Abstract—The use of commercial electronic components is increasingly attractive for the space domain. This paper discusses the current degree of use of these components in space avionics, the selection and qualification phases to be successfully completed before they can be used, and an overview of the constraints the designers of hardware and software architectures have to face regarding these components, with the corresponding solutions. Concerning the issue of upsets, this paper describes possible solutions at architecture and system level and illustrates them with real examples that have already flown or are being developed. The constraints inherent in space avionics do not allow the total performance range of commercial electronic components to be fully exploited; nevertheless, these components – and particularly microprocessors, on which this paper focuses – are among the technologies having a potential disruptive capability for future space missions.

Keywords—space avionics; commercial electronic components; COTS; performance limitation; fault-tolerant architectures; disruptive technology

I. INTRODUCTION

Performance – in the full meaning of the term – of COTS (Commercial Off-The-Shelf) electronic components is highly attractive compared to their Hi-Rel (High Reliability) / Rad-Tol (Radiation-Tolerant) / Rad-Hard (Radiation-Hardened) counterparts (computing power, integration/features, low power for some models, etc.). But not all these performances can be fully exploited in embedded applications; in particular in the space domain, which accumulates hard real-time constraints, a harsh environment, the impossibility of repairs (whence the strict rules for component qualification), a very high cost per kilogramme in orbit and also very low recurring production (which limits certain investments). This paper will show that designers of hardware and software architectures need to adapt the use of commercial components to the space constraints, generally by sacrificing part of their performance.

Firstly, the current degree of use of COTS components in space avionics will be reviewed, followed by the selection/qualification aspects – the first difficult phase that COTS need to pass.

It will then focus on a key electronic component in computers, the microprocessor. It will look at three areas of performance: bus frequency, power consumption and cache memory; the related constraints and restrictions of use will be detailed.

The main part of the paper will deal with the limitations related to SEE (Single Event Effects) sensitivity and the required protections at architecture and system level.

Examples of computers based on COTS microprocessors that have already flown or are being developed will be used to illustrate the presented constraints and solutions.

Finally, a methodology for the validation of fault-tolerant architectures, with its associated tools, will be proposed.

II. DEGREE OF USE OF COTS ELECTRONIC COMPONENTS IN SPACE AVIONICS

Alongside the Hi-Rel/Rad-Tol components commonly used in the development of space avionics, some COTS components have always been used from time to time (DSP "Digital Signal Processors", microcontrollers, memories, etc.) principally in some scientific missions where the performance and reduction of costs are feasibility factors for the project.

The space industry has compensated for the reduction of the Hi-Rel component market by integrating digital functions into Hi-Rel/Rad-Tol ASICs (in Europe, mainly from ATMEL), and microwave functions in Hi-Rel MMICs, at least for application satellites such as Earth observation or Telecommunications, as well as large scientific satellites with very constraining "Quality Assurance" requirements.

Although the use of COTS components is very limited at the moment, it is nevertheless increasing, particularly when they provide system-level performance that their Hi-Rel/Rad-Tol counterparts cannot achieve (e.g. for analogue to digital converters), and/or where there is a substantial quantity effect (e.g. memory components for mass memory units, or high speed serial links, where thousands of components are required). To take the case of the MYRIADE micro-satellite family, developed by CNES, the French Space Agency, and first launched in 2004, its two embedded computers were mostly developed using COTS components for integrated circuits (except for the Telecommand and Telemetry interfaces which used Hi-Rel/Rad-Tol ones), which enabled unrivalled relative performance regarding volume/mass for this family of satellites.

Nevertheless, it is still unusual for COTS components to be chosen in space avionics for projects having strong "Quality Assurance" requirements; and COTS microprocessors in particular have never been used in these kinds of satellites… until recently.
The following paragraphs give an overview of the steps to be completed before COTS electronic components can be used in space activities, each step bringing a certain limitation to their use. The difficulties will be presented, together with the current or potential solutions for resolving each of them [1].

