Laser Fault Injection Methodology for Software Reliability Testing

Federico Reghenzani, *DEIB, Politecnico di Milano, Italy*
Davide Baroffio, *DEIB, Politecnico di Milano, Italy*
Tomas Antonio Lopez, *DEIB, Politecnico di Milano, Italy*
William Fornaciari, *DEIB, Politecnico di Milano, Italy*
Thomas Borel, *ESTEC, European Space Agency, The Netherlands*
Florian Krimmel, *ESTEC, European Space Agency, The Netherlands*

Abstract—In the last decade, laser-based fault injection has become an interesting alternative for testing hardware components against transient faults compared to expensive and complex radiation test campaigns. The laser ability to target specific areas of the device under testing is a valuable feature not present in traditional radiation-based testing. While the laser technique has been the subject of several studies in recent years, an analysis from the software perspective that highlights the practical challenges is missing. This article presents the experimental procedures and the obstacles experienced in achieving repeatability and reproducibility, with a special focus on software reliability testing.

Qualifying the ability of integrated circuits to prevent or tolerate radiation-induced faults is essential for their use in the space environment or, in general, in safety-/mission-critical systems. Radiation effects represent a problem at high altitudes or outside of the Earth's atmosphere, as well as at ground level, due to package impurities and other manufacturing problems [13]. Therefore, their effects must not be ignored for certain high-integrity terrestrial applications.

Traditionally, to assess the radiation tolerance of such systems, the behaviour of their components is validated through irradiation facilities, where a particle beam is used to inject real faults into the devices under test. The particle beam used for this kind of test may be composed of protons, neutrons, electrons, or heavy ions, depending on the specific operational scenario for which a component is tested. Non-negligible economical, administrative, and technical barriers exist to access suitable irradiation facilities [1].

Radiation-based testing provides an estimate of the susceptibility of the hardware to the physical phenomena. In the traditional reliability analyses, any hardware failure leads to a system failure. With such assumption, the reliability of the system weighs entirely on that of the hardware without considering software intrinsic or on-purposely added capabilities to tolerate faults. In recent years, industry, and in particular space agencies [8], [11], began to consider moving from radiation-hardened components to Commercial Off-The-Shelf (COTS) components because of the compelling necessity to decrease costs and time-to-market while increasing volumes. The use of COTS not designed to be radiation tolerant worsens the problem of component qualification and testing software reliability starts to play a key role [12]. As discussed in this article, testing the software usually requires more time than hardware testing because of the introduced complexity and a new variable to explore: the time. The increased time required for software testing exacerbates the problem of scarce availability of radiation facilities.

XXXX-XXX © 2024 IEEE
Digital Object Identifier 10.1109/XXX.0000.0000000

Laser-based testing systems

An alternative solution for injecting realistic faults into semiconductor components consists of exploiting laser-based fault injection, a technique known since the late 1980s [14]. However, laser-based fault injection gained more interest only recently for the following reasons:

- The availability of laser systems and high-performance computers to run them is better nowadays compared to the previous decades.
- Previously, it was used only by hardware manufacturers to preliminary assess the design performance and was not used by final users.

- The advantage with respect to radiation facilities in terms of cost was not so critical in the past, especially for space applications where the budget in the past was higher than today's one.
- The explosion of applications often using COTS and non-radiation tolerant parts reduced the accessibility of radiation facilities, therefore increasing their cost, and consequently increasing the interest for alternative solutions, such as lasers.

Laser-based fault injection is cheaper, presents fewer safety hazards, can pinpoint specific locations in a silicon die, and its energy can be easily regulated, leading to a more controlled experimental setup. In fact, thanks to the controllability of the laser, it is possible to inject various kinds of faults into specific software components and specific time instants. Experimenters can carefully shoot the laser at the hardware locations used during execution. As an example, consider the case in which we want to inject a fault into an instruction operand. With laser testing, we can potentially shoot at the register containing the value while the value itself is alive in that register.

