

# Statistical Anomalies of Bitflips in SRAMs to Discriminate SBUs from MCUs

Juan Antonio Clemente, Francisco J. Franco, Francesca Villa, Maud Baylac, Solenne Rey, Hortensia Mecha, Juan A. Agapito, Helmut Puchner, Guillaume Hubert and Raoul Velazco

**Abstract**—Recently, the occurrence of multiple events in static tests has been investigated by checking the statistical distribution of the difference between the addresses of the words containing bitflips. That method has been successfully applied to Field Programmable Gate Arrays (FPGAs) and the original authors indicate that it is also valid for SRAMs. This paper presents a modified methodology that is based on checking the XORed addresses with bitflips, rather than on the difference. Irradiation tests on CMOS 130 & 90 nm SRAMs with 14-MeV neutrons have been performed to validate this methodology. Results in high-altitude environments are also presented and cross-checked with theoretical predictions. In addition, this methodology has also been used to detect modifications in the organization of said memories. Theoretical predictions have been validated with actual data provided by the manufacturer.

**Index Terms**—SRAMs, Single event upsets, multiple cell upsets, neutron tests

## I. INTRODUCTION AND RELATED WORK

IT IS well-known that the content of Static Random Access Memories (SRAMs) can be corrupted either due to the impact of energetic particles present in the environment where they operate (cosmic rays, heavy ions, protons, neutrons, ...), or radioactive impurities [1]. Single Event Upsets (SEUs) are a broad category of events by which a single particle strike eventually causes memory cell upsets.

If only one memory cell is flipped as a consequence of such an event, the device experiences a Single Bit Upset (SBU). Otherwise, if the charge generated by the particle is shared by adjacent cells [2], a Multiple Cell Upset (MCU) occurs. Multiple Bit Upsets (MBUs) are a particular kind of MCUs where the flipped bits belong to the same memory word. Memories in old technologies were built placing the

bits of the same word in adjacent cells. Therefore, MBUs were likely to occur. As MBUs, according to their multiplicity, may not be recovered by standard Error Correction Codes (ECCs), manufacturers reacted building the modern generations of memories with bit interleaving, which consists in placing bits from the same memory word physically distant from each other. This technique prevents bits of the same word from being simultaneously perturbed by a single particle impact. Thus, multiple events affect only one bit of several different words (MCU). Unlike MBUs, MCUs are recoverable with standard ECC techniques.

There are several procedures to estimate the SEU cross section of a given device. For years, this parameter was estimated by means of a “static test”. These tests are easy to perform, and they consist in writing the memory with a known pattern, exposing the device to the selected particle beam, and reading the memory only after the irradiation. A second option is to carry out “dynamic tests” such as the March tests [3]. Finally, there are also the so-called “pseudo-static” tests, in which the memories are periodically read during the irradiation, combining dynamic and static modes of operation.

Recent works have demonstrated that the best way to estimate the soft-error cross section is to carry out a dynamic test [4], [5], in order to detect catastrophic phenomena unobservable otherwise. However, static tests are still necessary in several situations, where a dynamic test cannot be performed. For instance, some modern systems reduce the SRAM power supply below its nominal value in order to minimize power consumption. Although the content remains in the memory, it is not accessible until the power supply returns to the nominal value since the peripheral logic blocks are inoperative. Another example is the test of SRAMs boarded in balloons that reach the stratosphere [6].

However, static tests have an inherent problem: when static radiation tests are performed, the researcher obtains a large set of addresses with bitflips sorted in increasing order and it is difficult to group them in SBUs or MCUs. Even if the timestamp of the events is available, when many events occur in a short period of time (i.e., in accelerated tests), only a knowledge of the physical layout would help to decide if two or more events are caused by a unique particle. Unfortunately, this information is usually restricted and therefore, alternative techniques are needed. The authors of [7] propose that the MCUs must not be identified in the address vector, but in a new vector built by combining the addresses in pairs, and subtracting. Moreover, it is quite interesting the introduction of that paper, which contains a well-referenced review of the

This work was supported in part by the Spanish MCINN projects AYA2009-13300-C03 and TIN2013-40968-P, by the mobility grant “José Castillejo” for professors and researchers, and by UCM-BSCH.

J. A. Clemente and H. Mecha are with the Computer Architecture Department, Facultad de Informática, Universidad Complutense de Madrid (UCM), Spain, e-mail: ja.clemente@fdi.ucm.es, horten@dacya.ucm.es.

F. J. Franco and J. A. Agapito are with the Departamento de Física Aplicada III, Facultad de Físicas, Universidad Complutense de Madrid (UCM), Spain, e-mail: fjfranco, agapito@fis.ucm.es.

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

H. Puchner is with Cypress Semiconductor, Technology R&D, 3901 San Jose, CA, USA. e-mail: hrp@cypress.com

G. Hubert is with the French Aerospace Laboratory (ONERA), Toulouse, France, e-mail: guillaume.hubert@onera.fr.

R. Velazco is with the Université Grenoble-Alpes & CNRS, TIMA, Grenoble (France), e-mail: raoul.velazco@imag.fr.

state-of-the-art techniques to discriminates SBUs from MBUs. Once the Difference-of-Addresses Vector ( $DAV$ ) is plotted as a histogram, some values appear many more times than the bulk of possible values. Thus, they are attributed to MCUs and the addresses originating these anomalies are identified. This procedure has successfully been used to study FPGAs and proposed, but not verified, for SRAMs.

MCUs have also been studied and classified depending on their multiplicity and regularity. The study in [8] establishes four categories: *Type-A*, *Type-B*, *Type-C* and *Type-D*, based on an analysis of the spatial and temporal distances among events. Thus, *Type-A* MCUs (the most common ones) are isolated clusters of at most a few tens of bitflips. *Type-B* MCUs consist in blocks containing from a few tens to several hundreds of errors in addresses physically close to each other. *Type-C* ones strongly depend on the access pattern since they are caused by temporary failures of the memory's I/O or synchronization circuitry. They comprise up to tens of thousands of bitflips. Finally, *Type-D* MCUs affect up to hundreds of thousands of addresses located at the edges of the memory.

