JOURNAL OF SPACECRAFT AND ROCKETS
Vol. 39, No. 4, July–August 2002

New Methodology for Simulation of Soft Errors in Digital Processors

Sana Rezgui* and Raoul Velazco†
Centre National de la Recherche Scientifique, 38031 Grenoble, France
Santiago Rodríguez‡
Instituto Nacional de Técnica Aerospacial, 28850 Torrejón de Ardoz, Spain
and
Robert Ecoffet§
Centre National d'Etudes Spatiales, 31055 Toulouse, France

Development and implementation are presented of a new fault injection technique for error rate prediction of processor-based digital architectures operating under radiation. Bit flips are injected in potentially sensitive memory locations concurrently with the execution of a program. The error rate derived from those fault injection experiments and the underlying cross section of the studied processor will allow the cross section estimation of this circuit running a given program. A comparative analysis of experimental results issued from both soft error injection and ground testing under radiation performed on a board built around a microprocessor demonstrates the efficiency of this new technique to predict the error rate of an application.

Introduction

WITH the reduction of transistor features, radiation effects on complex integrated processors become a major concern. Among these effects, bit flips resulting from ionization caused by charged particles hitting the circuit are considered critical because of their random occurrence. In particular, digital circuits operating in space are subject to different kinds of radiation, whose effects can be either permanent or transient.[1] The former are the result of trapped charges at the silicon/oxide interfaces. The latter may be caused by the impact of a single particle [single-event effects (SEE)] on sensitive circuit nodes. Depending on the impact dynamics, two kinds of SEEs are distinguished and considered in this paper: single-event upsets (SEUs) and single-event latchups (SELs). SEUs are responsible for transient changes in bits of information stored within an integrated circuit. These errors, that is, the consequences of SEUs, will be called upsets or bit flips in the rest of the paper. SELs result from the triggering of parasitic silicon controlled rectifiers (SCRs) (present in complementary metal-oxide semiconductor technologies) producing short circuits between the voltage drain drain (VDD) and the ground and may destroy the circuit if not powered off in time. On the other hand, the consequences of upsets depend on the nature of the perturbed information, ranging from erroneous results to system crashes.

The increasing demand of high dependability and reliability of safety-critical systems (spacecraft, satellites, etc.) requires development of methods suitable for the qualification of microprocessor-based boards, to predict error rate of flight software in the final environment. Methods usually adopted to perform SEU ground testing of processors differ in the way the circuit is exercised while exposed to the particle beam. The so-called register testing corresponds to a static strategy, in which the whole observable SEU-sensitive area (internal registers and memory) is continuously observed. An alternative approach, sometimes called dynamic testing, is to run simple programs on the processor and to observe only the program results, to activate the sensitive areas in a way that is closer to the final application.

The results of these two tests can be reported as the register-bit cross section, which is the underlying SEU cross section and the application error rate. Traditionally, the underlying SEU cross section in a processor is assumed to be a direct measure of the rate of observable errors induced by SEUs in a system. This assumption supposes that every SEU arising in a processor's memory cell induces errors in the program results. This interpretation is, in fact, the worst-case situation. Indeed, SEUs will induce observable errors only if they occur in a target during its sensitivity period, called the duty cycle.[2] The error rate of a particular application running on a microprocessor depends on these duty cycles. A practical cost-effective method to determine the observable SEU-induced error rate for different softwares has been developed and published.[3] The error rate τ_{SEU} of an application is defined as the weighted sum of the individual registers cross section, σ_i, where the weighted factor f_i of a register is its associated duty cycle. That is,

$$\tau_A = \sum_{i=1}^{n} \sigma_i f_i \tag{1}$$

where n is the total number of registers in the microprocessor.

Ground-test data[4] (issued from particle accelerator facilities) proved that the observable SEU-induced error rate is in fact some fraction of the underlying SEU cross section and depends on the executed software. In Refs. 3–5, a benchmark of simple programs [matrix multiplication, fast Fourier transform (FFT), sorting algorithm, etc.] was used to characterize the SEU vulnerability of the Harris H80C85 and the Motorola 68020 microprocessors. In spite of the simplicity of the tested programs, obtained results have shown significant differences in the observable SEU-induced error rates, which puts in evidence the dependence of the application cross section on the studied program.

This dependence on executed software calls for accurate studies of the behavior of microprocessor-based boards under radiation. Such a study should involve the investigation of the operation of the microprocessor in the presence of faults. Considering that the mean time between faults in radiation environments is long and ground

Received 11 December 2000; revision received 30 January 2002; accepted for publication 1 May 2002.

