60%</td>
<td>100.98%</td>
<td>108.16%</td>
<td></td>
</tr>
<tr>
<td>Power</td>
<td>100%</td>
<td>103.95%</td>
<td>156.7%</td>
<td>64.52%</td>
<td>70.81%</td>
</tr>
<tr>
<td>Duration Detectable Faults</td>
<td>---</td>
<td>100% path delay</td>
<td>---</td>
<td>---</td>
<td>72.4%</td>
</tr>
</tbody>
</table>

Table 1: GRAAL evaluation

For comparison purposes, we use "Reference" version, which is the unprotected icyflex version, synthesized without time borrowing by using a cell library targeting 1.10V supply voltage. The area and power of the "Reference" version are given in normalized terms (100% area and 100% power) in column 2 of table 1.

Column 3 shows the results for the GRAAL implementation, using the same cell library and no time-borrowing. We observe that this implementation achieves detection of faults that increase by 100% the delay of the affected path. This duration is very high and is achieved by means of very low cost (3.06% area and 3.95% power cost). Thus GRAAL architecture can achieve very high reliability in future highly defective technologies that could increase dramatically PVT (process, voltage, and temperature) variations, and circuit aging, due to a very aggressive...
reduction of device geometries. Thus, GRAAL is very suitable for the Cells framework.

To validate the efficiency of GRAAL also with respect to flip-flop based design, column 4 shows the area and power of the flip-flop implementation of icyflex, using the same cell library and targeting the same clock frequency (150 MHz). We observe that, the flip-flop version requires 7.28 % less area with respect to the “Reference” latch-based version, but much larger power (56.77 %). Thus, the GRAAL no-time borrowing implementation achieves its high fault detection efficiency (faults that increase by 100 % the delay of the affected paths) at a 11.73 % area cost with respect to the unprotected flip-flop-based implementation, but at much lower power (33.66 % less than the unprotected flip-flop-based implementation).

2. No time borrowing implementation
The implementation presented in the previous section does not use time borrowing and achieves the highest robustness, as faults of very high duration are detected (i.e. faults of so large duration that may duplicate the delay of the affected path). Note that the detection of faults of high duration can also be exploited for reducing power dissipation, by aggressively reducing the supply voltage level, and using GRAAL to detect and recover the resulting timing errors. However, a more efficient approach consists in using combining GRAAL with time borrowing.

To reduce power, we used the TT/0.90V/125°C cell library, which is designed for 0.90V supply voltage. However, as the delays of this cell library are higher than the delays of TT/1.10V/125°C, achieving operation at 150 MHz will impact power dissipation and reduce the gains achieved by the reduction of supply voltage. Thus, to maximize the power reduction, we used the Design Compiler to synthesize the time-borrowing version of icyflex2 for the TT/0.90V/125°C.

Column 5 in table 1 shows the results of the unprotected version of icyflex2 using this cell library and time borrowing, and targeting 150MHz clock frequency. This version requires 0.98 % more area with respect to the “Reference” version of column 2, and reduces power by 35.48 %.

Column 6 in table 1 shows the results concerning the GRAAL version of the time-borrowing implementation of icyflex2, for the same cell library and clock frequency. This version detects faults of duration up to 72.4 % of the delay of the path affected by the fault. Thus, the GRAAL architecture detects faults of very large duration, even for time borrowing implementations. This significant advantage is achieved at the small cost of 7.18 % more area and 6.29 % more power with respect to the unprotected time borrowing implementation of column 5. In comparison with the unprotected flip-flop implementation, this high detection efficiency is achieved at the cost of 16 % more area and at the drastic advantage of 45.19 % less power. Hence, the combination of GRAAL with time borrowing is also very suitable for the Cells framework, as power reduction is very important for Cells, and fault detection efficiency remains very high.

3. Protecting Flip-flop based designs
Note that though the preferred error detection scheme in Cells is GRAAL, Cells is also compatible with any other error detection scheme. For flip-flop based designs, two double-sampling architectures were developed:
- The one, referred a ADDA architecture [6][7], allows operating the same design in 3 modes: the first is the mode used in earlier double-sampling schemes; the second allows early failure prediction; the third allows detecting transient faults of large duration, that could be encountered in hostile environments like space.
- The other, presented in [7][8], reduces drastically the area and power penalties of the double-sampling architecture, by removing the redundant sampling element. With this solution, the only cost for implementing double-sampling consists in the comparison of the flip-flop outputs against their inputs.

The evaluation of these architectures is on-going by implementing them in the LEON3 processor.

4. References
[4] Icyflex2 user manual, from CSEM
Memory and Interconnect Self-Repair for High Defect Densities in the Cells Framework

Keywords: PVT variability; aging; timing faults; soft errors; memory repair; design for reliability, yield, and low-power

Members: T. Bonnoit, A. Charif, A. Coelho, P. Papavramidou, M. Nicolaidis, N.E. Zergainoh

Contracts: TOETS

Embedded memories occupy the largest part of SoCs and include even larger amounts of transistors. As memories are designed very tightly to the technology limits, they are more prone to failures than other circuits. Thus, they concentrate the large majority of fabrication defects and may affect yield adversely. In addition, failure rates are expected to be exacerbated as we approach the ultimate limits of CMOS, and should further worsen in post-CMOS technologies. Furthermore, low power requirements, which are stringent in modern electronic systems, will also be exacerbated as we move towards ultimate CMOS and post-CMOS, requiring aggressive reduction of supply voltage. Unfortunately, voltage reduction results in additional failures, as weak memory cells will not operate correctly at reduced voltage levels. Furthermore, as ultimate CMOS and post-CMOS are expected to exacerbate PVT variability and circuit aging, the ratio of weak cells will sharply increase. As a consequence, techniques enabling coping with high defect densities in memories are required in the Cells framework.