In this paper, we propose a modification of the procedure presented in [7]. The main difference is that it processes the values that appear too frequently after XORing (instead of subtracting) the addresses with bitflips, as it was sketched in previous works [9]. The main advantage being that it is possible to accurately predict the expected frequency of values in a system where only SBUs occur and to compare the predictions with the actual data. In other words, the existence of a theoretical model provides a well-founded reason to find the values that, probably, link addresses involved in one multiple event. We have validated this approach with experimental data issued from experiments on commercial Cypress SRAMs with 14-MeV neutrons, in the GENEPI2 neutron source (*GENérateur à NEutrons Pulsés Intenses*)[9], [10]. The different cross-sections issued from these experiments were used to predict the theoretical soft error rate of the memories, which was compared with experimental values obtained in high altitude environments. The presented procedure is valid for identifying the so-called *Type-A* MCUs described in [8] (i.e., isolated clusters of at most a few tens of bitflips), since the other types of MCUs can only appear when carrying out dynamic tests. A preliminary version of this method was presented in [11].

The remainder of the paper is organized as follows: Section II presents the theoretical background used by the proposed MCU detection algorithm. Section III discusses the experimental setup and Section IV describes the proposed approach and provides experimental results, which are discussed in Section V. Section VI concludes the paper.

## II. STATISTICAL PROPERTIES OF THE DAV

### A. Equiprobability After XORing

Let  $U_N$  be the space of addresses of a SRAM ( $N$  being the length of the address word,  $N \in \mathbb{N}$ ); and  $V_{N,q}$ , the set of  $q$  addresses where bitflips have been detected.  $U_N$  can be represented as the set of natural numbers ranging between 0 and  $L_N = 2^N - 1$ , which can be codified in binary format as

Table I  
XORING VS. SUBTRACTING 2-BIT ADDRESSES

		$Y \oplus X$				$Y - X$			
		00	01	10	11	00	01	10	11
X \ Y	00	–	01	10	11	–	01	10	11
	01	–	–	11	10	–	–	01	10
	10	–	–	–	01	–	–	–	01
	11	–	–	–	–	–	–	–	–

words of length  $N$ . The subset  $V_{N,q} = \{v_1, v_2, \dots, v_q\} \subset U_N$ , is built by taking  $q$  addresses from  $U_N$ , without repetition<sup>1</sup>, and arranging them in increasing order. Now, let us define the XORed Difference-of-Addresses Vector ( $XDAV$ ) as the set of addresses of  $U_N$  obtained from  $V_{N,q}$  as:

$$XDAV = \{x = v_j \oplus v_i \mid 1 \leq i < q, i < j \leq q\} \quad (1)$$

Besides, the Classical Difference-of-Addresses Vector ( $CDAV$ ) is defined in the same way as:

$$CDAV = \{x = v_j - v_i \mid 1 \leq i < q, i < j \leq q\} \quad (2)$$

Both sets share the following properties. First, as  $\forall i, j, i < j \Rightarrow 0 \leq v_i < v_j \leq L_N$ , every element in the  $XDAV$  or the  $CDAV$  are higher than 0 and lower or equal to  $L_N$ . Besides, it is easy to demonstrate that the number of elements in both sets is:

$$N_{DAV} = 0.5 \cdot q \cdot (q - 1) \quad (3)$$

The  $CDAV$  was successfully used in [7] to detect multiple events. However, the  $XDAV$  has an important property, absent in the  $CDAV$ , which is the conservation of the probability in special circumstances.

The random selection of an address  $v_k = (b_{N-1}b_{N-2} \dots b_1b_0)$  from  $U_N$ ,  $b_k$  being one of the  $N$  bits, is equivalent to choosing in  $N$  steps the values of  $b_k$  out of  $\{0, 1\}$  with a probability of  $\frac{1}{2}$ . When this address is XORed bit to bit with another one,  $v_j = (c_{N-1}c_{N-2} \dots c_1c_0)$ , created in the same way, there are four possible results for each bit:

$$\begin{aligned} b_k = 0, \quad c_k = 0 &\rightarrow b_k \oplus c_k = 0 \\ b_k = 0, \quad c_k = 1 &\rightarrow b_k \oplus c_k = 1 \\ b_k = 1, \quad c_k = 0 &\rightarrow b_k \oplus c_k = 1 \\ b_k = 1, \quad c_k = 1 &\rightarrow b_k \oplus c_k = 0 \end{aligned}$$

This means that the probability of obtaining 0 or 1 in any bit of the result is 50%, independently of the values of the addresses that were XORed. This property brings an important consequence: As the values of the bits in  $v_k \oplus v_j$  are equiprobable, its creation is formally equivalent to randomly taking  $v_k \oplus v_j$  from  $\{1, 2, \dots, L_N\}$  with identical probability. Provided that there are  $L_N$  elements, this probability is:

$$p_X = (L_N)^{-1} \quad (4)$$

<sup>1</sup>Actually, this condition is not strictly necessary but it simplifies the mathematical calculations without losing accuracy.

For illustrating purposes, columns 2-5 in Table I show all the combinations in a system with 2-bit addresses ( $N = 2$ ), assuming that  $X \neq Y$  and  $X < Y$ , being  $X, Y \in \{1, L_N\}$  (this is consistent with the operation of the proposed approach, explained further in detail). In this case, any element in the table between 1 and  $L_2 = 2^2 - 1 = 3$  appears exactly 2 times out of 6.

On the contrary, the last four columns in Table I demonstrate that, in the case of subtracting the pairs of addresses, result 01 is overrepresented (3 times out of 6). In fact, it can be demonstrated that the probability of  $x \in CDAV$  being  $k$  is:

$$p_C(k) = \frac{2}{L_N \cdot (L_N + 1)} \cdot [L_N + 1 - k] \quad (5)$$

with  $1 \leq k \leq L_N$ . The absence of symmetry in this distribution makes its study very difficult and inaccessible unlike the much simpler Eq. 4. This is the reason why the approach presented in this paper uses the *XOR* operation on the addresses with bitflips, unlike the previous version of this method [7], which used the subtraction. This allows building a theoretical model and a systematic methodology to accurately extract MCUs from SEUs in large sets of data.

### B. Statistical Properties of Only-SBU Scenarios

Eq. 4 allows making predictions for the statistical properties of the *XDAV*. Unfortunately, the lack of symmetry of Eq. 5 makes deductions very technical, so predictions about *CDAV* will not be done in this paper due to the space constraints.