*Teacher and Research Assistant, Circuit Qualification Group, Techniques of Informatics and Microelectronics for Computer Architecture Laboratory; sana.rezgui@imag.fr.

†Leader, Circuit Qualification Group, and Director of Research, Techniques of Informatics and Microelectronics for Computer Architecture Laboratory; raoul.velazco@imag.fr.

‡Head, Atmospheric Instrumentation Laboratory, Department of Earth Observation, Teledetection, and Atmosphere; rodriguezs@inta.es.

§Leader, Department of Space Environment and Radiation Effects; robert.ecoffet@cnes.fr.

tests are expensive, several techniques of fault injection, presented in Refs. 6–9, have been investigated to evaluate the error rate of different applications.

The state of the art shows the existence of few techniques for upset fault injection, using software or hardware tools. In fact, it has been demonstrated in Ref. 6 that a low cost, readily available, commercial off-the-shelf (COTS) microprocessor simulator can be adapted to inject bit flips in sensitive areas of a studied processor. This technique has been applied to the 80C51 microcontroller running two different applications by means of a simulator (developed mainly for software debugging purposes). The difficulties for both the automation and the generalization of this tool put in evidence the limitations of this technique. In fact, the inability to automate the simulation procedure limits considerably the speed per simulation cycle. Moreover, it was necessary to understand the detailed design of both the tested software and the microprocessor system to be able to use the simulation model efficiently.

Another method, Xception, made possible the injection of upsets.[7] This method is based on the generation of hardware exceptions randomly, in time and location, in the target processor. This software/hardware technique allows the injection of faults with minimum interference with the target application, which is not modified and is running at full speed. The main advantages of this technique are the possibility to inject faults during the execution cycle of one instruction, the low cost, and the low intrusiveness. In fact, implementing this technique supposes that the studied processor is equipped with special exception handlers to perform transient fault injection, which constitutes a limitation in migrating the method to others processors. To our knowledge, this technique has been implemented only on the PowerPC 601 microprocessor. A quite similar approach for transient error injection based on interruptions was recently mentioned within a list of suitable fault injection methods for the validation of safety-critical applications.[8] To our knowledge, no results of its application to actual circuits have been published as yet.

This paper presents a new strategy to characterize and to quantify the effects of upsets on the operation of microprocessor-based digital architectures. This technique was developed at the Techniques of Informatics and Microelectronics for Computer Architecture Laboratory (TIMA), for upset fault injection concurrently with the program execution on the tested processor. This fault injection approach makes possible the quantification of the rate of effective upsets for the tested programs and, thus, the estimation of realistic figures for the expected error rate in flight. Such experiments could lead to a sound methodology for the final application error rate estimation, based on both limited radiation testing (to evaluate the SEU cross section per register type) and fault injection experiments to evaluate statistically the fraction of upsets with consequences for the program execution.

In a previous work, we described the implementation of this technique and its validation for two different digital architectures,[9] based on the 80C51 microcontroller from Intel and the TMS320C50 Digital Signal Processor from Texas Instruments, respectively. Obtained results proved that the method leads, for the studied processors, to error rate predictions in very good agreement with error rates issued from radiation ground testing. The present work presents experimental data derived from upset fault injection in the accessible sensitive areas of the Thomson TS6833216A microprocessor. These results were obtained for a digital architecture built around this processor, using for ground-testing purposes both simple benchmark programs and flight software developed for a satellite project. In fact, this microprocessor is included in an instrument of CESAR, a satellite project from National Institute for Aerospace Techniques (INTA) of Spain. The comparison of the obtained SEU error rates to those derived from the commonly used strategy would allow objective conclusions to be made about the resulting deviations in measured error rates.

Main features of the operating modes of a dedicated testbed used to implement the proposed fault injection approach are detailed in this paper. Results derived from fault injection experiments carried out concurrently with the execution of the studied programs and ground-testing experiments are discussed. From those results, the sensitivity of an instrument based on this processor running the flight software is predicted.

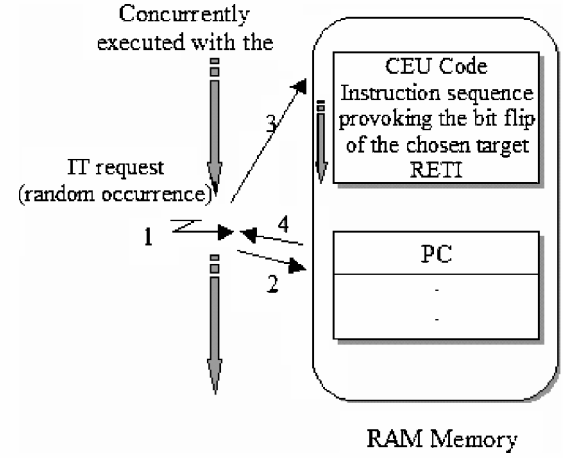

Fig. 1 Soft error injection by means of interruptions.