III. SELECTION AND QUALIFICATION OF COTS ELECTRONIC COMPONENTS

The CNES studies concerning the use of COTS electronic components for onboard applications were first initiated by the Product Assurance Department in 1992. With the shrinking of the space components market and the improved reliability of electronic commercial components, for over more than 15 years now CNES has developed the required expertise to enable large-scale use of COTS components onboard satellites. A methodology [2] [3] to cover all new aspects associated with COTS components from a product assurance perspective was developed in partnership with: EADS-Astrium France, Thales Alenia Space France, Thales AS, EADS-ST, DLR. An overall discussion of this methodology is presented hereafter.

The selection step takes input from multiple expertise fields and, more particularly, is carried through in close collaboration between the Design team and the Product Assurance team. One objective is to choose in priority the "most mature components from major suppliers", which usually are the most reliable and have the longest life cycle.

The goal of the procurement step is to obtain a single lot of components; this is indispensable because some qualification tests either generate some stresses or are destructive. It should be remembered that the date code indicates a single manufacturing date for encapsulation, but in no way guarantees a single foundry lot; only the manufacturer can undertake to provide a single foundry lot. Thus, it is obviously difficult to negotiate this type of service with very large mass manufacturers for the low volumes used in the space sector. Consequently, not all COTS components can necessarily be used in space.

The difficulty of the next step, i.e. validation/qualification of the lot, will depend on the kind of the user project. Let us first consider the needs for projects having strong "Quality Assurance" requirements. The qualification or Lot Acceptance Tests (LAT) step involves tests that are fairly conventional in all types of embedded applications: electrical characterisation at 3 temperatures, construction analysis – DPA (Destructive Physical Analysis), damp heat tests (e.g. HAST "Highly Accelerated Stress Tests"), life-test, etc. Other tests are linked to some problems that are common to both Hi-Rel and COTS components, but to which COTS are more sensitive, e.g.: thermal cycling tests (silicon and plastic packages having very different dilatation coefficients), the risk of 'whiskers' (intermetallic growths) that can generate short-circuits, the risk of 'purple plague' (a kind of corrosion) that can lead to debonding, etc. Finally, tests specific to the space domain are required, related to the environment or mission requirements: TID (Total Integrated Dose), SEL (Single EventLatch-up), SEU (Single Event Upset), ON/OFF test specific to mission for which it is important to optimise the power consumption as Earth observation payloads that can require 100,000 power cycles, etc.

This LAT step authorises to solder this lot of components for 7 years from the date code. For such components to be reused at a later date for recurrent production (as it is currently the case for the image chain of the PLEIADES satellite whose high speed serial link component – the commercial GLink from Agilent, now obsolete – with a date code of 2002), it is required to conduct "relieting" tests to verify whether the lot has not been degraded during storage (corrosion of pins that would make difficult to solder them, corrosion of bondings causing mechanical weakness, etc.). Relieting authorises to solder such components for another 3 years. The lot cannot be used after this period of 7 + 3 years for space developments. The very long cycle times of space projects means that 'strategic storage' (a solution among others against obsolescence) should be accompanied by monitoring of aging of the stock, and regenerating it if required.

COTS components can never be used completely "as is" in the space context. For projects having lower "Quality Assurance" requirements, the number of tests can be considerably reduced; the objective is to find the best trade-off between cost and quality. Some tests remain mandatory: SEL, DPA, etc.; also SEU for a few key components (processors, memories, etc.).

IV. PLACING LIMITATIONS ON THE USEABLE PERFORMANCE OF COTS ELECTRONIC COMPONENTS

Functional and performance limitations will have to be taken into account for COTS electronics components having successfully passed the selection and qualification steps, so that they can be used in a computer onboard a satellite. Here are examples from three different fields.

A. Bus frequency

On the one hand, in the European space industry, the usual frequency of a parallel bus carrying a data flow from imaging sensors is about 100 MHz: partly in order to be compatible with Rad-Tol ASICs, partly to avoid the need for double-ended links. For specific needs, however, higher performance may be required: a 300 MHz parallel point-to-point bus has been developed for one European space payload.