Typically, SBUs appear in randomly distributed addresses of the tested memory, not related to each other [12]. Therefore, the set of addresses is formally equivalent to the subset  $V_{N,q}$ , described in the previous subsection. Thus, in only-SBU scenarios, the elements of the *XDAV* can be supposed to be randomly and uniformly chosen from  $\{1, 2, \dots, L_N\}$ . The following question arises: Which is the probability of a value  $m \in \{1, 2, \dots, L_N\}$  appearing  $k$  times in the *XDAV*? This is a classical and well-known problem in the probability field, similar to that of winning the lottery  $k$  times in  $N_{DAV}$  trials with a probability of  $p_X$  in each round. Thus, the probability of an element appearing  $k$ -times in the *XDAV* is:

$$P_N^{XDAV}(k) = \binom{N_{DAV}}{k} \cdot p_X^k \cdot (1 - p_X)^{N_{DAV} - k} \quad (6)$$

As there are  $L_N$  possible candidates, the predicted number of elements appearing  $k$ -times in the *XDAV* is:

$$N^{XDAV}(k) = L_N \cdot P_N^{XDAV}(k) \quad (7)$$

Another interesting property of the *XDAV* in only-SBU scenarios is that the number of elements that have  $k$  ones in binary format is:

$$N_1^{XDAV}(k) = \frac{1}{L_N} \cdot \binom{N}{k} \cdot N_{DAV} \quad (8)$$

$N$  being the number of address bits. This fact is demonstrated since it is formally equivalent to the classic problem of obtaining  $k$  heads after tossing a coin  $N$  times, and repeating the experiment  $N_{DAV}$  times.

### C. Search of anomalous *XDAV* elements

In order to search MCUs from a pool of events, the first step is to decide if the number of multiple events is negligible compared to the number of SBUs. Thus, both the *XDAV* and the number of times that every element appears in said *XDAV* must be obtained from experimental data. If the results are in agreement with the predictions issued from Eq. 7, then the number of MCUs is negligible.

On the contrary, if the histogram does not follow Eq. 7, the experiment is not consistent with the only-SBU model. In this paper, it is postulated that *XDAV* elements repeated too many times (with respect to the predictions issued from Eq. 7) are the signature of multiple events. This step is equivalent to the seek of anomalous frequencies in the histogram depicted by Wirthlin *et al.* in [7].

Another interesting property can help to identify these values that link related addresses. Typically, large SRAMs consist in a number of replicated modules controlled by demultiplexers. Thus, SRAMs are divided into  $2^{N_Q}$  quads or "banks", every quad is divided into  $2^{N_B}$  blocks and, finally, in every block there is a row decoder, demultiplexed by  $N_R$  bits; plus a column decoder, demultiplexed by  $N_C$  bits. Therefore,  $2^{N_Q + N_B + N_R + N_C} = 2^N$  words can be addressed in the memory. Blocks and quads are physically separated, so it is extremely unlikely that 14-MeV neutrons, not very energetic at all, can induce events in cells belonging to different blocks or quads (this point has been confirmed with the manufacturer). Thus, if a 2-bit multiple event occurs in cells in the same row, they will share at least  $N_Q + N_B + N_R$  bits. If both addresses were combined to generate an *XDAV* element, the result would contain at least  $N_Q + N_B + N_R$  zeroes, if expressed in binary format. On the contrary, if it occurs in the same column, the number of shared address bits is  $N_Q + N_B + N_C$ . This leads to an interesting conclusion: In binary format, *XDAV* values relating addresses involved in MCUs will probably contain a large quantity of zeroes. It will be shown later that this fact is useful to accept or reject candidates for the set of critical *XDAV* values.

## III. EXPERIMENTAL SETUP

Two commercial 2M×8-bit CMOS SRAMs, the CY62167DV and CY62167EV, from Cypress Semiconductor and in 130 & 90 nm technologies respectively, were irradiated in the 14-MeV neutron source GENEPI2, available at the LPSC (*Laboratoire de Physique Subatomique et de Cosmologie*), in Grenoble (France) [9], [10]. SRAMs were set at a fairly large distance from the target (40 cm), to limit the neutron flux to approximately  $3 \times 10^4 \text{ n}\cdot\text{cm}^{-2}\cdot\text{s}^{-1}$ . Under these conditions, the memories were exposed to a fluence of  $0.7\text{-}1.1 \times 10^8 \text{ n}/\text{cm}^2$  within 1 hour. It must be taken into account that, after these tests, the facility was upgraded obtaining much higher neutron flux values.

The memories were irradiated at their nominal power supply (3.3 V) with different patterns ( $0 \times 00$ ,  $0 \times FF$ ,  $0 \times 55$ ) and incidence angle ( $0^\circ$ ,  $45^\circ$ ) in rounds of about 1 hour. As the memories' contents were checked every 45 seconds, tests fall in the "pseudo-static" category. More than 100 errors were

**Algorithm 1** The proposed algorithm to extract MCUs

**Input:**  $addr\_errors$ , addresses where bitflips were observed, sorted increasingly.

**Output:**  $MCUs$ , set of MCUs.

*#Step 1: Extraction of critical values by XORing*

- 1:  $XDAV := calculate\_XDAV(addr\_errors)$ ;
- 2:  $H := calculate\_histogram(XDAV)$ ;
- 3:  $k_0 := calculate\_boundary\_repetitions\_MCUs(H, 0.05)$ ;
- 4:  $cr\_values := select\_most\_repeated\_values(H, k_0, 15)$ ;

*#Step 2: Refinement of the critical values search*

- 5:  $trace := calculate\_trace(XDAV)$ ;
- 6:  $refinement\_trace(&cr\_values, trace)$ ;
- 7:  $refinement\_pattern(&cr\_values)$ ;
- 8:  $refinement\_XOR(&cr\_values)$ ;

*#Step 3: Obtention of the final results*

- 9:  $addr\_MCUs := select\_addresses(addr\_errors, cr\_values)$ ;
- 10:  $MCUs := group\_addresses(addr\_MCUs)$ ;

observed in each experiment, but never more than 7 errors were detected in each reading round. An average number of 95 rounds were carried out per experiment, which were performed at room temperature ( $\sim 20^\circ\text{C}$ ). The addresses of each experiment were mixed and sorted increasingly as if they had been obtained in a static test (thereby intentionally losing the timestamp of the events). Two reasons motivated this decision: 1) To have a more complex case study in order to check if the algorithm does not merge in the same MCU addresses affected in different rounds of reading, and 2) to imitate a scenario that would occur in case of carrying out tests where a periodic read is not possible.