Fault Injection Methodology

The approach relies on injection of bit flips, randomly in time and location, concurrently with the execution of a program. This fault injection method can be achieved with minimal "intrusiveness" by software/hardware means, using the interrupt mechanism. In fact, implementing this method supposes that the tested application is a processor-based digital board, organized around a device capable of executing instruction sequences and able to handle asynchronous signals (interruptions). The key idea is the generation and storage at an appropriate memory address, of a piece of code called code emulating an upset (CEU), the execution of which will provoke content inversion of the selected bit (CEU target). If the processor is properly configured, the CEU-code execution can be triggered by the assertion of an interruptlike signal, as shown in Fig. 1. The interrupt activation time and the CEU target can be pseudorandomly chosen by an ad hoc external mechanism. In this way, bit flips may be injected in all of the processor's accessible CEU targets [internal registers and static random access memory (SRAM) memory area] as well as in the external SRAM where program data and code are stored. Note that the CEU code may include instruction sequences to read, modify, and overwrite values stored in the stack. Hence, it is possible to inject CEUs on critical control registers (program counter, stack pointer, status registers, etc.), which are often not directly accessible by the instruction set.

The main advantages of this fault injection strategy are the reduced intrusiveness in the system, the low costs, the possibilities of automation, and the flexibility. Nevertheless, two limitations of the CEU injection approach must be mentioned: 1) Because interruptions are always taken into account at predetermined fixed instants, the effects of SEUs occurring during instruction execution can not be simulated. 2) Not all possible sensitive targets can be reached. In spite of these limitations, we assume that the performance of modern processors and their huge internal memory space make the accessible area represent a significant proportion of the total sensitive area, giving some significance to the results of the proposed approach.

The implementation of the proposed fault injection approach requires extra hardware to load the memory with data corresponding to the desired CEU code, to trigger the interrupt signal and to compare the program execution time and outputs to expected values. The architecture of a dedicated test system, called the testbed for harsh environment studies on integrated circuits (THESIC), developed at TIMA for SEU ground-testing purposes,[10] offered a suitable platform for the CEU injection. Indeed, THESIC is organized in two boards: a motherboard for both the control of test operation under radiation and user-interface purposes and a daughterboard for the adaptation of the device under test (DUT) to the motherboard bus protocol (Fig. 2). The communication between the two boards is achieved in an asynchronous way through a common memory area, called the memory mapped interface (MMI). During a test, the DUT indicates when the MMI area has data to be transferred by sending an

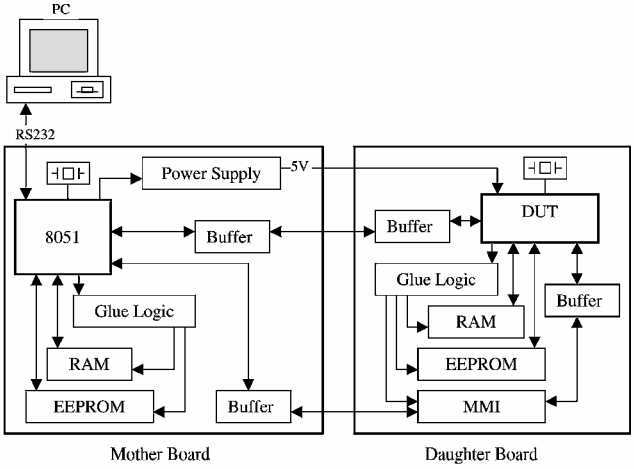

Fig. 2 THESIC experimental setup.

interrupt to the motherboard. When this happens, the motherboard interrupts the DUT board to read the results and, thus, to detect eventual errors. Cases where we do not get any answer from the processor are classified as sequence loss errors. To cope with those errors, a programmable software watchdog was implemented in the motherboard.

The THESIC motherboard was enhanced with pseudorandom interrupt generation capabilities and a new operation mode providing different options for CEU injection. With these options, the selection of the two parameters of simulated upsets, the bit location to be corrupted and the instant of fault occurrence, can be chosen either pseudorandomly or deterministically. This flexibility appears to be very useful for the investigation of the effects of upsets in complex applications. For instance, repeated experiments with pseudorandom choice for both the CEU target and the occurrence instant, allows objective statistics on the fraction of upsets that have no effects for a given program. Moreover, the deterministic choice of the CEU parameters (occurrence instant and bit location) may provide information on the most sequence loss upsets when running a given program.