On the other hand, COTS microprocessors may have 64-bit buses running at hundreds of MHz, which is obviously well beyond the usual scope of space developments. Such buses also require memories capable of sustaining these data-rates; here again, Hi-Rel memory components (exclusively SRAM), or even COTS SDRAM, will be unable to match high commercial frequencies. DDR2 memories have not yet been qualified for space use in Europe; they require delicate interfacing.

Therefore, for European space projects it will probably be necessary to limit the frequency of the processor bus, for example between 50 and 100 MHz for the PowerPC7448 microprocessor from Freescale which has a bus with a maximum of 200 MHz. This can increase the bottleneck that often occurs at the memory bus, thus reducing the computing power available for the mission.
B. Power consumption

The usual power consumption of a Hi-Rel command-control computer for the platform of a large satellite is about 30 to 40 W; e.g. less than 40 W (average) for PLEIADES, a CNES Earth observation satellite to be launched in 2010.

On the other hand, an estimate of the maximum power dissipation of a printed-circuit board in double-Europe format shows that 15 to 20 W seems to be a reasonable objective that should not be exceeded in order to avoid the need for a complex mechanical interface and cooling system (due to a purely radiative and conductive heat transfer into a satellite).

However, a microprocessor such as that of the latest generation of single-core PowerPC, the PPC7448 mentioned above (SOI 90 nm implementing the Power Architecture™) consumes as much as 30 W alone in a worst case scenario (maximum frequency for both bus and core)! To this must be added an ASIC companion-chip, a memory array, I/Os, etc. It is therefore obvious that for such a component, it is difficult to use the maximum possible frequency in space applications.

One example is GAIA [4], the Billion Star Surveyor from ESA (European Space Agency), a satellite planned for launch in 2012. GAIA uses a processor board from MAXWELL Technologies based on the commercial PowerPC750FX from IBM (see Section V-H); its power consumption is 20 W (typical) at maximum frequency. For GAIA, the frequency of the core of the PowerPCs has been reduced in order to reduce power consumption (which does not significantly slow down data processing in this case because the bottleneck for that project occurs mainly at the memory bandwidth level).

C. Cache memory

Since the external memory array cannot follow the rhythm imposed by the frequencies of the cores in modern processors, cache memories are essential so as not to slow down the core excessively. However, for embedded computers having hard real-time constraints, such as in command-control computers, cache memories generate several difficulties:

• Software is less deterministic: depending on whether the cache is full or not (requiring or not a flush sequence), execution times vary. Such an effect on 'real-time' interrupts can result in time jitter on all software tasks, which is a strong difficulty with regard to hard real-time constraints.

• Less deterministic software is more difficult to validate: there are more combinations of possible cases and configurations to be tested, and it is more complex to define the configurations and ways of activating them.

• Since cache memories are internal, if they do not contain an EDAC (Error Detection And Correction) module, they are difficult to protect against SEEs.

There are several possible ways of solving these problems by bounding the functional use of cache memories, without completely cancelling the benefits they can bring to performance. For example:

• Static management of the cache: pre-load the caches with software modules whose processing time requires optimisation (some alarm procedures, functions frequently called up, etc.), then lock the caches which will then only serve as 'fast internal memories'. The modules not loaded in the caches then run only in external memory, i.e. less rapidly.

• Deny the use of cache memory to software modules whose determinism is essential (interrupt procedures, etc.); for these modules, the running time is then longer but deterministic. All the other modules (algorithmic data processing, etc.) then have access to cache-related performance.

• Work in 'write-through' mode (instead of the usual 'copy-back' mode), which does away with the need to flush the memory and the indeterminism this generates; then only 'read' accesses are faster because of the caches, which considerably slows down computing power.

In Europe, the space community currently does not use cache memories; the first Rad-Tol microprocessor including caches, the LEON2 from ATMEL, has not yet flown. We must therefore take an example from aeronautics: on the Airbus A380 'fly-by-wire' computer it was necessary to limit the use of the cache memory of the PowerPC MPC755 microprocessor from Motorola by managing part of it statically, so that only part remains usable dynamically. A large amount of work was also invested in estimating the WCET (Worst-Case Execution Time) so that the software could be certified to the DO-178B standard [5].