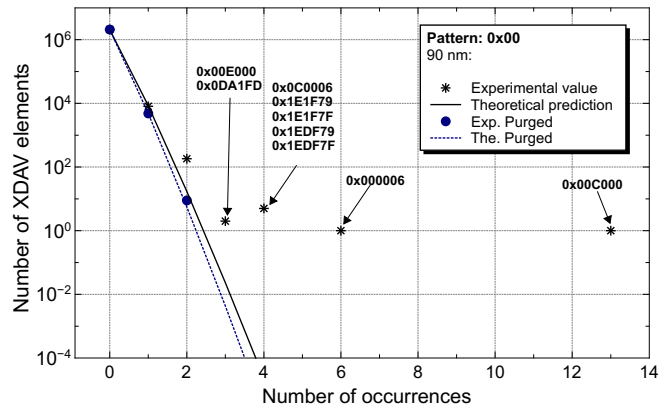
#### IV. THE PROPOSED APPROACH AND RESULTS

The pseudo-code in Algorithm 1 describes the proposed methodology to detect MCUs. It is divided in three steps, which are described in detail in the following subsections.

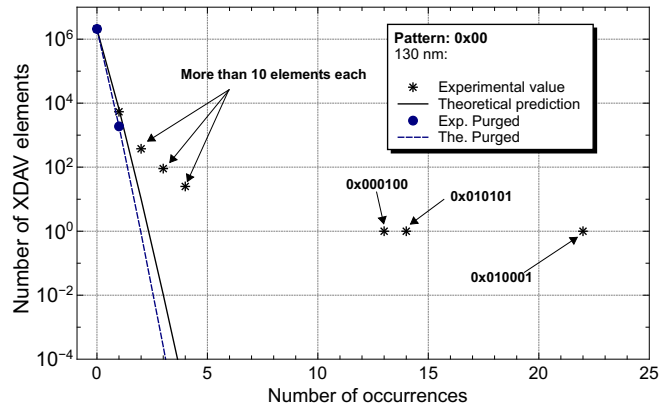
##### A. Step 1: Extraction of Main Critical Values by XORing

First of all, the algorithm obtains the  $XDAV$  from the set of addresses with errors,  $addr\_errors$  (Line 1). Then, the histogram  $H$  of the number of repetitions in  $XDAV$  is obtained (Line 2). Once  $N_{DAV}$  is calculated, the system uses Eq. 7 to find out a critical  $k_0 \in \mathbb{N}$  such that:  $N^{XDAV}(k_0) < 0.05 < N^{XDAV}(k_0 - 1)$  (Line 3).  $k_0$  is the boundary that separates the number of repetitions that can be attributed to randomness from those that cannot be explained in an only-SBU scenario (Line 3). 0.05 is an arbitrary value that we have used in our algorithm based on the well-known “95%-confidence”.

The following step is to check the histogram to find those  $XDAV$  elements that appear  $k_0$  or more times (Line 4). As randomness is excluded, they are somehow related to the occurrence of MCUs. However, it is not advisable to keep all of these elements. Experimentally, we have verified that it is better to select no more than 15-20  $XDAV$  values to avoid the selection of false positives (i.e., addresses falsely attributed to MCUs). Therefore, the algorithm selects the



(a) 90-nm memory



(b) 130-nm memory

Figure 1. Number of elements of  $XDAV$  (Y-axis) that were found the number of times specified in the X-axis. Experimental values vs. theoretical predictions. For a given number of repetitions  $n$ , the absence of star or dot indicates that no element was found  $n$  times in the  $XDAV$

candidates in decreasing order of occurrences and it adds them in  $cr\_values$ . When this set contains 15 or more values, this selection process stops, even if there are still elements that appear more times than expected.

Fig. 1a shows the data for the 90-nm SRAM with the  $0 \times 00$  pattern. In this case, 131 addresses contained errors, each one affecting only one bit. Therefore,  $N_{DAV} = 8515$ . First of all, let us pay attention to the raw data obtained in the experiments (stars), in comparison with the theoretical value (straight line)<sup>2</sup>. Some values strongly deviate from the predictions deduced from Eq. 7. For instance, the value  $0 \times 00C000$  appears 13 times in the  $XDAV$ , but according to the prediction, no elements should appear such number of times (see the straight line in the figure). In fact, the probability of a value appearing 13 times in the  $XDAV$  is  $2.7 \times 10^{-35}$ . A similar deviation applies to  $0 \times 000006$ , which appears 6 times, and there are 5 and 2 elements that appear 4 and 3 times, respectively. Thus, these data are not compatible with an only-SBUs scenario and MCUs can be extracted from the bitflip pool. The remainder of the experimental data match the theoretical predictions.

For the 130-nm SRAM with the  $0 \times 00$  pattern (Fig. 1b, with

<sup>2</sup>The data labeled as *Purged* will be discussed further in this section

115 bitflips), there are 29 elements that appeared 4 times or more. Even if they do not match Eq. (7), and they are clearly over-represented, the algorithm does not select any of them. Thus, the algorithm starts selecting elements that appear most frequently in the  $XDAV$  and, given a set of values that are repeated  $n$  times, it only adds them to the  $cr\_values$  set (with size  $m$ ) if  $n + m \leq 15$ . This value was selected after a “trial and error” approach. In case this method was used for other particles, energies and/or devices, a different value rather than 15 could be used, but it is important to keep it low enough in order to avoid the aggregation of false positives to the selected  $XDAV$  values. We believe that, even if the new value is very conservative, this method will work very well if the refinement described in Section IV-B is applied.

Back to the 130-nm memory, the selected values are  $0 \times 010001$ ,  $0 \times 010101$ , and  $0 \times 000100$ , which appear in the  $XDAV$  22, 14, and 13 times respectively. If the 26 values that appeared 4 times were also selected, the size of the  $cr\_values$  set would be 29, which is considerably greater than 15 (note that there were no values that appeared from 5 to 12 times). It is also worth noting that these three elements are related, since  $0 \times 010001 \oplus 0 \times 000100 = 0 \times 010101$ . As we will see, this property seems to appear more times.