Memories and interconnection buses have very regular structure and enable easy repair. Various efficient memory and interconnections Built-In Self-Repair (BISR) schemes have been proposed in the past, and several of them are already industrial practice. However, for high defect densities, conventional memory repair schemes will induce excessive area and power penalties. Thus, innovative solutions are required able to drastically reduce these penalties. The ECC-based memory repair scheme, introduced by our group, is the only known scheme able to cope with high defect densities. However, while the ECC-based repair scheme drastically reduces area and power penalties with respect to conventional schemes, power penalty is still non-negligible. Furthermore, implementing self-diagnosis for the ECC-repair scheme, further increase area and power penalties. Our recent developments have eliminated these issues [1]. First, a new family of memory test algorithms was proposed eliminating diagnosis-related area and penalties at the expense of test length [2][3]. Then, an iterative diagnosis architecture was proposed [4], trading test length with area cost. In a third innovation [5], a hardware architecture is proposed drastically reducing power dissipation for both ECC-based repair and non-ECC-based repair. Further reduction of this power is achieved by introducing an approach that rearranges the faulty addresses stored in the repair circuitry, in a manner that disables most of the time a part of this circuitry [6]. Thanks to these improvements and the use of the ECC-based repair, an impressive power reduction is achieved with respect to conventional repair.

For instance, for a SoC comprising a total of 10 Gbit memory capacity distributed over 300 embedded SRAMs using words of 32 bits (plus 7 ECC check bits), conventional repair for a 10⁻³ probability that a memory cell is faulty, conventional repair induces a huge power penalty of 1629 %, which is due to the fact that repairing such high fault rates requires using large repair CAMs, and also that CAMs are power hungry. Then, the combination of ECC-repair with the improved repair architectures in [5][6], reduces this penalty to only 7.36 % [6]. Hence, these developments provide a comprehensive framework enabling low cost memory repair for high defect densities, which is very suitable for the Cells framework.

In other developments, we proposed a BIST architecture enabling the cooperation of transparent BIST with ECC-based repair [7]. Transparent BIST is an approach we introduced at 1996, and further developed and used by numerous authors and fault-tolerant systems. This approach transforms memory test algorithms into reversible processes, which preserve the information stored in the memory. Transparent BIST is essential in the context of Cells in order to preserve application context during periodic memory self-tests. The fundamental problem for making cooperating transparent BIST with ECC-based repair is that transparent BIST uses signature analysis allowing determining if a memory is faulty (go-no-go test), while ECC-memory repair requires more subtle diagnosis (i.e. determining if there are memory words containing multiple faulty cells). While the signature analysis alone does not allow making such diagnosis, and also the error detection and correction circuitry alone can also misdiagnose words containing more than two faulty cells, the new transparent BIST architecture we have developed in the context of Cells resolves this problem thanks to a subtle cooperation between the signature analysis and the ECC circuitries.
Integrating these approaches in the Cells project will enable healing tera-device processor chips comprising billions of defective memory cells. For instance, a massively parallel tera-device chip comprising thousands of embedded memories of a 1 Tera-bit total capacity and affected by a $10^3$ defect density, will comprise $10^9$ faulty memory cells (including definitely faulty cells as well as weak cells failing during aggressive voltage reduction modes). The Cells project will be able making such a chip operational, and last but not least, this unprecedented goal will be achieved at very low area and power penalties.

Interconnections may also represent an important reliability challenge, especially for 3D systems where the vertical interconnects (TSVs) may be affected by high defect densities. Efficient Built-In Self-Repair schemes for TSV interconnect, using parallel message transmission and serial message transmission, allow resolving this issue in cost effective way.

References
Array-Level Approaches in the Cells Framework

Keywords: Ultimate CMOS and post-CMOS technologies, high defect densities, reliability, yield, low-power, massively parallel single-chip tera-device computers, array level self-adaptive approaches

Members: A. Charif, A. Coelho, M. Nicolaidis, N.E. Zergainoh

1. Adaptive fault tolerant routing
In complex grids comprising thousands of nodes, routing algorithms based on routing tables are congestion prone and have low adaptability to new failures. Hence, we developed distributed algorithms using local opportunistic decisions (get another path when a node/router/link is faulty or congested). Our simulation results show that they easily scale for grids comprising thousands nodes; tolerate multiple faulty nodes/routers/links; avoid congestions; and cope with new failures occurring at any time. But, as they are based on local decisions, they are deadlocks prone. To cope with we use virtual networks and pertinent rules acting at the local level.

2. Adaptive fault tolerant task scheduling and allocation and error recovery
At a first step we integrate in a single algorithm our FT routing approach (discussed above) with distributed FT scheduling and allocation. We also integrate in the same algorithm check-point-free rollback recovery. This is done by an approach which: uses an hierarchical task organization into parent-children trees; maintains this hierarchy during execution; and if a persistent fault occurs in a resource executing a child it goes back to the parent to redistribute the workload in fault-free resources. This way we avoid saving the internal states of the grid to external media (no check-pointing), which would congest the grid I/Os, and we can recover correct operation even after the occurrence of any multiple fault (by going up in the tree until a fault-free parent, which redistributes the aborted workload to fault-free resources).

Error recovery algorithms based on check-pointing were also implemented. A basic issue is related to the fact that as the state of the system is huge; we have to perform check-pointing for partial states of the system (e.g. states of tasks), while preserving its overall coherence. To cope with this issue a check-pointing algorithm for parallel processors maintaining system coherence was developed.

3. Variability awareness
In further developments we extended the algorithm to enable variability aware and power aware task scheduling and allocation. Again we use a distributed non-deterministic approach to handle computation complexity. One of the developed schemes maps different groups of tasks into different regions of the grid according to the energy dissipated by each group and the power-dissipation characteristics of the regions. Then, the leader of each region maps task clusters into sub-regions, and the leader of each sub-region maps each task to a node, by using similar power dissipation considerations. The clock frequency required for meeting the deadline of each task and the power/reliability priorities of the task are encapsulated in the header of each task. Each node knows its frequency/Vdd operating domains for various error occurrence rates (by monitoring its concurrent error detection signal). Thus, it can determine its clock frequency and Vdd level to reach the task deadline, minimize the energy dissipated, and achieve the target reliability level (in terms of error occurrence rate). Thus, circuit parameters (clock frequency and Vdd) are continuously regulated to minimize power, preserve reliability and adapt to circuit degradation induced by aging.

Our work on array level self-adaptive algorithms was presented in numerous articles published in various international conferences during the past few years [1][10].

The next step of our efforts concern the integration of these algorithms (together with the other techniques used in Cells), in a unified framework implemented in the GEM5 simulator and the Structural Simulation Toolkit (SST).