Despite these advantages with respect to traditional radiation testing, laser testing is currently considered only at the beginning of the experimental phase, as it is used mainly in preliminary evaluations and not for actual validation. The reasons are manifold: (1) the limited availability of well-defined practices to conduct the experimental campaign, especially when testing the reliability of the software; (2) uncertain correlation between the results obtained through laser-based testing and traditional radiation testing; (3) the lack of many commercial lasers specifically tailored to this task, which makes the existing solutions very expensive[1]; (4) the non-trivial preparation process of the components to be tested, which involves decapping the chip from its package and polishing the exposed die.

Laser injection is used to test Single-Event Effects (SEE) in general, most commonly to test the detection of Single-Event Upsets (SEU), Single-Event Transients (SETs), and Single Event Latchups (SELs) [7].

This article provides an overview of how to perform sound laser-based fault injection from a methodological standpoint, with a focus on software reliability testing. First, we discuss the architecture of the experimental setup and the preliminary phases required before actually running the test. Then, we describe how to configure the laser for the experiments via empirical and systematic procedures. Finally, we provide the methodology for performing device exploration and actual software testing campaigns.

State of the Art

Laser fault injection has been widely used to inject different types of faults in microprocessors and memories, to either test their radiation tolerance or to drive the radiation-hardened hardware design process [10]. A vast majority of the scientific papers on the subject provide results on the events occurring in SRAM memories (e.g., [7], [15]). Some works exploited the laser to test DRAMs (e.g., [6], [3]). Other researchers studied the effect of the laser on FPGA memories [9], [16]. All of these works focused on deriving hardware metrics related to electrical characteristics. Regarding software testing using lasers, only a few articles have considered laser fault injection, but mainly as an attack vector to circumvent security protections (for example, bypassing a cryptographic algorithm) [4], [5].

Two recent technical reports by the European Space Agency[2] and by the Defense Threat Reduction Agency[3] provide guidelines for SEE laser testing. These two documents contain detailed information on how lasers interact with the micro-components of integrated electronics and provide an important background for conducting SEE experiments.

For illustrative purposes, this paper contains data obtained from our experiments on a specific microcontroller of interest to us. However, it is neither the goal of the article to claim any performance of such a microcontroller, nor to perform an in-depth analysis of it, nor to provide generally applicable numbers for any possible scenarios, that, instead, heavily depend on the specific device tested and its manufacturing process. The quantitative results collected during our experimental campaign, not reported here, were aimed at proving the effectiveness of some Software-Implemented Hardware Fault Tolerance approaches [2], which are out of the scope of this article.

Reference system

The device used in our research is a microcontroller unit containing the main memory, flash mass storage,

[1] At the time of writing, to the best of our knowledge, only a single company exists in the world that commercializes this equipment.

[2] https://indico.esa.int/event/444/contributions/7789/attachments/5308/8540/Single-Event%20Effects%20Testing%20with%20a%20Laser%20Beam%20-%20Guidelines.pdf

[3] https://apps.dtic.mil/sti/trecms/pdf/AD1204115.pdf

and many peripherals in a single package. The considerations presented in this paper are general enough to be applicable also to microprocessors and other devices.

Overall architecture

The overall architecture of the interconnected components is depicted in Figure 1. The microcontroller running the software subject of the testing campaign is called *Device Under Test (DUT)*. The DUT used as an example in this article, called *reference DUT*, is an STM32F427VIT6TR 90nm microcontroller. The DUT is soldered on a custom-built PCB that we call *Board Under Test (BUT)*. It must be noted that if the DUT needs to be decapped from the bottom, the BUT needs to have an opening below the DUT to allow the laser to access the die. The BUT is then connected to another component that governs the experiments. In our experimental evaluation, this component was also a custom-designed PCB even if, in principle, it could also be a debugger or any other suitable device, depending on the specific tests that should be performed. Whichever the case, these devices are then usually connected to a host PC for data visualization and recording. For simplicity, we refer to this device (or set of devices) as *testing equipment* (this definition does not include the laser and related devices). The BUT needs to be fixed to a high-precision mechanical *holder* that keeps it in the correct position and allows the experimenter to regulate the inclination of the BUT with respect to the laser beam. The X, Y, and Z references used in this article are depicted in Figure 2a. To reduce the interference from the environment, the *holder* is positioned over a compressed-air stabilizer, isolating it from external vibrations. Finally, the *laser* is the emission source, oriented to have the light beam perpendicular to the die surface of the DUT, and it typically integrates an optical microscope camera sensible to IR light. The die is indeed transparent to IR light: the IR camera can see through the silicon and take (mirrored) pictures of the circuit feature. The camera is essential for understanding where the laser is pointing at. The laser and the camera have swappable lenses, allowing the experimenter to change the magnification level. A picture of the whole setup is visible in Figure 2b.