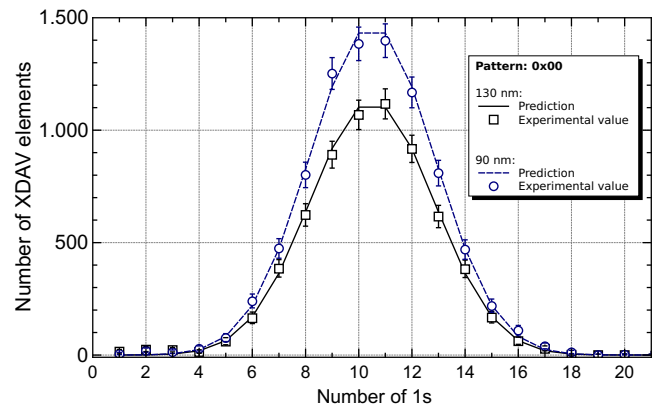
### B. Step 2: Rules to refine the search

If only Step 1 is applied, some of the actual critical  $XDAV$  values may not be identified, and/or false positives might be selected as well. The following rules allow refining the search:

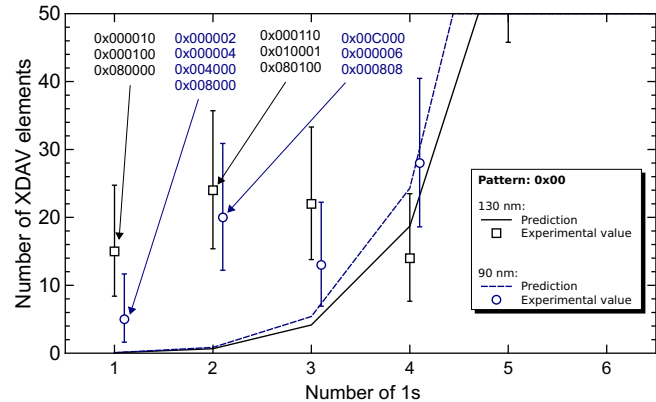
1) *Trace Rule*: Let the *trace* of an element  $e \in XDAV$  be the number of 1's existing in it when expressed in binary format. Provided that SRAMs are modular (as detailed in Section II-C), elements overrepresented in the  $XDAV$  with very low trace (1 or 2) are candidates for the set of critical  $XDAV$  values, even if they were discarded in Step 1. On the contrary,  $XDAV$  elements with a medium to high trace are discarded even if they were selected in the previous step (Lines 5 and 6 in Algorithm 1).

2) *Pattern Rule*: The written pattern can affect the relative position of the cells involved in multiple events. Therefore, it is highly advisable to perform static tests with different patterns (all-zeroes, all-ones, checkerboards, etc). This rule determines that, if a test is repeated in identical conditions and with different patterns, only the true critical values will appear simultaneously in the histograms obtained from both  $XDAV$ s. Thus, if this information is available, our algorithm applies this rule and updates the  $cr\_set$  accordingly (Line 7). As a consequence of applying this rule, some new critical values may be included and/or others may be rejected.

3) *XOR candidates rule*: The third rule consists in accepting dubious  $XDAV$  elements: in particular, those appearing more times than  $k_0$ , but which were not selected in Step 1, since the number of selected values would be  $> 15$  (Line 8 in Algorithm 1). For this purpose, the XOR operation is used again. For example, let us assume that  $M_1$  and  $M_2$  are two confirmed critical  $XDAV$  values and it is suspected that another value  $M_3$  could also be critical as well. If  $M_3 = M_2 \oplus M_1$  or  $M_2 = M_1 \oplus M_3$ ,  $M_3$  can be confirmed.



(a)



(b)

Figure 2. (a) Number of elements in the  $XDAV$  with  $k$  ones inside ( $X$ -axis). Error bars were calculated with the inverse  $\chi^2$  function, as explained in [13]. An equivalent graph, but related to the  $CDAV$ , can be found in [14]. (b) Zoomed left tail of the distribution in (a)

$M_1$  could link addresses in the same row ( $X$ -axis) and  $M_2$  does likewise in the same column ( $Y$ -axis); or viceversa. As address bits are not shared by the column and row decoders simultaneously,  $M_3 = M_2 \oplus M_1$  is a value linking cells along the line intersecting the  $X$ -axis at  $45^\circ$  or  $135^\circ$ .

Now, let us apply the algorithm to experimental results. Fig. 2a compares the occurrence ( $Y$ -axis) of  $XDAV$  elements with  $k$  ones ( $X$ -axis) in the SRAMs with the  $0 \times 00$  pattern. One can see that Eq. 8 predicts the experimental results since the number of SBUs is much higher than MCUs. However, in the left side of the distribution (zoomed in Fig. 2b), disagreements appear. The reason of this discrepancy is the existence of MCUs. Some of the values with trace 1 or 2 had already been discovered in Step 1 of the Algorithm, but others were not.

Let us focus on the 130-nm memory. In this case, there is an exceptionally frequent value,  $0 \times 000100$ , appearing 13 times. The other two possible candidates with trace 1,  $0 \times 000010$  and  $0 \times 080000$ , only appear once so they can be just the result of randomness. However, further inspecting the elements with trace 2, one can observe that, apart from  $0 \times 010001$ , there are two elements ( $0 \times 000110$  and  $0 \times 080100$ ) that only appear once and that can be derived by XORing  $0 \times 000100$ , the recently accepted critical value that was found

13 times, with  $0 \times 000010$  and  $0 \times 080000$ , respectively. In conclusion, hints to consider these values as MCU signatures are really strong. This is supported by two facts: Firstly, pairs of addresses yielding the  $XDAV$  values appeared in the same round, and secondly, they were also identified as critical values when other patterns were used.

In the 90-nm memory, the elements with trace 1 appear only once except  $0 \times 008000$ , which appears twice. However, there is a value ( $0 \times 000006$ ) with trace 2, which appears more times than expected. Since  $0 \times 000006 = 0 \times 000002 \oplus 0 \times 000004$ , we believe that these are also critical  $XDAV$  values. In the case of  $0 \times 004000$  and  $0 \times 008000$ , XORing both elements yields  $0 \times 00C000$ , one of the critical values. Therefore, both of them are also added to  $cr\_values$  (remember Line 8 in Algorithm 1). As in the 130-nm memory, this fact is backed up by three facts: 1) Addresses originating these values appear in the same round, 2) These values are recurring when other patterns are used, and 3)  $0 \times 000002$  is one of the evident critical  $XDAV$  values with a  $0 \times FF$  pattern. On the contrary, nothing indicates that  $0 \times 000808$  is a critical value. Neither does it appear too many times nor can it be derived by XORing other values, and it does not appear when other patterns are used. In fact, it involved addresses taken in different rounds, so it is a random value.