The CEU injection technique allows us to perform exhaustive bit flip injection experiments in a particular register or in an internal memory position during the studied program execution. This allows us to simulate the effect of an upset in a particular target during each instruction of the program. If the considered target is used (read), it may induce errors in the program's output. Because the duty cycle of a register is defined as the period during which the register is holding useful data expressed as a percentage of the total program execution time, the error rate of each register can be determined by the CEU injection method.

Case Study

CESAR Project

The CESAR project is an Earth observation satellite mission developed in cooperation between two space agencies, INTA of Spain and the National Commission of Space Activities of Argentina. The project, (2002/2003), consists of the design, construction, launching, and operation of a small satellite, weighting approximately 400 kg, with the update of the existing ground segment capabilities in Spain and Argentina to receive and process the CESAR-generated data. The satellite's primary objectives will be cartography, topography, thematic studies, and geophysics, with a satellite payload composed of different cameras (Fig. 3):

1) The panchromatic camera [infrared interferometer spectrometer (IRIS)] is used in the visible range of the spectrum. This camera will be used for cartography and topography.

2) The multispectral camera has six bands in the visible and near infrared range of the spectrum. This camera will be used for natural resources applications.

3) The spectrometer (MEGA) is to measure the concentration of the atmospheric gases involved in the ozone destruction process.

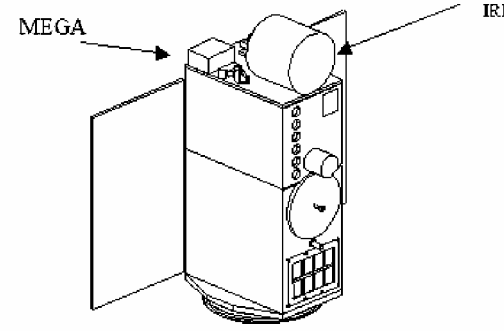

Fig. 3 CESAR satellite current flight configuration.

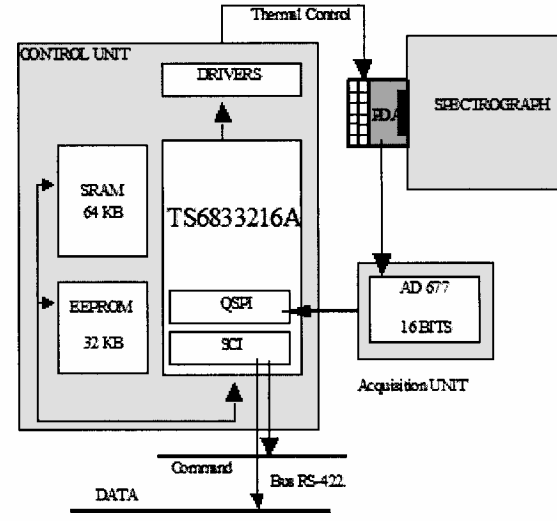

Fig. 4 Architecture of the MEGA instrument.

4) The high-sensitivity panchromatic camera is for the visible range of the spectrum, but has very high sensitivity. This camera will be used to take images of clouds and polar vortex in the nighttime.

INTA is responsible for the design and development of two instruments of CESAR's payload: IRIS and MEGA. Both instruments benefit from the use of state-of-the-art architectures and COTS devices to allow optimum performance. The CESAR program has specified that, for the level of qualification, the electronic components must comply with the MIL-STD-883 B standard and that those complex devices for which there is no SEE information should be tested in a radiation environment. The studied flight software in this paper concerns only the control of the spectrometer. To perform this task, a military version of the TS6833216A microprocessor and an A/D converter have been selected: The TS6833216A (THOMSOM-CSF) is a 32-bit, 16-MHz microprocessor that will serve as control CPU in the MEGA spectrometer. The AD 677 (Analog Devices) is an A/D converter with a 16-bit resolution and 100,000 samples per second with serial interface. This converter is used for reading a linear image sensor (1024 pixels) in the MEGA spectrometer.

Details about the control software are presented in the next section.

Tested Architecture

The TS6833216A microprocessor performs control maintenance, calibration, and test functions of the MEGA instrument as well as downloading data to the satellite storage unit via the MIL-1553 data bus. The microprocessor also controls the exposure time of an array sensor of 1024 photodiodes and reads it by means of the AD677 converter connected to a synchronous channel, the queued serial port interface (QSPI), in the TS6833216A, as shown in the Fig. 4.