4. References
Interconnects", Design Automation and Test in Europe Conference (DATE), March 14–18, 2011, Grenoble, France, (BEST IP AWARD)


Fault Tolerance Capabilities of Multicore Systems

Keywords: Fault Injection, Reliability, Multi-processing, Soft Errors

Members: P. Ramos, V. Vargas, R. Velazco, N.E. Zergainoh

Cooperations: KALRAY

1. Context and goals
The new paradigm in computing systems based on multicore processors affects the traditional criteria in software and hardware technologies. Nowadays, the characteristics of reliability, dependability, and availability are crucial not only for critical applications.

The complexity of multicore architectures, due to the number of cores, concurrency issues, shared resources and interconnections among cores is a potential source of a wide scope of errors. In order to facilitate the handling errors, the manufacturers have introduced the error reporting architecture by the machine check error registers to provide information of the error sources.

Fault injection in processor-based architectures was a topic largely addressed by scientific community to validate the reliability of critical applications. The CEU (Code Emulated Upset) approach, developed at TIMA/ARIS in 2000, based on the assertion of interrupt signals was demonstrated by previous researches as being very efficient and able to provide error-rate results close to those obtained in radiation experiments. Its application in the past researches to a complex processor, the PowerPC4748, allowed validating it for aeronautics application.

In this project, the CEU approach principle was adapted for the first time to a multicore processor. The execution of the interrupt handler provokes the selected error (SEU, MBU) in a randomly chosen target. In the case of multicore-processors, we can benefit of the multiplicity of cores for using one of them as a fault injector while the others run the application.

2. Recent outcomes
Results obtained for a 256-core processor
For evaluating the impact of SEEs on applications running on the MPPA many-core processor, two case-studies were proposed. The first one maximizes the use of internal compute-clusters resources while the second one exploits massive parallelism. The evaluation platform used to appraise the many-core processor is the MPPA Developer. The evaluation was achieved through fault injection campaigns and/or neutron radiation testing. Details of each case are described in the related sections.

The MPPA is a distributed memory system. Each Compute Cluster (CC) is built around a multi-banked local Static Memory (SMEM) of 2MB shared by 17 identical Very Long Instruction Word (VLIW) cores: 16 Processing Engine (PE) + 1 Resource Manager (RM) without cache coherency. This configuration creates an interconnection with high bandwidth and throughput between PEs. The PEs are dedicated to execute the application code while the RM is in charge of managing the NoC interfaces by means of dedicated event lines and interrupts.

The main memory areas of the many-core processor are covered by error protection mechanisms except the instruction and data cache memories of the Very Long Instruction Word (VLIW) core that are protected by parity. The SMEM of the clusters interleaves bits of 8 adjacent 64-bit words which allow localized errors spread as multiple Single Error Correcting Code (SECC) errors. They are detected and corrected on the fly. The NoC router queues (512 of 32-bit flits each) are also protected by ECC.

Many-core execution with minimal use of NoC services
This case-study aims at evaluating the sensitivity of the internal computing-cluster resources. To do this, the many-core processor is configured in bare-metal where each cluster executes independently a same Matrix Multiplication application. Its behavior was evaluated both by fault injection and neutron radiation tests.

Results show that during the radiation test campaigns on the MPPA, no errors were produced in SMEMs of the clusters since they implement ECC and interleaving. Consequently, this work only considers fault injection in processors’ registers. On the other side, for the evaluation of the device under neutron radiation, two scenarios are considered: the first one with cache memories enabled and the second one with cache memories disabled.

In the fault-injection experiments, the SEU faults were injected at a random instant within the nominal duration of the executed program which was around 5.3 x 10^8 clock cycles. A total of 94316 faults were injected in the General Purpose Registers (GPRs) and System Function Registers (SFRs).

<table>
<thead>
<tr>
<th></th>
<th>Silent faults</th>
<th>Erroneous results</th>
<th>Time-outs</th>
<th>Exception</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPRs</td>
<td>36472</td>
<td>16387</td>
<td>6678</td>
<td>1996</td>
</tr>
<tr>
<td>SFRs</td>
<td>22745</td>
<td>2034</td>
<td>6365</td>
<td>1639</td>
</tr>
<tr>
<td>Total</td>
<td>59217</td>
<td>18421</td>
<td>13043</td>
<td>3635</td>
</tr>
</tbody>
</table>

Table 1: MPPA Fault injection campaign results

Result on Table 1 shows that 37.21 % of the injected SEUs in the accessible registers cause...
errors in the application. Since registers have no protection mechanisms, this campaign proves to be very useful to emulate the behavior of the application in presence of SEUs.

On the other hand, the goal of the neutron radiation test is to evaluate the dynamic behavior of the many-core processor when no operating system is used. To do this, two different scenarios are considered. In the first scenario, the cache memories of the VLIW cores are all enabled and configured in write-through mode. In the second one, the cache memories are all disabled. In both cases, the application cross-section and the average execution time are obtained. The Single Error Correcting Code (SECC) and other errors such as Data Parity (DPAR) and Instruction Parity (IPAR) are reported with a message.

### Table 2: MPPA Neutron radiation campaign results

<table>
<thead>
<tr>
<th>SEE type</th>
<th>Errors CE</th>
<th>Errors CD</th>
<th>Consequences</th>
</tr>
</thead>
<tbody>
<tr>
<td>SECC</td>
<td>676</td>
<td>602</td>
<td>None</td>
</tr>
<tr>
<td>Data Cache Parity</td>
<td>36</td>
<td>N/A</td>
<td>None</td>
</tr>
<tr>
<td>Inst. Cache Parity</td>
<td>6</td>
<td>N/A</td>
<td>None</td>
</tr>
<tr>
<td>Register Trap</td>
<td>1</td>
<td>1</td>
<td>Hang</td>
</tr>
<tr>
<td>Memory comp. failed</td>
<td>2</td>
<td>1</td>
<td>Error Result</td>
</tr>
<tr>
<td>Total</td>
<td>721</td>
<td>604</td>
<td></td>
</tr>
</tbody>
</table>

From Table 2, it is possible to identify five different types of errors. Among them, the SECC and the Instruction and Data Cache Parity errors were all corrected by the ECC and parity protections. On the contrary, Register Trap and Memory-Comparison Failed errors are non-correctable errors since processor registers do not implement protection mechanisms. The Memory Comparison Failed error was detected by the RM core when it identifies differences between the results of the application and the expected values. Note that these errors observed during radiation tests were similar to those occurred during fault injection campaigns in GPRs.