One final question to answer is why some other strange values appear so often in the  $XDAV$ . For example, in Fig. 1a, one can see that elements such as  $0 \times 1E1F70$ ,  $0 \times 1E1F7F$  ..., appear up to 4 times. After analyzing the results, it was discovered that they were the result of an anomalous boost, due to the interaction between two large multiple events. Thus, let us suppose that a 4-bit event affects addresses  $A_0$ ,  $A_0 + k_1$ ,  $A_0 + k_2$ , and  $A_0 + k_3$ . Later, another similar multiple event affects  $B_0$ ,  $B_0 + k_1$ ,  $B_0 + k_2$ , and  $B_0 + k_3$ . This pair of MCUs yields 16 elements of the  $XDAV$  with very close values around  $A_0 \oplus B_0$  that, apparently, indicate the existence of several critical addresses. Fortunately, these false  $XDAV$  values are not difficult to identify since they do not fit with the rest of critical values, which usually consist in elements whose trace is low. Moreover, in practice, the only consequence is that both events are merged in one anomalous and nonsense event of 8 bits, which is easy to discover.

In fact, after selecting the addresses involved in the  $cr\_values$  set and removing them from  $XDAV$ , a new histogram with the remaining values (*purged* data) is in concordance with the theoretical prediction for only-SBUs scenarios (dashed lines and black dots in Figs. 1a and 1b). This is very interesting because it demonstrates that the algorithm did the right thing at not selecting these anomalous elements in the  $XDAV$ , which were overly numerous.

### C. Step 3: Obtention of the Final Results

Finally, the addresses involved in the critical values of  $cr\_values$  are selected (Line 9 of Algorithm 1), and they are finally grouped in MCUs (Line 10). If the same address appears in two different values from  $cr\_values$ , the involved addresses are merged in an MCU of multiplicity 3. This is done as many times as necessary, until the same address does

Table II  
CRITICAL VALUES IN THE EXPERIMENTAL XDAVS

90 nm			130 nm		
Pattern	Values	Pairs	Pattern	Values	Pairs
$0 \times 00$ $N = 131$	$0 \times 000002$	1	$0 \times 00$ $N = 115$	$0 \times 000010$	1
	$0 \times 000004$	1		$0 \times 000100$	13
	$0 \times 000006$	6		$0 \times 000110$	1
	$0 \times 004000$	1		$0 \times 010001$	22
	$0 \times 008000$	2		$0 \times 010101$	14
	$0 \times 00C000$	13		$0 \times 080000$	1
	$0 \times 00E000$	3		$0 \times 080100$	1
$0 \times 55$ $N = 120$	$0 \times 000002$	2	$0 \times 55$ $N = 146$	$0 \times 000080$	2
	$0 \times 000006$	3		$0 \times 000100$	1
	$0 \times 000100$	1		$0 \times 010000$	1
	$0 \times 004000$	1		$0 \times 010001$	19
	$0 \times 00C000$	7		$0 \times 080100$	8
	$0 \times 00C002$	3		$0 \times 090101$	7
	$0 \times 00E000$	5			
$0 \times 040004$	1				
$0 \times FF$ $N = 108$	$0 \times 000002$	6	$0 \times FF$ $N = 129$	$0 \times 010001$	20
	$0 \times 008002$	1		$0 \times 080100$	6
	$0 \times 00C000$	4		$0 \times 090101$	7
	$0 \times 00C002$	3		$0 \times 0C0100$	6
	$0 \times 00E000$	3		$0 \times 0D0001$	6

Table III  
EVENTS IN THE TESTS WITH NORMAL INCIDENCE

90 nm						
Pattern	1-bit	2-bit	3-bit	4-bit	5-bit	6-bit
$0 \times 00$	92	12	1	3	0	0
$0 \times 55$	86	12	2	1	0	0
$0 \times FF$	80	11	2	0	0	0

130 nm						
Pattern	1-bit	2-bit	3-bit	4-bit	5-bit	6-bit
$0 \times 00$	62	10	5	2	2	0
$0 \times 55$	100	13	2	2	0	1
$0 \times FF$	81	13	2	4	0	0

not appear in more than one event. For the data presented in Figs. 1a and 1b, the resulting MCUs were double-checked and it was verified that the addresses grouped in the same event were not detected in two different rounds of reading.

Table II shows the critical values, anomalously overrepresented in the  $XDAV$ , attributed to the occurrence of MCUs. The table breaks down the results for the six experiments that were carried out: three different patterns:  $0 \times 00$ ,  $0 \times 55$  and  $0 \times FF$ , for each one of the two memories studied. In columns 1 and 4,  $N$  is the number of errors that were observed in the experiment. Columns 2 and 5 show the different critical values that were identified by our algorithm, whereas columns 3 and 6 indicate the number times that each value was found in  $XDAV$  (or similarly, the number of pairs of addresses involved in the calculation of those  $XDAV$  values). Table III classifies the events, according to their multiplicity. The estimated values for the different SEU cross sections are shown in Table IV, calculated with a 95-% confidence as explained in [13].

Table IV  
SEU CROSS SECTIONS WITH NORMAL INCIDENCE

90 nm			
	$0 \times 00$	$0 \times 55$	$0 \times FF$
1-bit	18 – 28	18 – 27	16 – 24
2-bit	1.5 – 5.2	1.6 – 5.4	1.4 – 4.8
3-bit	0.01 – 1.4	0.06 – 1.9	0.06 – 1.8
4-bit	0.15 – 2.2	< 0.95	< 0.91
5-bit	< 0.92		
$\times 10^{-16} \text{ cm}^2/\text{bit}$			

130 nm			
	$0 \times 00$	$0 \times 55$	$0 \times FF$
1-bit	8.4 – 14	14 – 21	11 – 18
2-bit	0.9 – 3.3	1.2 – 3.9	1.2 – 3.9
3-bit	0.3 – 2.1	0.04 – 1.3	0.04 – 1.3
4-bit	0.04 – 1.3		0.19 – 1.8
5-bit		< 0.64	< 0.64
6-bit	< 0.65		
7-bit	< 0.64		
$\times 10^{-16} \text{ cm}^2/\text{bit}$			

Table V  
EVENTS AT THE PIC-DU-MIDI FACILITY

	Round 1	Round 2	Round 3	Round 4
1-bit	85	55	48	100
2-bit	19	17	5	18
3-bit	3	10	4	6
4-bit	3	2	0	3
5-bit	3	0	0	0
6-bit	0	0	1	0
7-bit	1	0	0	0
Total bitflips	166	127	76	166
Time	26 months (Rounds 1-4)			

#### D. Real-Life Tests

A set of 64 90-nm CY62167EV SRAMs (1 Gbit in total) was used in real-life tests at the Pic-du-Midi scientific station, located at 2885 m above sea level using a  $0 \times 5555$  pattern. Unlike the tests in the GENIEPI2 neutron accelerator, the memories were configured as  $1\text{M} \times 16$  bits. Data acquisitions in the Pic-du-Midi were started in May 2011 and they allowed for obtaining a significant amount of data ( $> 200$  SEUs). Furthermore, a neutron spectrometer has been installed by ONERA at the Pic-du-Midi [15], allowing characterizing the local neutron environment.