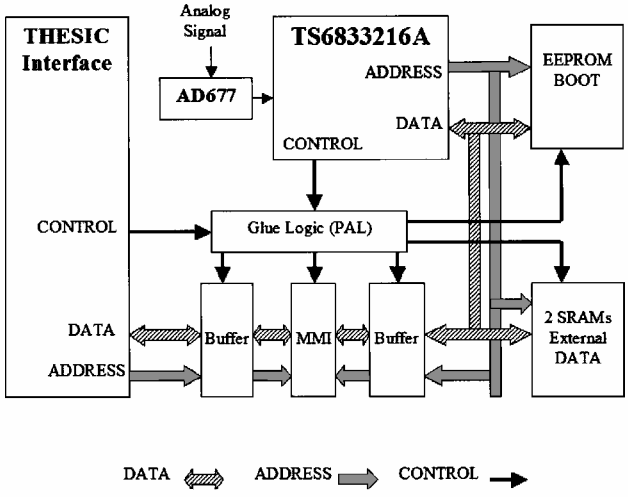

Fig. 5 Block diagram of TS6833216A daughterboard.

A new THESIC daughterboard was designed and developed for the TS6833216A microprocessor for both ground-testing and fault injection purposes.

Experimental Setup

A daughterboard for THESIC testbed is composed mainly of the DUT, glue logic, the MMI for the communications with the motherboard, external SRAM memory, an electrically erasable programmable read only memory (EEPROM) for program storage and clock system. Glue logic adapts the signals of the DUT to the bus of the motherboard. The TS6833216A board also includes an A/D converter (AD677), needed for the adaptation of programs developed for the CESAR instrument. During ground tests, appropriate signal stimuli will exercise the A/D converter, simulating in some way the measurements performed in flight. Figure 5 shows a block diagram of the TS6833216A daughterboard.

Fault injection experiments have been performed on the TS6833216A daughterboard for two benchmark programs (FFT and a matrix multiplication) and the MEGA flight software. These experiments aimed at quantifying the rate of effective upsets for the tested programs, to derive realistic figures for the expected error rate in the final environments.

Experimental Results: CEU Injection

Targets of CEU Injection

Internal memory and registers are considered sensitive areas for the TS6833216A. The registers of this microprocessor are used to control five modules: 1) the system integration module for the system control (watchdog, protection, etc.); 2) the CPU for the code (CPU32); 3) the time processor unit (TPU), which is a dedicated microengine operating independently of CPU32; 4) the queued serial module, which contains a serial communication interface (SCI), and a queued serial peripheral interface; and 5) the control registers for the TPU microcode emulation RAM (TPURAM CTL). Reserved internal memory areas represent 320 B. Notice that reserved zones in this microprocessor (whose content is unknown to the user) and control parts of this processor may cause errors during radiation tests. Those zones, where bit flips can not be injected, could make some difference in the resulting error rate compared to the one obtained by radiation ground tests.

Effects of CEU injection differ according to the occupation of sensitive areas (registers, internal memory, etc.) while executing the considered program, on the target processor. Table 1 summarizes the percentages of occupied sensitive areas, for both internal memory and registers, for each of the two considered benchmark programs, as well as for the flight software running on this microprocessor.

Notice that data needed for the operation of the FFT and the MEGA software are stored in the external memory. When it is considered that these zones were not exposed to heavy ion beams in our experiment, they will not contribute in the calculation of the error rate of a studied program running on this DUT.

Table 1 TS6833216A sensitive area occupation for each tested program

Area	Matrix	FFT	MEGA
Internal memory (2048 B), %	29.3	0	0
Registers (398 B), %	16	21	40
Global sensitive area, %	27.1	3.4	6.5

Table 2 Results of CEU fault injection experiments for the TS6833216A

Group	Matrix	FFT	MEGA
Tolerated errors, %	86.48 ± 0.92	97.8 ± 0.98	99.3 ± 0.99
Result errors, %	12.6 ± 0.35	0.1 ± 0.03	0
Sequence loss, %	0.86 ± 0.09	2.1 ± 0.14	0.7 ± 0.08
Error rate τ_{CEU}, %	13.52 ± 0.36	2.2 ± 0.15	0.7 ± 0.08

Results of CEU Injection

For each program, 40,000 pseudorandom bit flips were injected. Obtained results are classified in three groups: tolerated errors, result errors, and sequence loss errors. Tolerated errors correspond to those bit flips injected on memory elements, which do not cause any effects at the outputs of the program. Result errors include cases where the obtained results and the expected ones differ in at least a single bit. Finally, cases where, after fault injection, we do not get any answer from the processor are classified in the loss of sequence group. The consequences of CEUs belonging to this last malfunction type are unrecoverable, needing a hardware reset to restart program execution. The error rate τ_{CEU} predicted for each program, is calculated according to

$$\tau_{CEU} = \frac{\text{number of errors}}{\text{number of CEUs injected}} \tag{2}$$

Table 2 summarizes obtained results of the CEU injection sessions for the microprocessor TS6833216A.