For a 95 % confidence interval, the lower and upper limits for the dynamic cross-section when cache enabled were: 20.29x10^4-9 cm2/device. Since the code is quite small and only occupies around 400 bytes that easily fit in the PEs' caches, there were no cache misses for the instruction caches.

Many-core execution with minimal use of NoC services

This case-study aims at evaluating the sensitivity of the MPPA when massive parallelism is used. For that, a CPU-bound (TSP) and a memory-bound (MM) were implemented as parallel inter-cluster applications. The programming paradigm used intracluster was POSIX and the inter-cluster communication were done by NoC services. The dynamic response of the TSP and the MM were only evaluated through fault injection in processors’ registers, since the SMEM of the MPPA many-core implement effective protection mechanisms.

The many-core processor was configured in a typical master/slave scheme for running parallel multi-cluster applications. The master runs on the IO cluster while the slaves run on the Compute Cluster (CC). The MPPA part was loaded through the JTAG port to the IO-DDR0. For this scheme, 3 RM cores of the IO-DDR0 were configured: The RM3 is the fault injector while the RM0 coordinates the actions between the application and the fault injector. The fault is generated at a random instant within the nominal duration of the application. In order to avoid the propagation of errors to the next execution, the HOST resets the platform and reloads the code to the MPPA processor after each run.

Table 3 shows a general overview of the fault-injection campaigns on the TSP and MM applications. These results confirm the intrinsic fault-tolerant capability of the TSP application.

### Table 3: MPPA Fault injection campaign results (TSP and MM campaigns)

<table>
<thead>
<tr>
<th></th>
<th>Silent faults</th>
<th>Erroneous results</th>
<th>Timeouts</th>
<th>Exception</th>
</tr>
</thead>
<tbody>
<tr>
<td>TSP</td>
<td>8197</td>
<td>2</td>
<td>21</td>
<td>197</td>
</tr>
<tr>
<td>MM</td>
<td>62071</td>
<td>5215</td>
<td>112</td>
<td>5099</td>
</tr>
</tbody>
</table>

From these results, the error-rates of both applications were calculated. The erroneous results, timeouts and exceptions were considered as errors. Figure 1 summarizes the obtained results. Three types of exceptions were produced: (1) the PE targeted by the fault-injector was completely stuck and does not respond anymore to the JTAG requests "core stuck", (2) the bit-flip caused the core tried to access a memory not allocated producing a "segmentation fault", and (3) the MPPA device stopped its execution and produced an exit of the process "device exit". Figure 1 illustrates the details of the error consequences in both applications during fault-injection campaigns. As expected, during the fault-injection campaigns, it was observed that the critical registers are application dependent.

Although the application error-rate is underestimated because of the inaccessibility of certain registers depending on the configuration of the system, fault-injection campaigns targeting registers are meaningful to emulate the behavior of the application in presence of SEUs. Considering the redundancy capability of the MPPA many-core processor as well as the lack of protection mechanism at register level, it is imperative to implement a fault-tolerant approach to reduce the SEE impact on the running application. The goal is to have a reinforced system to be considered for safety-critical and embedded parallel applications.
3. References (vérifier noms avec Raoul)


Method and tools to emulate soft errors in HDL-based designs

Keywords: single event transients, fault injection, hardware description language, and soft-errors


1. Context and goals
In the last years, the semiconductor industry has been particularly interested in the effects of radiation on integrated circuits, reprogrammable platforms and embedded systems in general. The rationale behind this motivation lies not only in the use of these systems in radiation environments but also in the increasing degree of integration of devices embedded in the same chip. Recent studies have shown that the more miniaturization, the greater the sensitivity to radiation-induced errors. As a consequence, modern embedded systems may be susceptible to low-energy particles including those observed under the Earth's atmosphere. In order to study the effects of Single Event Upsets (SEUs) and Single Event Transients (SETs), accelerated radiation test campaigns allow analyzing the behavior of digital circuits under a large particle flow.

However, these campaigns are too costly; they require at least a prototype of the integrated circuit and also demand considerable technical and programmatic efforts. In this context, simulation and emulation methodologies are becoming increasingly popular as efficient alternatives to evaluate and predict the behavior of these circuits before manufacturing. In general, simulations allow greater flexibility and controllability while emulations exploit reprogrammable hardware platforms (i.e., FPGAs) to execute the experiments in real time.

Emulation techniques involve including saboteurs circuits in charge of injecting faults (modifications) to the sensitive parts of the Device Under Test (DUT). Emulations does not incur in any time penalty; although they require extra area margin to locate the saboteurs in the FPGA. Among existing emulation techniques, NETList Fault Injector (NETFI) was proposed and extended as a method to inject faults at the Register-Transfer Level (RTL) by modifying Xilinx library components. The combination of the controllability and observability of the experiment with the ability to inject faults within each single clock cycle is probably the strongest point of NETFI. However, this methodology has been criticized for the complexity and rigidity associated with the error injection controller.

The original methodology of NETFI was extended and improved in several aspects in its second version called NETFI-2. Unlike NETFI, NETFI-2 allows controlling the injection campaign from the same FPGA where the DUT is instantiated, thus minimizing the amount of hardware required. NETFI-2 includes the design of a more efficient and flexible controller based on an embedded MicroBlaze processor, which allows a better design of the fault injection campaign since it can be conveniently programmed in software. Results demonstrate NETFI-2 advantages and validate the methodology by means of a fault injection on a DUT implementing a Bayesian Machine.

NETFI-2 Description
Figure 1 illustrates the workflow proposed for NETFI-2. Each step is detailed below.

Step 1: The HDL code of the DUT is synthesized using the Synplify tool in order to obtain the corresponding netlist for the Xilinx FPGA.

Step 2: The first obtained netlist is then used as input for the MODNET tool. The second netlist is then synthesized in an EDIF format using a modified version of the sensitive components, which include signals to access them to fault injection.

Step 3: The EDIF file obtained in step 2 is then attached to the MicroBlaze and the last synthesis is performed to generate the bitstream based on the target FPGA.

Step 4: Finally, the experiment is executed in a hardware-based FPGA platform.

Figure 2 illustrates the general architecture of NETFI-2 comprising both the campaign controller and the DUT instantiated in the same FPGA. The DUT is contained within an Advanced Extensible Interface (AXI) Lite protocol slave managed by the Xilinx MicroBlaze processor. The AXI interface allows communicating the DUT with the processor with minimal FPGA area usage.