Bitflips were registered and it was possible to classify them as SBUs (1-bit events) or MCUs (from 2-bit to 7-bit events) using the same strategy. Table V summarizes the results obtained in different rounds, explaining the number of events according to their multiplicity. In general, the critical values are identical to those in Table II including several new ones:  $0 \times 00005$ ,  $0 \times 00007$ ,  $0 \times 08000$ , or  $0 \times 08002$ . The appearance of these new values can be explained by a change in the memory organization.

In order to link the radiation field measurement and the observed SEUs, a modeling approach named MUSCA-SEP3 was used in this work [16]. Calculations consider a dynamic

Table VI  
PREDICTED AND MEASURED SER, SEPARATING SINGLE AND MULTIPLE EVENTS AND SPECIFYING THE EVENT MULTIPLICITY. THE CONFIDENCE LEVEL FOR THE INTERVALS OF COLUMN 4 IS 95 %

Event multiplicity	Predicted SER ( $SEU \cdot Gbit^{-1} \cdot h^{-1}$ )	Predicted SER ( $event \cdot Gbit^{-1} \cdot h^{-1}$ )	Measured SER ( $SEU \cdot Gbit^{-1} \cdot h^{-1}$ )
1-bit	328	328	314 – 397
2-bit	113	56.6	96 – 163
3-bit	72.4	24.2	56 – 133
4-bit	50.7	15.8	27 – 124
5-bit	38.5	7.70	8 – 111
6-bit	19.0	4.53	1 – 70
7-bit	6.34	0.906	1 – 35
Total	628	438	628 – 747
	$\times 10^{-5}$	$\times 10^{-5}$	$\times 10^{-5}$

neutron spectrum issued from the neutron spectrometer, a technological model (i.e. elementary cell topology) determined thanks to a reverse engineering and an occurrence model derived from neutron and proton ground tests.

Columns 2 and 3 in Table VI present the predicted soft-error rate (SER, expressed in  $SEU \cdot Gbit^{-1} \cdot hr^{-1}$  and  $event \cdot Gbit^{-1} \cdot hr^{-1}$ , respectively) separating single and multiple events. Column 4 shows the actual SER that was deduced from the data presented in Table V. The orders of magnitude of the predictions and the measurements are consistent.

## V. DISCUSSION

### A. Some Lessons Learned from the Experiments

The methodology proposed in this paper has demonstrated to be quite successful and computationally efficient. It is clear that the MCUs shown in this paper could have been discovered by careful visual inspection. However, in some situations that is completely unfeasible. For example, in later tests with 14-MeV neutrons, the authors have registered more than 1500 bitflips in only one 5-minutes round. Another advantage is the possibility of recycling the results from experiment to experiment. Let us suppose that the critical  $XDAV$  values of a memory with a known pattern are discovered, as those shown in Table II. Then, if the memory (or at least a sample of the same batch) is tested again, the critical values can be used to identify the MCUs. Thus, the required time to determine the multiple events is reduced even more. Also, independent researchers working with different CY62167 versions can use Table II to classify the registered bitflips in SBUs or MCUs.

The procedure would have to be slightly modified in case of appearance of MBUs in the results. MBUs were not observed in our experiments, but combinations of MCUs and MBUs might appear if other memories are tested. However, this fact does not affect the detection of physically adjacent cells, since our approach does not make use of the word content, and a little modification would be needed only when the addresses are grouped (Line 10 in Algorithm 1). In that case, all the addresses with several affected bitflips should be computed as many times as flipped bits exist in that word.

Unfortunately, a few MCUs escaped the screening of the algorithm (no more than 1 or 2 per experiment). This happens

only when the addresses are related with an uncommon  $XDAV$  value, impossible to extract from the background. However, we have observed that the uncertainty introduced by the undetected events is much smaller than the statistic error margin issued from the relatively low number of events.

### B. Information about internal organization

Another interesting point is the relationship about the anomalous values in the  $XDAV$  vector and the internal organization. This information is not usually at the disposal of the users, but some interesting data can be deduced from Table II. The most interesting fact is that it is doubtlessly demonstrated that, in the transition from 130 to 90 nm, not only did the transistor size decreased, but the memory organization changed as well. Otherwise, the critical  $XDAV$  values would be similar and this is not true.

Unfortunately, it is difficult to use this information to deduce how the memory is internally organized. As explained in Section II-C, the address bits in the studied SRAMs are grouped in 4 sets: to select the quad, block, row and column, respectively. MCUs occur inside a block in cells above/below and/or to the left/right of the central cell. That means that, in the set of critical  $XDAV$  values, mainly two groups of values must appear that are related either to the row or to the column decoder bits. This fact is actually observed in Table II. However, it is impossible to deduce to which direction every  $XDAV$  critical value is related. Besides, during the physical implementation of the SRAM, address bits can be arbitrarily chosen to feed the decoders. Indeed, the placement of the address pins determines the played role in the selection of the cells, and not the bit position inside the logic address. Thus, the study of the actual structure provided by the manufacturer for the 130-nm memory showed that the horizontal MCUs appear as  $XDAV$  values of  $0 \times 010001$ , differing in bits 16 and 0, and vertical ones appear as  $0 \times 000100$ , but also as other apparently unrelated values such as  $0 \times 080000$  or  $0 \times 0C0000$ .

Another point to investigate is the clear dependence of the  $XDAV$  critical values on the stored pattern. An extreme example is the 130-nm memory, in which events in close logic addresses are very unlikely with a  $0 \times FF$  pattern, but constitute 28% of the pairs with  $0 \times 00$  pattern. It was previously reported that the shape of the multiple events are affected by the written pattern [17], and the reason postulated was that the drains of adjacent transistors were differently biased so the charge shared among cells was affected as well. In other words, the electric fields in the space between cells, which depends on the written pattern, facilitates or hinders the propagation of the multiple event in specific directions.