For the FFT and the MEGA applications, the number of the injected bit flips causing program outputs deviations was negligible. However, for the matrix multiplication application, where all of the matrix operands and results are stored inside the microprocessor internal memory, a significant percentage of the injected CEU provoked errors in the program outputs. These results can be explained by that both the FFT and MEGA programs, in contrast to the matrix multiplication program, do not use the internal memory for the storage of variables or constants. Globally, the low sensitivity of FFT and MEGA programs is certainly a direct consequence of both that the number of registers used is small and that no internal memory area is occupied.

Finally, only a few percentage of the injected CEUs in the matrix multiplication program caused sequence loss errors. This may be because this program is using few control registers. Concerning the detected error types, the sequence loss errors were significantly more frequent for the FFT and MEGA programs than for the matrix multiplication program. The large number of registers used to store critical parameters during the calculation loop can explain this behavior. Indeed, bit flips in these targets will lead to microprocessor crashes exceeding the time limit set by the software watchdog implemented in the motherboard.

Calculation of Duty Cycles

One of the advantages of the CEU injection method is the possibility to perform exhaustive experiments in particular registers, or in critical data stored in internal memory, or even in code area. The case of the program counter was particularly investigated when executing the matrix multiplication program and showed that $14.7 \pm 0.38\%$ of the injected CEU are tolerated, $37.7 \pm 0.61\%$ caused errors in the matrix result, and $47.7 \pm 0.69\%$ provoked system crashes. Hence, the sensitivity of this register is not equal to 100% as generally assumed. We have performed exhaustive CEU injection on the operands of one of the operand matrixes to determine the duty cycles corresponding to this area of the internal memory. Obtained results show that the duty cycle for each position of the internal

memory reserved for the terms of one of the operand matrixes is approximately $11 \pm 0.33\%$.

Experimental Results: Radiation Testing

Radiation testing campaigns, in which the TS6833216A processor was exposed to beams of several heavy-ion species, were performed at the end of October 2000 at the Tandem Van de Graaff particle accelerator (Institute of Nuclear Physics, Orsay, France) and at the end of November 2000 at the cyclotron of Louvain-la-Neuve in Belgium. The latter is a multiparticle, variable energy cyclotron capable of accelerating protons (up to 85 MeV), alpha particles, and heavy ions. The main characteristics of the heavy ions, to which the studied processor was exposed, are given in the Table 3, where Q is the ion charge state and M is the mass in atomic mass units. Further details about this facility are provided in Ref. 11.

These ground-test experiments allowed us to measure the SEE cross sections of the TS6833216A (military version) by executing a memorylike test pattern (static strategy) while exposing the circuit to the heavy ion beams. The obtained SEE cross sections for the studied processor are shown in Fig. 6.

The main contribution of this work is the prediction of the cross section of the studied processor exclusively from the underlying cross section and fault injection experimental results, thus, without running the evaluated program under beam exposure. Obviously, radiation experiments exposing the tested circuit to ion beams are needed to determine the underlying cross section [Eq. (3)], but the hardware and software developments needed to perform such experiments are significantly simpler than the ones corresponding to run the final applications:

$$\sigma_{\mathrm{SEU}} = \frac{\text{number of errors}}{\text{number of particles}} \tag{3}$$

We have run the three mentioned programs while exposing the TS6833216A to the neon heavy ions. Table 4 summarizes the measured error rates for each program.

Except for the matrix multiplication program, these cross sections are negligible and show that, if an application running on the TS6833216A does not use the internal memory, SEUs will be practically without effect. Moreover, except its apparent sensitivity to latchups, this processor is intrinsically robust when running the MEGA flight software and can, therefore, operate under heavy ions in the final environment without any risk of damage by SEUs. Note that part of these results also have been presented in Ref. 12.

We have run the matrix multiplication program under different heavy-ions beams. Obtained sensitivities for the matrix multiplication program are given in Table 5, the corresponding cross sections are shown in Fig. 7.