In the space domain, the problem remains of the sensitivity of cache memories to SEEs. In this respect, recent microprocessors, that are starting to include parity or EDAC in their cache memories (such as the PPC7448), are particularly more attractive, on condition that all bits of the cache memory are protected (data bits and tag bits) and that the implemented protection code is effective enough concerning the sensitivity of memory cells with regard to the space environment (presence or not of MBU "Multiple Bit Upsets").

V. HOW TO COPE WITH SEE AT SYSTEM LEVEL?

Last but not least, the use of COTS forces designers to take into account their sensitivity to SEEs:

• either the microprocessor has low sensitivity and/or the mission constraints are low (mildly aggressive environment, mission with low availability specifications), in which case usual simple protection mechanisms are sufficient (watchdog, etc.);

• or the microprocessor is more sensitive and/or the mission constraints are stronger, in which case it is required to protect the computer at architecture and system level.

It should be noted that a part of the implemented computing power will be used for SEE mitigation, and that it could have a non-negligible impact on mass and power consumption.

This Section will review the available solutions and illustrate them with real examples.
A. Mix of different elementary protection mechanisms

Some methods have been developed as ABFT (Algorithm-Based Fault Tolerance) for space [6], BIST (Built-In Self Test), signature analysis with watchdog processors, wrappers, etc.

The computer of the NASA SMEX (SMall EXplorer) mini-satellite family (first launch in 1992) is based on the INTEL 80386/387 couple protected by a set of simple SIFT (Software Implemented Fault-Tolerant) mechanisms [7] (watchdog, ...). MYRIADE implements several WDs (WatchDog) including micro-latchup and processor hangs are listed hereafter; MYRIADE implements several WDs (WatchDog) including one that acts at several levels:

- The internal PIC WD is set to 100 ms (each I/O block is constituted by a PIC nanocontroller and interface components).
- A global WD for each I/O block is set to 250 ms.
- A local WD for the microprocessor (i.e. Transputer T805) is set to 500 ms.
- A global WD for the computer is set to 1 sec with four levels of actions having a deeper and deeper effect on the computer (if the first action is not successful, the second one is attempted, and so on):
  - Transputer reset.
  - Transputer power cycling (in case of SEL).
  - Processor board power cycling (at this level, the Transputer memory content is lost).
  - Computer power cycling (in order to passivate any residual SEL).

B. Rationale for duplex architectures

The Bi-MR (Bi-Modular Redundancy), or duplex architecture, replicates the processing channels then compares the results of the two channels.

It is mainly a fail-stop architecture, since a duplex can not intrinsically identify which of the channels is faulty. Thus, extra mechanisms are required for diagnosis and recovery.

C. Time replication at applicative task level

The time replication architecture family consists in executing twice successively the same piece of software on the same processor in a time multiplexed way.

The low-cost DMT (Duplex Multiplexed in Time) CNES patented architecture aims at, but is not limited to, scientific missions and small satellites [9] [10].

DMT is macro-granularity oriented: each applicative task (e.g. satellite thermal control, attitude and orbit control) is executed twice successively (i.e. two "virtual channels").

The DMT error detection is based on a bit-to-bit comparison (thanks to acquisition consistency) at the end of each task. Only the main results are compared: commands to actuators, parameters to other tasks, current context. The large amount of local variables are not checked, thus reducing the data to be compared to few items e.g. in case of a command-control computer, and minimizing the associated overheads.

The DMT recovery principle is checkpointing oriented, associated with specific mechanisms. A safe context storage independent for each virtual channel is available, thanks to an hardware support mechanism – called CESAM – to be implemented inside a FPGA or an ASIC designed to be SEE-free; thus, a dedicated 'safe memory' is not required.

The DMT architecture has been validated during the CNES TAFT1 R&D study completed in 2005.

D. Time replication at instruction level

The time replication architecture family also includes micro-granularity solutions: each instruction is performed twice successively on the same processor, then a compare and a conditional branch (in case of results mismatch) are performed before writing back the result in memory. A classical implementation of such a solution is costly from a memory expansion and an execution time point of view. The three following approaches have been developed in order to help to reduce these overheads.