Testing equipment

Each specific experiment needs a different configuration of the testing equipment. With the exception of specific needs, we identified some capabilities that the testing equipment must generally have regardless of the objectives of the specific experimental campaign:

- A method to read from and write to specific memory locations of the DUT. While the DUT itself can perform these tasks, the testing equipment must have access to these functionalities, allowing the experimenter to perform manual inspection and modifications on the DUT memories.
- Electrical connections to perform I/O from/to the testing equipment to/from the DUT to test the behaviour of software as it would be in a real scenario.
- A programming device, if not integrated with the BUT, which allows the experimenter to change the code or reprogram the memory areas of the DUT.
- An automatic, configurable, and fast power disconnection circuit or device that constantly monitors the power consumption and quickly reacts to any *Single Event Latchup (SEL)* by removing the voltage to the BUT/DUT. This device must have a way to log these events and immediately inform the experimenter. This device is essential to avoid breaking the DUT since a manual intervention is unlikely to react fast enough.
- A way to perform a hard reset of the DUT in case its firmware gets stuck for any reason. This can be achieved either by removing the power supply or by using the dedicated reset pin usually available on the DUT.

Preliminary phases

The DUT needs to be prepared before beginning the actual experiments, i.e., the plastic package must be partially removed to allow the laser to access the die. Manufacturers do not often provide images or other information on the die placement inside its package. Therefore, an X-ray picture should be taken to understand how the bonding wires are attached to the die (which can be either in a "normal" or "flip-chip" configuration) and their location. The X-ray picture of the reference DUT is visible in Figure 3 and shows it to be a normal chip (not flip-chip), where the bonding wires are attached at the top edges of the die (and, therefore, the metallization layer is also on the top).

The following considerations must be taken into account before starting the decapping process for the subsequent laser testing:

- Opening side: the position of the metallization layer determines the opening side for laser testing, which often is the opposite of the one used for radiation testing, as the infrared laser does not penetrate the metallization layer, while

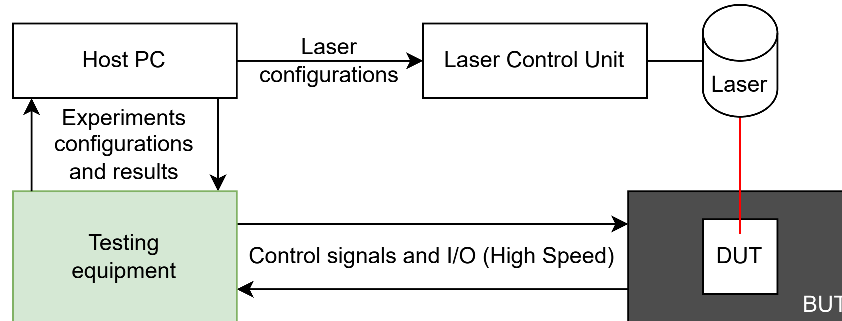
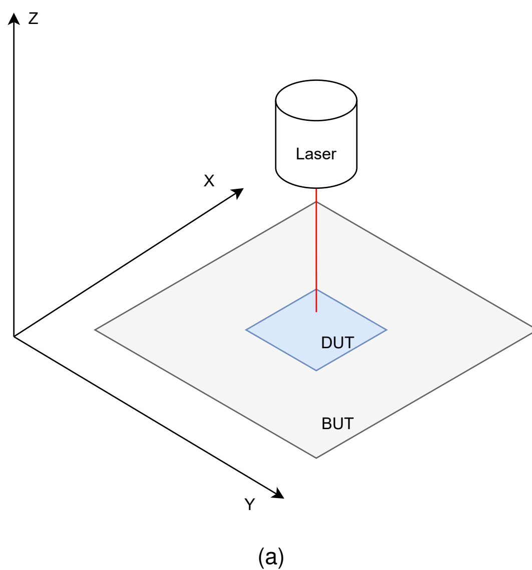
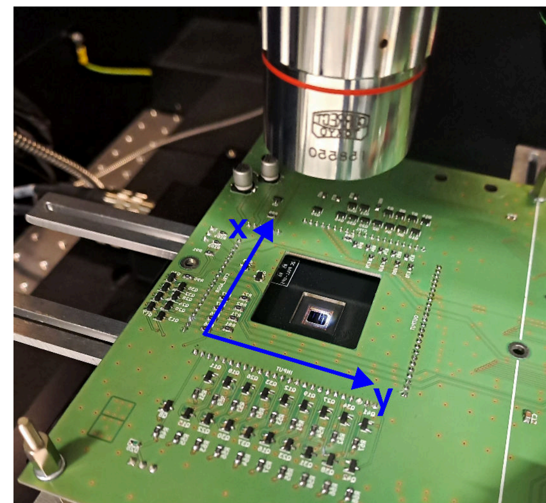