The good agreement between the predicted and measured error rates proves the efficiency of this new methodology to predict the error rates. This calculation relies on the assumption that all of the locations (registers and internal memory) of the considered processor have the same sensitivity to SEUs. Otherwise, the calculation of duty cycles and cross sections for each location used in the DUT for the studied program is necessary.

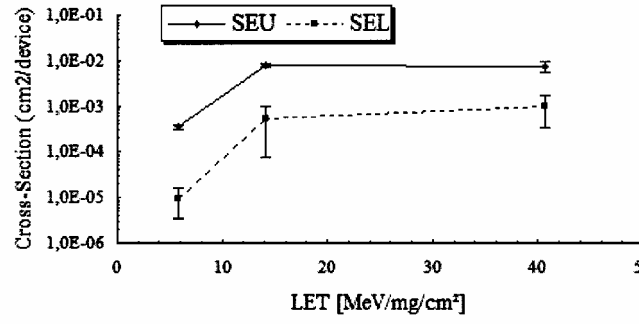

Fig. 6 Sensitivities to SEE of the TS6833216A.

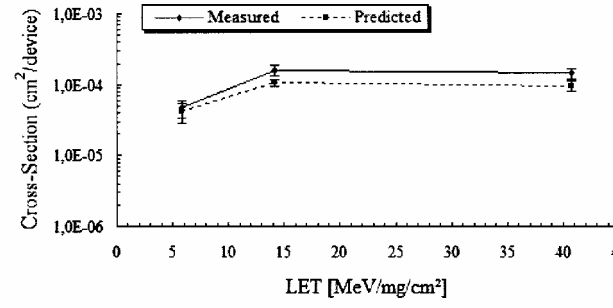

Fig. 7 Predicted and measured error rates of the TS6833216A running a matrix multiplication (10 × 10) program.

Table 3 Different beams used for the TS6833216A

Heavy ion	DUT energy, MeV	Range, μm Si	Linear energy transfer, MeV · mg · cm^2
Br[a]	192	26	40.7
$^{40}Ar^{8+}$[b]	150	42	14.1
$^{20}Ne^{4+}$[b]	78	45	5.85
$^{15}N^{3+}$[b]	62	64	2.97
$^{10}B^{2+}$[b]	41	80	1.7

[a]Tandem Van de Graff, Orsay, France.
[b]Cyclotron, Louvain-la-Neuve, Belgium; $M/Q = 5$.

Table 4 Predicted and measured error rates for the TS6833216A under neon ions

	Matrix		FFT		MEGA	
Cross section	Measured	Predicted	Measured	Predicted	Measured	Predicted
Result errors	$(3.9 \pm 1.2) \times 10^{-5}$	$(3.84 \pm 1.2) \times 10^{-5}$	$<10^{-6}$	$(3.05 \pm 0.9) \times 10^{-7}$	$<10^{-6}$	$<10^{-6}$
Sequence loss	$(0.8 \pm 0.6) \times 10^{-5}$	$(0.26 \pm 0.14) \times 10^{-5}$	$(8 \pm 5.6) \times 10^{-6}$	$(6.4 \pm 1.96) \times 10^{-6}$	$(2 \pm 2.8) \times 10^{-6}$	$(2.1 \pm 4.4) \times 10^{-6}$
Application	$(4.7 \pm 1.3) \times 10^{-5}$	$(4.1 \pm 1.3) \times 10^{-5}$	$(8 \pm 5.6) \times 10^{-6}$	$(6.35 \pm 2.25) \times 10^{-6}$	$(2 \pm 2.8) \times 10^{-6}$	$(2.1 \pm 4.4) \times 10^{-6}$

Table 5 Measured and predicted error rates for the TS6833216A running matrix multiplication (10 × 10)

Heavy ion	LET, MeV · mg · cm^2	Measured application cross section	Predicted application cross section
Neon (Ne)	5.85	$(4.7 \pm 1.3) \times 10^{-5}$	$(4.1 \pm 1.3) \times 10^{-5}$
Argon (Ar)	14.1	$(1.60 \pm 0.25) \times 10^{-4}$	$(1.06 \pm 1.3) \times 10^{-4}$
Bromine (Br)	40.7	$(1.42 \pm 0.23) \times 10^{-4}$	$(1 \pm 1.6) \times 10^{-4}$

Conclusions

In this paper, we have presented a new methodology for prediction of error rates in digital architectures operating under radiation. A flexible tool for bit flip injection (CEU injection), which allowed calculation of duty cycles, was developed concurrently with the execution of a program and implemented on a dedicated tester. Its application to a digital architecture built around the TS6833216A microprocessor was used to study the behavior of this processor in presence of bit flips for various programs. The results of fault injection sessions allowed us to forecast a low sensitivity of the flight software to bit flips provoked by charged particles.