A first approach of such a technique has been developed by the Politecnico di Torino and the TIMA labs [11].

A second approach consists in a solution called EDDI (Error Detection by Duplicated Instructions) developed by Stanford University, that has flown on the USAF ARGOS large satellite (launched in 1999) [12]. The EDDI computer is based on the COTS IDT-3081 processor (R3000 instruction set). A technique called CFSS (Control Flow Checking by Software Signatures) allows to enhance the control-flow error detection. Over a 350-day period in orbit, 321 errors have been detected and 98,7 % have been corrected.

A third approach consists in a solution called TTMR (Time Triple Modular Redundancy), that has been implemented into the Proton100k computer developed by SPACE MICRO Inc. [13], that has flown on the USAF Roadrunner experimental small satellite (launched in 2006) and on the ISS (International Space Station). It runs in a time duplex mode when no error is detected, and in a time triplex mode for recovery. It takes advantage of DSPs having VLIW (Very Long Instruction Word) parallelism for reducing the time processing overhead. The SEFI mitigation is included in a Rad-Tol FPGA. Other products have been developed by Space Micro Inc., including one based on a dual-core PowerPC.

E. Structural duplex

The Bi-MR, or structural duplex, is a more conventional solution, hereafter illustrated by several space developments.

The simplest Bi-MR architecture is the Master-Checker concept which consists in two microprocessors connected to the same memory and working in lock-step mode (both microprocessors access to the same data in the memory exactly at the same clock cycle).

The low-cost and high performance DT2 (Double Duplex Tolerant to Transients) CNES patented architecture aims primarily at application missions and large satellites [14].
The DT2 is a "structural mini-duplex": the hardware duplication is limited to the PUC (Processing Unit Core, i.e., the microprocessor, its memory and CESAM); thus lowering the hardware cost of the DT2. The compare function and I/O coupler can be implemented inside a FPGA or an ASIC – called SYCLOPES – designed to be SEE-free.

The main version of the DT2 is macro-synchronized: each PUC is working independently, and a synchronization process between them is achieved only during I/O phases.

The recovery strategy is the same than the DMT one. For instance, in case of backup recovery, both PUCs simultaneously roll-back without requiring communication between themselves, thanks to the fact that each PUC has its own safe context storage.

The DT2 architecture has been partially validated during the TAFT1 study, and the current CNES TAFT2 R&D study will validate the DMT-NG (New Generation) and the full DT2 architectures [15].

In the launcher domain, the Ariane 5 telemetry generation unit called UCTM-C/D developed by IN-SNEC (first launch in 1996), is a double-duplex based on a DSP not immune to radiation. When the two channels of the master duplex disagree, the second duplex becomes the master and generates the telemetry; then the first duplex is reinitialized without requiring a software context from the current master duplex due to the kind of algorithms used for generating telemetry.

The platform computer of the DLR (Germany) BIRD micro-satellite (launched in 2001) is based on a PowerPC MPC623 micro-controller [16]. There are four PE (Processing Element) channels, two channels are switched-off and used as spares in case of transient or permanent failure; moreover, they age less with regard to total integrated dose. The two switched-on channels operate in Master-Checker mode, with the 'Worker' controlling the satellite, while the 'Supervisor' checks the Worker. If one channel detects an anomaly on the other channel, it forces it to start a recovery procedure while the Worker is working independently, and a synchronization process between them is achieved only during I/O phases.

Concerning the TMR architecture, several studies (as [17]) or developments have been made for the space domain based on Rad-Tol components and are mainly a protection against failure for safety-critical missions, such as man manned ones. Nevertheless, the know-how resulting from these developments could be re-used when using COTS components. Developments targeting e.g. the Hermès European space shuttle project (EADS-Astrium, LAAS-CNRS, CNES) [18], as the USR quadruplex breadboard developed by EADS-Astrium for CNES, have resulted in the production of the DMS-R (EADS-Astrium), two triplex computers for the Russian module on the ISS launched in 2000. A FTC triplex system (EADS-ST) based on the DMS-R CPU board and on a specific inter-channel network has been developed for the ESA ATV (Automated Transfer Vehicle servicing the ISS, first launch in 2008) [19].