Specifically, the master uses memory mapping to read or write values into 32-bit registers (extendable to 64 bits) contained in the slave device. NETFI-2 uses this communication channel to configure both the fault injection in the components already intervened by MODNET as well as the input and output values of the DUT. The MicroBlaze control makes use of standard interfaces like UART or Ethernet to report the progress of the experiment as well as its final results. It should be noted that fault injection does not necessarily depend on this external communication, thus the execution speed is not compromised. This is a key advantage of NETFI-2 against existing methodologies based on external controllers.
DUT for SEU injection requires an additional 25 % of hardware while for the SET injection a 9.25 % respectively. This means that the extended DUT for SEU and SET injection requires an additional hardware of 25 % and 9.25 % respectively. Consequently, the greater the number of BM-Slices, the higher the stochastic output of the system. In theory, the complete system composed of ‘n’ BM-Slices should be intrinsically robust against failures. In this first work we will evaluate the robustness of an individual BM-Slice in order to validate the NETFI-2 method. A complete evaluation of a full BM is left as further work.

### Evaluation and Validation

In order to evaluate and validate NETFI-2, a DUT is proposed as an example to be modified by MODNET and then submitted to the fault injection both by means of NETFI-2 as well as by an alternative method of simulation that will serve as reference. Then, in addition to comparing the equality of results, we will study the time required by these methods.

As an example DUT, we propose the use of a module of a Bayesian Machine (BM) implemented in VHDL. A BM is a type of stochastic compiler capable of calculating Bayesian inferences based on a set of probability distributions presented in their input

$$\text{cite}[/text]{Mazer2015}$$. These machines base their internal architecture on variables represented as stochastic bitstreams which can make them relatively resistant to failures. In particular, the proposed BM is composed of small parallel modules called BM-Slices. Each of these takes as input 13 bits representing the probabilistic bus and provides 1 output bit.

Table 1 details the resources required for each component in the target FPGA. The Xilinx (i.e., Vivado) libraries had to be modified to support the Nexys 4 board. Indeed, the original MODNET tool was updated accordingly in order to support state-of-the art FPGAs from this manufacturer. Synthesis results show that the Microblaze processor (including the AXI interface) used as the controller of the experiment only requires 2.53 % LUTs and 1.26 % registers of the target FPGA. On the other hand, it can be observed that MODNET incorporates 27 extra LUTs for the modified DUT with the SEU injectors (1 multiplexer must be incorporated for each sensitive FF detected). Also, 8 LUTs and 2 extra registers were added for SET injectors (2 registers for the AXI signals required for the 81 injection bits and 8 LUTs to extend the sensitive combinational circuits). This means that the extended

![Figure 2: NETFI-2 Architecture](image)

Table 1: Detail of Resource Utilization in the FPGA target

<table>
<thead>
<tr>
<th></th>
<th># Used</th>
<th>Available</th>
<th>% Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>MicroBlaze</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Slide LUTs</td>
<td>1607</td>
<td>63400</td>
<td>2.53 %</td>
</tr>
<tr>
<td>Slide Registers</td>
<td>1599</td>
<td>126800</td>
<td>1.26 %</td>
</tr>
<tr>
<td>MicroBlaze + BM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Slide LUTs</td>
<td>1687</td>
<td>63400</td>
<td>2.66 %</td>
</tr>
<tr>
<td>Slide Registers</td>
<td>1627</td>
<td>126800</td>
<td>1.28 %</td>
</tr>
<tr>
<td>MicroBlaze + BM + Injection SEU</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Slide LUTs</td>
<td>1714</td>
<td>63400</td>
<td>2.70 %</td>
</tr>
<tr>
<td>Slide Registers</td>
<td>1627</td>
<td>126800</td>
<td>1.28 %</td>
</tr>
<tr>
<td>MicroBlaze + BM + Injection SET</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Slide LUTs</td>
<td>1695</td>
<td>63400</td>
<td>2.67 %</td>
</tr>
<tr>
<td>Slide Registers</td>
<td>1629</td>
<td>126800</td>
<td>1.28 %</td>
</tr>
</tbody>
</table>

The results obtained from the fault injection by simulation and emulation are summarized in Table 2. The error rates obtained by both methodologies show a strong correlation. A susceptibility of the BM-Slice circuit was determined slightly higher than 10 % for SEU and 30 % for SET. In other words, the circuit under test has a bit more than 10 % and 30 % chance of providing an erroneous result at the output in case of the presence of a SEU or SET event respectively.

<table>
<thead>
<tr>
<th></th>
<th>Error rate SEU</th>
<th>Error rate SET</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulation (ModelSim)</td>
<td>13.69 %</td>
<td>31.77 %</td>
</tr>
<tr>
<td>Emulation (NETFI-2)</td>
<td>11.30 %</td>
<td>30.10 %</td>
</tr>
</tbody>
</table>

Table 2: Simulation and Emulation Injection Results

In the simulation-based fault injection campaign, a behavior simulation was used disregarding information on propagation delays or timing constraints. The idea was to analyze just the behavior of the DUT when faults were injected on sensitive components. In this case, the results of NETFI-2 can be considered more realistic since they also consider physical effects such as propagation delays and time constraints. Indeed, these values depend on the routing between the logic cells of the target FPGA.

Table 3 details the time measurements made for fault injection in the BM-Slice circuit. The simulations require considerable computational effort even for simple circuits...
such as BM-Slice. In fact, the greater the size of the DUT, the greater the importance of emulation methodologies such as NETFI-2. In this particular case, NETFI-2 allowed an exhaustive fault injection campaign to be carried out in less than one second (i.e., approximately 300 % faster than simulations).

<table>
<thead>
<tr>
<th>Simulation (ModelSim)</th>
<th>Execution Time SEU</th>
<th>Execution Time SET</th>
</tr>
</thead>
<tbody>
<tr>
<td>Emulation (NETFI-2)</td>
<td>&lt; 1 s</td>
<td>&lt; 1 s</td>
</tr>
</tbody>
</table>

Table 3: Simulation and Emulation Injection Execution Times

In conclusion, obtained results of software-based simulations were compared with NETFI-2 verifying that NETFI-2 yields consistent results with those obtained by simulation. During the experiments, it was demonstrated that this new methodology improves the speed of fault injection. Furthermore, it was shown that the area overhead of NETFI-2 is negligible in contrast with previous versions of the tool. The evaluation of the robustness of a LEON-3 processor (widely used in space application) is the most important future work in the development and validation of NETFI-2.