FIGURE 1: Schema of the experimental setup.



(a)

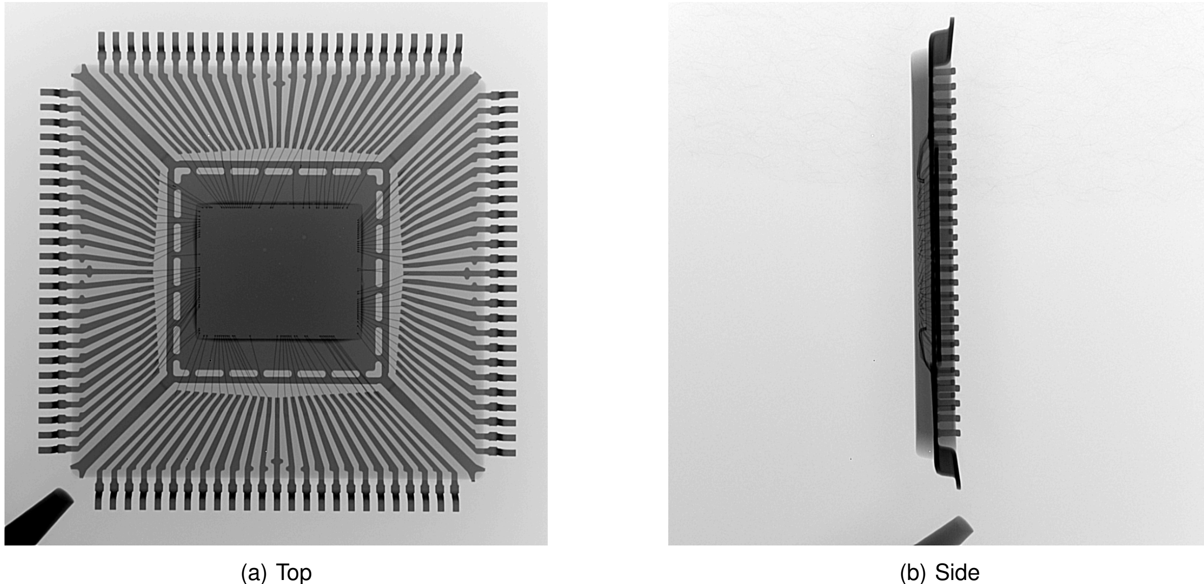


(b)

FIGURE 2: Figure (a) shows the coordinate reference system used in this article in isometric view. Figure (b) shows a picture of the experimental setup: the DUT is seen inverted because decapped from the bottom; the BUT is the black PCB under the green PCB where the DUT is soldered; the green PCB contains electronics to support the experiments; the *holder* is not visible because below the green PCB, but its metallic bars where the PCB is attached are visible; finally, the lens of the laser is visible perpendicular to the DUT/BUT.

it does penetrate the silicon. Therefore, if the device is not a flip-chip (like in the example of Figure 3), the package must be removed from the bottom (because the metallization layer is on the top side). Conversely, if the device is a flip-chip, the package must be removed from the top (because the metallization layer is on the bottom side).