The computer of the ISAS-JAXA (Japan) REIMEI (INDEX) micro-satellite (launched in 2005), is based on an Hitachi SH-3 commercial micro-controller protected by a 'light' version of a triplex architecture (centralized voter integrated into a space qualified FPGA); the computer is stopped for the faulty channel reinsertion phase during about 2 seconds [20].

The computers of the USEF (Japan) SERVIS-I (launched in 2003) [21] and SERVIS-2 (launched in 2009) satellites are based on COTS microprocessors. Onboard SERVIS-2, the patented CRAFTSYSTEM (Compare and Rational Abort for Fault-Tolerant System) based computer is implemented in a triple redundant way; but instead using a single centralized voter which is a single point failure weakness, the three PEs are connected through a specific ring-type anomaly detection network allowing firstly comparison result of each PE to be transferred to adjacent PEs, and secondly to declare one of the PEs as 'Master' (having the right to output data outside the computer) and the others as 'Checkers' [22].

Another alternative is to N-plicate only the microprocessor itself (neither the companion chip, nor the memory), and to micro-synchronize them in a lock-step architecture.

The franco-american CALIPSO mission (launched in 2006), based on a CNES PROTEUS mini-satellite platform, includes a payload computer developed by General Dynamics Advanced Information Systems (GDAIS) [23]. This computer is based on a set of four PowerPC603r running in lock-step and working in quadruplex mode, and on a single voter implemented in a redundant Rad-Hard ASIC.

The SC5750 space-qualified board has been developed by MAXWELL Technologies, based on a set of three IBM PowerPC750FX micro-synchronized microprocessors working in TMR mode [24]. The centralized voter is included in a Rad-Tol FPGA that is immune to SEE. Due to the single centralized voter, the SC5750 board is not tolerant to a permanent failure, thus it must be duplicated (cold redundancy). The ESA Gaia...
mission yet mentionned implements seven SCS750 boards (launch planned in 2012), and the two satellites US constellation NPOESS dedicated to civil/military weather forecasting implements four SCS750 boards per satellite (launch planned in 2014).

But lot of difficulties are linked to the lock-step mode [25]. On the one hand, generally speaking there is no formal commitment from the microprocessor manufacturer concerning the lock-step mode and it is difficult to prove the correctness of this mode. On the other hand, the lock-step mode requires microprocessors having fully deterministic timing which is less and less feasible with deep submicronic technologies (e.g. microprocessors will include low-level fix-up routines to tolerate timing violations and soft-errors), making micro-synchronization more and more difficult. Thus, the lock-step mode seems not to be a trend for future space avionics.

VI. VALIDATION

With regard to Hi-Rel/Rad-Tol based computers, fault-tolerant mechanisms for COTS components protection are additional functions. Thus, they require to put in place a methodology and it associated tools for validating these added mechanisms.

A methodology for the validation of fault-tolerant architectures and appropriate tools for injecting faults were proposed and developed as part of the CNES TAFT1 study yet mentioned [26] and have recently been improved for the TAFT2 study. Although they have not yet been either standardised or produced industrially, they are precursors of the solutions that will be required for validating these kind of architectures. They are briefly described here after.

This validation methodology targets more specifically fault-tolerant architectures which are used for the first time by the space industry (i.e. with a strong debugging aspect). Obviously, for a second use of the same fault-tolerant architectures, the validation process could be lightened.

The validation methodology selected by CNES is based on a pragmatic approach; this hybrid methodology combines two deterministic injection phases and one random injection phase. Each phase has a precise and complementary objective.

The first phase is based on deterministic fault injection; it is oriented 'debugging' (as 'unit tests') of fault tolerance mechanisms. Pattern generation is done manually and, because it is an important effort, this first phase is limited to few hundreds of errors; the goal being to inject some errors for each class of faults. A classification system was therefore adopted on the one hand to obtain a relatively limited number of injection patterns per class for easy handling, and on the other hand to try to be exhaustive concerning the type of fault effect at the system level. The classification is partly depending on the chosen fault-tolerant architecture (e.g. the reaction of DMT/DT2 architectures at the system level is the same for a given type of fault whatever is e.g. its datation inside a given applicative task) and its definition is based on an analysis of the 'fault types', 'similarity of architecture reaction', and so on. During this first phase, the observability must be at the maximum level.