2. References
Delay/Disruption Tolerant Networks Performance and Reliability

Keywords: Delay/disruption tolerant networks, satellite constellations, nanosatellites

Members: A. Charif, J.A. Fraire, M. Solinas, N.E. Zergainoh, R. Velazco

Cooperations: CONAE, JPL, TUD

1. Context and goals
The autonomous transmission of information resources and services through Internet has changed the lifestyle on Earth. Moreover, the potential benefits of extending Internet into space have been analyzed by the community. Nonetheless, the consideration of Internet for space missions has been limited due to fundamental environmental differences. In particular, in a space flight mission, the highly varying communication ranges, the effect of planet rotation and on-board power restrictions compels to face several disruptive situations nonexistent on Internet systems. Furthermore, the propagation delay of signals on Deep Space environments is generally in the order of minutes or even hours. These delay and disruption conditions contraindicate traditional Internet protocol operations as they are largely based on instant flow of information among sending and receiver nodes.

As a result, Delay/Disruption Tolerant Networks (DTNs) have recently been considered as an alternative to extend Internet boundaries into space. In particular, recent studies have considered their applicability in Low-Earth Orbit (LEO) satellite constellations. To overcome link disruptions, DTN nodes temporarily store and carry in-transit data until a suitable next-hop link becomes available. To overcome delays, end-to-end feedback messages are no longer assumed continuous or instantaneous. This distinctive characteristic allows DTN to operate in environments where communications can be challenged by latency, bandwidth, data integrity, and stability issues.

During the last decade, the DTN Bundle protocol, along with different adaptation layers, has been proposed, several routing strategies were studied, and diverse software stacks were publicly released. Furthermore, some of the latter approaches were successfully validated both on LEO and Deep Space missions driven by the UK Space Agency and NASA respectively. Also, DTN has been in pilot studies in the International Space Station (ISS) since 2009 and has been operational on ISS since May of 2016.

In spite of the recent advances in the area, the analysis of the fault tolerance of existing DTN solutions remains an open research topic. Studying reliability of DTN is mandatory before seriously considering its applicability in the harsh space environment where radiation effects, vibrations, collisions, out-gassing among others pose significant challenges for man-made spacecraft.

2. Recent outcomes
The team has provided an extensive fault injection analysis based on two appealing and realistic case studies of delay-tolerant satellite constellations. An initial performance comparison of these topologies is one of the contributions of this work. For the reliability analysis, the state-of-the-art version of Contact Graph Routing (CGR) algorithm was considered. The CGR algorithm was implemented in DtnSim, a new simulator specifically designed to evaluate space DTNs. DtnSim is also introduced in this document and is expected to be available via open-source licensing. Results obtained from DtnSim are the final contribution in this area: an assessment of the fault-tolerance capability of the CGR algorithm in DTN constellations.

Fault model
Over the last years, the semiconductor industry has been particularly concerned by the effects of radiation on integrated circuits and embedded systems in general. The rationale behind this motivation lies not only in the use of these systems in radioactive environments but also in the increasing degree of integration of devices embedded in the same chip. Recent studies have shown that the smaller the feature sizes, the greater the sensitivity to radiation-induced errors. As a consequence, modern embedded systems may be susceptible to low-energy particles including those observed within the Earth’s atmosphere.

This effect is even more dramatic in space missions, which require systems that can operate reliably for long periods of time with little or no maintenance. This is the case in satellite constellations under study in this work. Among the possible errors, transient errors occur in the system temporarily and are usually caused by radiation interference, also known as Single Event Effects (SEEs). The random occurrence in time and space of such failure phenomenon, the probability of the error to happen, and the probability of effectively detecting an unwanted behavior, can be modeled by means of an exponential (Poisson) distribution. The outcome of this kind of failure model is also known as memoryless distribution. The exponential distribution model is the most commonly adopted fault model, mainly due to its simplicity and effectiveness. It takes a single known average failure rate parameter, and can be conveniently described by means of the following equation where lambda is known as the failure or recovery rate, and tau is the time.

\[ F(t) = \int_0^t \lambda e^{-\lambda t} \, dt = 1 - e^{-\lambda t} \]

Adopted exponential distribution model

By means of the configuring lambda with Mean Time to Failure (MTTF) and Mean Time to Repair (MTTR), a fault injection system was designed to study the resulting traffic flow of a DTN based on CGR algorithm under failure conditions.

Simulator Platform

[Image: Simulator Platform]
There exist several tools to evaluate DTNs. Among them, the ONE simulator has been extensively used for DTN studies; however, the platform is specifically designed to model opportunistic DTNs (social networks, vehicular networks, etc.). On the other hand, emulation environments such as CORE are available to directly test existing DTN implementations; but CORE has to be executed in real time, which hinders its application in extensive analysis. To tackle these limitations, a new simulator called DtnSim was implemented in Omnet++, a discrete event network simulator platform.

An indefinite number of DTN nodes can be spanned and configured by a single file in DtnSim. Each of these nodes is based on the layered architecture illustrated in Figure 1. This architecture is a simplification of the original DTN architecture that has been adapted for DTN-based satellite systems and is comprised of an application (APP), network (NET), and Medium Access Control (MAC) layer. Physical layer effects such as bit error rate can be modeled within the MAC layer if required.

The APP layer is the element that generates and consumes user data. In general, in the case of Earth observation missions, a large amount of information is produced in on-board remote sensing instruments and is required to be delivered to a centralized node on Earth. The NET layer is the element in charge of providing delay-tolerant multi-hop transmission (i.e., routes) and is probably the most mature module of DtnSim at the moment. In this layer, each DTN node includes a local storage unit (non-volatile memory model) in order to store in transit bundles. Also, the CGR routing algorithm was implemented as an exchangeable sub-module of the network layer; thus, other algorithms can easily be supplied via a clearly defined interface. Finally, the MAC layer of the DtnSim node is designed to provide a reliable wireless link and to multiplex the shared medium among nodes with wireless interfaces. Also, DtnSim was extended with a fault-injector module based on the exponential model previously described.