To determine the SEE sensitivities of the TS6833216A microprocessor and to estimate the error rate in-flight for the benchmark programs as well as for a real application, radiation testing campaigns were performed, with two different particle accelerator facilities, in which the microprocessor TS6833216A was exposed to different beams of heavy ions. The good agreement between the predicted and the measured error rates clearly puts in evidence the efficiency of the proposed methodology.

Acknowledgments

This work was supported by Center National d'Etudes Spatiales Grant 721/CNES/2/99/043. The authors would like to thank Guy Berger from the Cyclotron Research Center of Louvain-la-Neuve in Belgium and Thierry Nuns from ONERA–Departement d'Environnement Spatial, Toulouse, France, for their help during radiation testing. The authors would also like to thank the anonymous reviewers for their contribution to the improvement of this paper with their comments.

References

[1]Ma, T., and Dressendorfer, P., *Ionizing Radiation Effects in MOS Devices and Circuits*, Wiley, New York, 1989.

[2]Koga, R., Kolasanski, W. A., Marra, M. T., and Hanna, W. A., "Techniques of Microprocessor Testing and SEU-Rate Prediction," *IEEE Transactions on Nuclear Science*, Vol. 32, No. 6, 1985, pp. 4219–4224.

[3]Elder, J. H., Osborn, J., Kolasinsky, W. A., and Koga, R., "A Method for Characterizing Microprocessor's Vulnerability to SEU," *IEEE Transactions on Nuclear Science*, Vol. 35, No. 6, 1988, pp. 1679–1681.

[4]Bezerra, F., Hardy, D., Velazco, R., and Ziade, H., TILMICRO, "A New SEU Latch-up Tester for Microprocessors Initial Results on 32-bit Floating Point DSPs," *Proceedings of the Third European Conference on Radiation and Its Effects on Components and Systems*, Arcachon, France, 1995, pp. 296–301.

[5]Velazco, R., Karoui, S., Chapuis, T., Benezech, D., and Rosier, L. H., "Heavy Ion Tests for the 68020 Microprocessor and the 68882 Coprocessor," *IEEE Transactions on Nuclear Science*, Vol. 39, No. 3, 1992, pp. 445–449.

[6]Asenek, V., Underwood, C., Velazco, R., Rezgui, S., Oldfield, M., Cheynet, P., and Ecoffet, R., "SEU Induced Errors Observed in Microprocessor Systems," *IEEE Transactions on Nuclear Science*, Vol. 45, No. 6, 1998, pp. 2876–2883.

[7]Carreira, J., Madeira, H., and Silva, J. G., "Xception: A Technique for the Experimental Evaluation of Dependability in Modern Computers," *IEEE Transactions in Software Engineering*, Vol. 24, No. 2, 1988, pp. 125–136.

[8]Benso, A., Civera, P. L., Rabudengo, M., and Sonza Reorda, M., "A Low Cost Programmable Board for Speeding up Fault Injection in Microprocessor-Based Systems," *Annual Reliability and Maintainability Symposium*, Washington, DC, 1999, pp. 171–177.

[9]Velazco, R., Rezgui, S., and Ecoffet, R., "Predicting Error Rate for Microprocessor-Based Digital Architectures Through CEU (Code Emulating Upsets) Injection," *IEEE Transaction of Nuclear Science*, Vol. 47, No. 6, 2000, pp. 2405–2411.

[10]Velazco, R., Cheynet, P., Bofill, A., and Ecoffet, R., "THESIC: A Testbed Suitable for the Qualification of Integrated Circuits Devoted to Operate in Harsh Environment," *IEEE European Test Workshop*, Inst. of Electrical and Electronics Engineers, New York, 1998, pp. 89, 90.

[11]Berger, G., Ryckewaert, G., Harboe-Sorensen, R., and Adams, L., "Cyclone—A Multipurpose Heavy Ion, Proton and Neutron SEE Test Site," *Proceedings of the Fourth European Conference on Radiation and Its Effects on Components and Systems*, Cannes, France, 1997, pp. 296–301.

[12]Rezgui, S., Velazco, R., Ecoffet, R., Rodriguez, S., and Mingo, J. R., "Estimating Error Rates in Processor-Based Architectures," *IEEE Transaction of Nuclear Science*, Vol. 48, No. 5, 2001, pp. 1680–1687.

T. Vladimirova
Guest Associate Editor