The second phase is based also on deterministic fault injection; it is oriented 'validation' (i.e. 'intensive test') of fault tolerance mechanisms, in order to fully validate and, if needed, to improve them. This step is based on pseudo-random fault injection, generation of injection patterns being manually "guided". The goal of this second phase is to inject about ten thousands of errors. During this second phase, the observability is kept at its minimum level to reduce the analysis work, but, thanks to the deterministic aspect of this phase, if required it is possible to play again a given vector with full observability.

The third phase is based on random fault injection; it is oriented 'global end-to-end validation' and 'error recovery rate measure' with near real space environment conditions, thanks to the usage of laser [27] and heavy ion facilities. And it is an efficient way to obtain an availability rate of a COTS-based computer including fault tolerance protections. With regard to a heavy-ion beam, a laser beam allows finer targeting on devices under test, specifically chip areas where the most critical errors occur (e.g. instruction errors, control errors). As the result of feedback from the TAFT1 study, CNES proposes gradual and incremental validation based firstly on several laser tests distributed over the development period (in order to avoid the need to validate all the mechanisms on a single occasion on a full and complex application software), and secondly on an overall and final validation using a heavy-ion test. Laser tests can, at least partially, replace the second phase. The goal of this third phase is to inject few thousands of errors. During this third phase, the observability is kept at its minimum level.

It should be noted that with modern microprocessors incorporating millions of cells sensitive to SEEs, only a small percentage of these cells will have undergone error injection during the validation activity; furthermore, the ratio of 'control cells' (e.g. sensitive to SEFI "Single Event Functional Interrupts") to 'data cells' (partially cache memories) is significantly reduced. This raises the question for the space community of what exact level of validation has been reached. To avoid excessively time-consuming error-injection phases, the following points need to be considered. For one thing, for the first and second phases as proposed here above, i.e. deterministic injection (laboratory tests), a classification of errors must first be done so that only a small number of errors need to be injected for each class; this can at least be done with DMT and DT2 architectures. In addition, for the third phase of random injection (beam tests), the number of SEEs injected must be greater than the estimated number of in-orbit events by a factor K, with K being at least 100. In this case, the laser is an efficient tool, allowing the critical areas to be more targeted than purely data areas. This issue remains an open question.

During CNES TAFT1 and TAFT2 studies, several tools have been developed or adapted, particularly TRIAGES which is a generic and reusable automatic injection results analysis tool. Its operating mode is based on a learning process using 'golden runs'; the first and second phases described above allow inherently to generate the data base for TRIAGES without requiring extra work. A software implemented fault injection tool developed by TIMA Laboratory has also been adapted.

VII. CONCLUSION

It is easy to see that the 'cost of ownership' of COTS components (procurement cost + cost of lot acceptance test + cost
of fault-tolerant mechanisms and their validation) is potentially (far) higher than for their Hi-Rel/Rad-Tol counterparts (apart from some rare components used in very large quantities, see Section II). Thus, only the contribution of COTS components to system performance can justify their use in space avionics.

On the one hand, there is a growing gap between the performance of commercial electronic components and that of their Rad-Tol counterparts. On the other, the full range of performance of COTS electronic components is not fully accessible in the space context. Different constraints have been listed here showing why the designers of hardware and software architectures need to adapt their approach when using COTS components and, in most cases, have to accept reduced performance.

Concerning the issue of SEEs, a set of fault-tolerant architectures have been reviewed, with practical examples either already flying or under development. It should be noted that there is no convergence towards a unique architecture, as the search for an optimum solution for each space project takes precedence over other factors.

However, the use of commercial electronic components in space avionics may become more widespread in the years to come. Indeed, their high performance means that they are likely to become a disruptive technology and make feasible some ambitious projects, as it is the case with the GAIA and NPOESS missions currently being developed.

REFERENCES