Simulation Results
In order to assess CGR behavior under transient failures in realistic delay-tolerant satellite constellations, two appealing and realistic configurations are proposed and discussed below: a sun-synchronous along-track and a Walker-delta formation. Both constellations are based on 16 cross-linked LEO satellites (max. link range of 1000 Km at 500 km height), 25 ground target points, and 6 ground stations. Systems Tool Kit (STK) software was used to propagate these parameters for an analysis period of 24 hs. An intuitive illustration of the node’s location on a world map is provided in Figure 2. The left side of the picture plots the ground tracks of the Walker formation while the along-track is on the right.

Simulation results are plotted in Figure 3. The abscissa axis shows the variation in MTTF including an infinite (INF) value standing for a single simulation execution without the occurrence of failures. Indeed, measurements at this last point in the horizontal axis stand for a reference performance for each of the proposed constellations. For the rest of MTTF values (MTTF from 100 s up to 2200 s were considered with a step of 300 s), resulting metrics are averaged over 160 simulation runs and then averaged over bundles or nodes accordingly. On the other hand, the mean recovery time (MTTR) was set to 5 minutes mimicking a full system reboot. Failures were enabled only in orbiting satellites; ground stations and ground targets were not assumed to fail in this analysis.

The effective fail time curve on the left, shows the accumulated amount of time that a given node in the network refrained from transmitting a bundle either because of a local or a remote (i.e., next-hop node) fault condition. In general, the along-track formation exhibits a higher effective fail rate which is consistent with the higher connectivity of a permanently connected constellation. In other words, the probability of finding a failure during a contact is higher in the along-track than in the Walker system.
A not so intuitive result is shown in the total bundles received curve. This metric measures the quantity of bundles that arrived at the final. The walker constellation is able to deliver the full traffic. However, the along-track formation is unable to deliver such load even without any failures injected in the nodes. After a thorough analysis of the simulation traces, it was found that a pathological routing behavior impeded CGR to find all feasible routes leaving certain bundles without valid routes (i.e. in the limbo). These miscalculations are magnified as the failure rate increases. This was further discussed in [3].

The mean bundle delay curve confirms that the Walker formation provides a better overall bundle delivery time. However, the along-track system results significantly more stable with the variation of the MTTF. The re-routed bundles curves are also expressed in absolute and relative format. These results show the number of bundles that had to be routed while in transit because a contact ended with a non-empty outbound queue. Indeed, the result without failures (infinite MTTF) shows the quantity of re-routing required due to congestion in both constellations. Such an increase can also be justified by the higher delivery rate of this formation. A more detailed analysis of the other results can be found in [3].

Closing Remarks and future perspectives
Results include the first evidence of the performance of Walker and along-track formations under different failure rates. As expected, the higher the failure rate, the more significant the performance degradation. The Walker formation proved to provide better delivery and resource utilization metrics, while the along-track was found more insensitive (i.e., robust) towards faults. Even though the intrinsic redundancy present in the along-track topology favors an improved fault-tolerance, analysis showed that current version of CGR was unable to make an optimal utilization of the communication resources. A proper identification of the algorithm weakness was presented and discussed based on the detailed overview of CGR. The presented analysis becomes a solid starting point towards an improved CGR statement, which is currently under study by the authors. Also, future work includes the exploration of further CGR enhancements that could improve its robustness when implemented in fault-prone DTN systems.

3. References
Irradiation of advanced integrated circuits performed at the high-energy neutron facility GENEPI-2 (Grenoble)

**Keywords:** Accelerated radiation tests, high energy neutron, SEE, SEU, MCU, MBU, SER

**Members:** P. Ramos, V. Vargas, R. Velazco, N.E. Zergainoh

**Cooperation:** KALRAY

1. Context and goals

Ionization resulting from charged particles present in the environment where integrated circuits operate, may result in a wide range of consequences gathered under the acronym SEE (Single Event Effects) and classed as soft and hard errors. Soft errors are the consequences of the modification of memory cell’s contents while hard errors, SEL (Single Event Latchup) may provoke the destruction of the circuit by ground/power-supply short circuits. The soft-error rate (SER) determination is still a challenge to evaluate the technology sensitivity and to extrapolate the trends for future IC’s generations.

A collaboration between TIMA/RIS and KALRAY was started in 2015 aiming at executing radiation test experiments on the MPPA 256-core many-core processor IC. Furthermore, other experiments were performed in similar processors such as Freescale P2041 and Adapteva Epiphany E16G301. Results allowed comparing the reliability of each of these ICs.

In addition, another collaboration between TIMA/RIS and the GHADIR research group in Universidad Complutense de Madrid was started in 2014, whose objectives were the evaluation of the sensitivity against neutrons of Commercial-Off-The-Shelf (COTS) SRAMs. Different tests were performed with 180-nm memories manufactured by Renesas Electronics and Cypress 130-nm and 90-nm ones, both at nominal and low bias voltages.

All these tests were performed GENEPI-2 (GENérateur de NEutrons Pulsé Intense) which is a particle accelerator located at LPSC (Grenoble, France) dedicated to high energy neutrons which are produced under the impact of a deuteron beam onto a tritium (T) or deuterium (D) target by fusion reactions (Figure 1). The beam intensity is measured on target continuously and the 14 MeV neutron production is monitored on line by a silicon detector.

The device under test (DUT) faces the centre of the beam line at a distance adjusted to the desired neutron flux. Typically the DUT is placed to reach a neutron flux of \( \approx 3 \times 10^8 \text{ n.s}^{-1} \text{cm}^{-2} \). Under these conditions, the DUT is exposed to a dose of \( \approx 10^9 \) neutrons within one hour.

Since neutrons propagate in the whole experimental room, the DUT test platform (hardware platform providing the suitable environment to the DUT and the readout electronics) also sits in the neutron flux. Therefore a shield of borated polyethylene was assembled and used to protect it (see Figure 2).

![Figure 1: The GENEPI-2 neutron facility](Image 345x558 to 509x679)

![Figure 2: DUT facing the neutron source](Image 361x359 to 492x457)

GENEPI-2 proved to be an efficient irradiation facility to study the neutron impact on integrated circuits: relevant statistics on induced SEU can be reached within one hour. Obtained results during the tests performed in multi-core and many-core circuits allowed to demonstrate and compare the reliability of modern processors.