## VI. CONCLUSIONS

This paper has presented a statistical methodology to discriminate MCUs from SEUs in large sets of data. Experimental results issued from radiation ground tests with 14-MeV neutrons and in high-altitude environments (2885 m. above sea level) have been presented and analyzed. Theoretical predictions with MUSCA-SEP3 [16] match the measurements that were carried out, after discriminating MCUs from SEUs.

Modifications in the internal organization were found between the CY62167DV and CY62167EV Cypress SRAMs, and they were validated with private data issued from the manufacturer.

As future work, we are planning to compare these results with other particles, such as protons and heavy ions, as well as to use it in larger sets of data (up to 1800 events).

## REFERENCES

- [1] R. Velazco and F. J. Franco, "Single Event Effects on Digital Integrated Circuits: Origins and Mitigation Techniques," in *IEEE International Symposium on Industrial Electronics (ISIE)*, pp. 3322–3327, Jun. 2007.
- [2] J. Black, P. Dodd, and K. Warren, "Physics of multiple-node charge collection and impacts on single-event characterization and soft error rate prediction," *IEEE Tran. Nucl. Sci.*, vol. 60, pp. 1836–1851, Jun. 2013.
- [3] A. van de Goor, "Using March tests to test SRAMs," *IEEE Design & Test of Computers*, vol. 10, pp. 8–14, Mar. 1993.
- [4] G. Tsiligiannis, L. Dilillo, A. Bosio, P. Girard, S. Pravossoudovitch, A. Todri, A. Virazel, H. Puchner, C. Frost, F. Wrobel, and F. Saigné, "Multiple Cell Upset Classification in Commercial SRAMs," *IEEE Tran. Nucl. Sci.*, vol. 61, pp. 1747–1754, Aug. 2014.
- [5] G. Tsiligiannis, L. Dilillo, V. Gupta, A. Bosio, P. Girard, A. Virazel, H. Puchner, A. Bossier, A. Javanainen, A. Virtanen, C. Frost, F. Wrobel, L. Dusseau, and F. Saigné, "Dynamic Test Methods for COTS SRAMs," *IEEE Tran. Nucl. Sci.*, vol. 61, pp. 3095–3102, Dec. 2014.
- [6] L. Artola, R. Velazco, G. Hubert, S. Duzellier, T. Nuns, B. Guerard, P. Peronnard, W. Mansour, F. Pancher, and F. Bezerra, "In Flight SEU/MCU Sensitivity of Commercial Nanometric SRAMs: Operational Estimations," *IEEE Tran. Nucl. Sci.*, vol. 58, pp. 2644–2651, Dec. 2011.
- [7] M. Wirthlin, D. Lee, G. Swift, and H. Quinn, "A Method and Case Study on Identifying Physically Adjacent Multiple-Cell Upsets Using 28-nm, Interleaved and SECDED-Protected Arrays," *IEEE Tran. Nucl. Sci.*, vol. 61, pp. 3080–3087, Dec. 2014.
- [8] A. Bossier, V. Gupta, G. Tsiligiannis, A. Javanainen, H. Kettunen, H. Puchner, F. Saigne, A. Virtanen, F. Wrobel, and L. Dilillo, "Investigation on MCU Clustering Methodologies for Cross-Section Estimation of RAMs," *IEEE Tran. Nucl. Sci.*, vol. 62, pp. 2620–2626, Dec. 2015.
- [9] J. Beaucour, J. Segura-Ruiz, B. Giroud, E. Capria, E. Mitchell, C. Curfs, J. Royer, M. Baylac, F. Villa, and S. Rey, "Grenoble Large Scale Facilities for Advanced Characterisation of Microelectronics Devices," in *2015 15th European Conference on Radiation and Its Effects on Components and Systems (RADECS)*, pp. 312–315, Sep. 2015.
- [10] F. Villa, M. Baylac, S. Rey, O. Rossetto, W. Mansour, P. Ramos, R. Velazco, and G. Hubert, "Accelerator-Based Neutron Irradiation of Integrated Circuits at GENEPI2 (France)," in *2014 IEEE Radiation Effects Data Workshop (REDW)*, pp. 1–5, Jul. 2014.
- [11] J. Clemente, F. Franco, F. Villa, M. Baylac, S. Rey, H. Mecha, J. Agapito, H. Puchner, G. Hubert, and R. Velazco, "Statistical Anomalies of Bitflips in SRAMs to Discriminate MCUs from SEUs," in *2015 15th European Conference on Radiation and Its Effects on Components and Systems (RADECS)*, pp. 507–510, Sep. 2015.
- [12] D. Falguere and S. Petit, "A statistical method to extract mbu without scrambling information," *IEEE Tran. Nucl. Sci.*, vol. 54, pp. 920–923, Aug. 2007.
- [13] R. Velazco, J. A. Clemente, G. Hubert, W. Mansour, C. Palomar, F. Franco, M. Baylac, S. Rey, O. Rossetto, and F. Villa, "Evidence of the Robustness of a COTS Soft-Error Free SRAM to Neutron Radiation," *IEEE Tran. Nucl. Sci.*, vol. 61, pp. 3103–3108, Dec. 2014.
- [14] A. Hands, P. Morris, C. Dyer, K. Ryden, and P. Truscott, "Single Event Effects in Power MOSFETs and SRAMs Due to 3 MeV, 14 MeV and Fission Neutrons," *IEEE Tran. Nucl. Sci.*, vol. 58, pp. 952–959, Jun. 2011.
- [15] A. Cheminet, V. Lacoste, G. Hubert, D. Boscher, D. Boyer, and J. Poupene, "Experimental Measurements of the Cosmic-Ray Induced Neutron Spectra at Various Mountain Altitudes With HERMEIS," *IEEE Tran. Nucl. Sci.*, vol. 59, pp. 1722–1730, Aug. 2012.
- [16] G. Hubert, S. Duzellier, C. Inguibert, C. Boatella-Polo, F. Bezerra, and R. Ecoffet, "Operational SER Calculations on the SAC-C Orbit Using the Multi-Scales Single Event Phenomena Predictive Platform (MUSCA-SEP3)," *IEEE Tran. Nucl. Sci.*, vol. 56, pp. 3032–3042, Dec. 2009.
- [17] D. Radaelli, H. Puchner, S. Wong, and S. Daniel, "Investigation of multi-bit upsets in a 150 nm technology SRAM device," *IEEE Tran. Nucl. Sci.*, vol. 52, pp. 2433–2437, Dec. 2005.