2. Recent outcomes

On the one hand, three kinds of processor devices were tested in GENEPI-2:

- The Freescale P2041 processor manufactured in 45nm SOI process technology which integrates four em500c processor cores [1].
- The Adapteva Epiphany E16G301 microprocessor manufactured in 65nm CMOS process which integrates 16 processor cores.
- The Kalray MPPA-256 many-core processor manufactured in 28nm TSMC CMOS technology which integrates 16 compute clusters, each one with 17 VLIW core processors [2-3].
On the other hand, two types of COTS SRAMs were tested in the same facility:

- A memory manufactured by Renesas Electronics (R1LV1616RSA-SSI) in 180-nm CMOS process that has a great robustness against ionizing radiation. This is achieved by means of a particular type of SRAM cells, Advanced Low Power SRAM (A-LPSRAM), which is implemented with a 3D structure instead of a planar one [REF1].
- Two SRAMs manufactured by Cypress Semiconductor in successive 130 and 90-nm CMOS processes (CY62167DV and CY62167EV, respectively).

**Results obtained for the Freescale P2041 processor**

Obtained results from the evaluation of the P2041 multi-core demonstrate that fault injection allows identifying vulnerabilities in the application, and improving the programming strategy for reducing the impact of faults in the results. From the static test, it was confirmed that SOI process technology is more robust than traditional bulk CMOS. On the other hand, dynamic tests have demonstrated that in spite of the parity and ECC protection mechanisms, there were errors in the result of the application caused by MBUs in the address tags and data array. Finally, results show an underestimation of the predicted error-rate, since not all sensitive zones were targeted during the static test and fault injection campaigns. Furthermore, the implementation of ECC and parity in the device’s cache memories may affect the prediction.

**Results obtained for the Epiphany E16G301 processor**

From the evaluation of the Epiphany E16G301 multi-core processor, it can be seen that the proposed extension of the CEU approach was effective for predicting the application error rate. The fact that this device does not implement protection mechanisms has allowed a good estimation of the error-rate, confirming that protection mechanisms affect the testing and error-rate prediction. During the dynamic radiation input matrices were also checked to identify silent faults. It was done in order to obtain the experimental error-rate of the application which has a good correlation with the error rate obtained from fault injection.

**Results obtained for the MPPA-256 processor**

The evaluation of the MPPA-256 many-core processor shows that both, ECC and interleaving implemented in the SMEMs of the clusters are very effective to mitigate SEU type errors, since all the detected SEUs in the SMEMs were corrected during the static test. In addition, dynamic tests have demonstrated that by enabling the cache memories it is possible to gain in application performance without a reliability penalty, since cache memories implement an effective parity protection. Regarding the radiation experimental results, the prediction of the error-rate was based only on the registers’ contribution since they do not implement any protection mechanism. Despite the complexity of this many-core processor, the prediction of the error-rate has a small underestimation that confirms the applicability of the extension of the CEU approach to these devices. The possible reasons for this underestimation are: only accessible registers were targeted, communication infrastructure was not targeted as protection mechanisms may affect the error-rate prediction.

The overall results presented in this work confirm that chip manufacturing process plays a preponderant role in the dependability of the device. A FIT comparison of the studied devices shows that the P2041 multi-core is the most reliable device since it is built in SOI technology. Nevertheless, the difference with the MPPA-256 built in 28nm CMOS is not so ample. In fact, if the FIT/Mb is considered, the MPPA-256 is the most reliable device. Both devices implement ECC and parity in their internal memories. However, the effectiveness of the MPPA-256 is greater than the efficiency of the P2041 due to the implementation of interleaving in its shared memories. Consequently, the selection of the appropriate device should be done based on the requirements of the application to be implemented (e.g., available memory space for code and data, number of cores working in parallel etc.), and a cost-benefit analysis.

**Results obtained for the Renesas 180-nm COTS A-LPSRAM**

This memory was tested with nominal and low bias voltages (from 0.5V to 3.3V). First tests were carried out at nominal voltage as early as in 2014, and no errors were observed when exposing this device against 14-MeV neutrons, at a flux of $3 \times 10^8$ cm$^{-2}$s$^{-1}$. Under these conditions, the LPSRAM was exposed to a dose of $1.1 \times 10^8$ n·cm$^{-2}$ within one hour [4].

In 2015, this memory was radiated at bias voltages between 0.5V to 1V. It is important to note that, even under these conditions, the memory retained information. This time, the neutron dose was $1.6 \times 10^{11}$ n·cm$^{-2}$. It was observed that, when powered up at a voltage above 1V, the memory was immune to SEUs. However, between 0.5V and 0.95V, SEUs and clusters of errors with no more than 128 errors started to occur. It was also interesting to note that hard errors were also provoked, which were visible only at low voltage, even when the memory was no longer exposed to radiation [5].

**Results obtained for the Cypress 130- and 90-nm SRAMs**

These memories were firstly tested under nominal voltage. Several hundreds of bitflips were
observed in both devices under a neutron flux of $3 \times 10^4$ n-cm$^{-2}$s$^{-1}$. These results were used as a case study for the development of a statistical methodology that is able to discern MCUs from the bulk of SEUs in an experiment with a large number of errors.

This methodology is intended for being used in case the underlying layout of the tested part is not available, which is very common in commercial devices. Basically, it studies the bitflips that were observed in a radiation experiment by XORing all the addresses that were affected (the result of XOR two addresses will be called an "element"), then comparing the occurrence of the yielded elements w.r.t. a theoretical situation where only SBUs occur, and attributing the significant differences between these two scenarios to the occurrence of MCUs.

The experiments that were carried out demonstrated that there are three clear hints that point to the existence of MCUs: 1) The existence of elements that are repeated too many times (for instance, it is statistically possible that a given element is repeated 2 or even 3 times in all the experiment, but not 10) 2) The existence of too many elements with low trace (number of 1's) and 3) The fact that a given element complies with any of the previous hints for experiments with different patterns.

Thus, once the candidates are identified, the related addresses are identified and they are merged into the same MCU.

By following these ideas, we developed an algorithmic methodology that proved to be very efficient when extracting MCUs/SBUs in said Cypress SRAMs. Comparisons with MUSCA-SEP3 were also made and were in good concordance with the experimental results [6-7].

Finally, the 90-nm SRAM was also tested under neutron radiation at a flux of $3 \times 10^4$ n-cm$^{-2}$s$^{-1}$, for bias voltages ranging between 0.5V and 3.3V. A clear sensitivity increase was observed for bias voltages below 1V, not only for SBUs, but also for 2-bit to 4-bit MCUs. MCUs were extracted by using the statistical methodology above described.

3. References