- Opening process: depending on the lab capabilities and on the side that must be opened, the decapping process can be performed by using either a CNC milling/grinding/polishing machine with sub-micrometer precision, an oxygen plasma etching machine, chemical wet etching processes, laser ablation, or a combination of them. In our case, a copper heat spreader was present below the plastic package, and a silver epoxy paste known as *die attach* separated the metal plate from the die. After removing the copper plate with a milling machine, the die attach was removed by hand with a scratching tool.
- Polishing process: in order to prevent an unacceptable distortion of the laser beam wavefront, which would have a significant impact on the focused spot size, an optical grade polishing of the back side of the silicon is necessary. For laser testing, it is extremely important to have a silicon-air interface that is as even as possible, to avoid experimental biases due to non-



(a) Top (b) Side

FIGURE 3: X-Ray pictures of the DUT. It is possible to notice the bonding wires connected to the top of the die: it is not a flip-chip.

uniform light refraction. In addition, a polished silicon surface allows a better visualization of the structures using the NIR (Near Infrared) camera connected to the microscope.

- Silicon thickness measurement: The choice of the operating parameters for the laser experiments strongly depends on the optical properties of the polished die. For the sake of reproducibility, it is therefore important to quantify, among other factors, the thickness of the silicon volume traversed by the laser beam via a specifically designed interferometer.

Laser parameter configurations: empirical and systematic methods

Selecting the laser magnification

The lasers used in SEU testing are usually provided with different lenses for different magnification levels. In our experimental setup, we have x5, x20, and x100 magnification lenses. As expected, increasing the magnification level reduces the power needed to cause SEUs and the affected areas. This article focuses on experiments with the most powerful lens (x100) that reduces the laser spot size to about 1µm. This size is at the physical limits of the laser-based techniques: smaller spot size would be possible only for wavelength smaller than 1064 nm, which are however not optimal for SEU testing.

Laser alignment procedure

The DUT is rigidly attached to a mechanical structure that allows the tuning of the inclination with respect to the three axes at the micrometer scale. This way, the DUT can be placed on a plane perpendicular to the laser beam to achieve the best penetration uniformly across the whole device. The alignment procedure consists of selecting a single origin, aligning the laser on the Z axis until the origin point is in focus, then moving to different far points on the X and Y axis and refocusing them by manually tuning the mechanical structure inclination. This process lasts several minutes and must be periodically performed (at least once a day). The focus is verified using the microscope camera included in the laser structure and which shares the same lens with the laser.

Performing the alignment for some devices presents non-trivial challenges. In particular, the laser focus is affected by the die surface polishing process, which must be as uniform as possible to have a constant incident energy across the whole die surface. A further challenge is represented by the phenomenon known as *die relaxation*, a slight deformation of the die that occurs when the package is removed from only one side of the component: the package still present on the other side may create a force that is not counterbalanced anymore by the now removed package, "incurvating" the die. The magnitude of the Z-axis deformation varies with the size of the die itself and may reach up to $10\mu m - 30\mu m$. This phenomenon

not only makes the focus inevitably different across the whole die, but also renders the polishing process less effective, as the polishing machine cannot apply the same pressure over the entire surface and, consequently, cannot remove as much material from the corners as it does for other areas.

After the laser is aligned, it is possible to take several pictures and create a mosaic of the whole die, as shown in Figure 4a. This picture helps with the identification of the components and with the navigation across the die in both manual and automatic exploration.

Selecting the laser power level

The power level of the laser is a parameter that needs to be explored before beginning the actual testing of the software. If data from radiation testing is available, this can be used to estimate a suitable power level. If this information is not available, a trial-and-error strategy (possibly together with advice from experts) is the only way to proceed. A single shot may not capture this information entirely because, other than the statistical effects inherent in measurements, some areas of the device may be less sensitive than others, or even completely insensitive to the beam. Consequently, a small exploration of a portion of memory is necessary to get some valid data.

The laser power also impacts the probability of generating a SEL, which may not be desirable when testing software, because it can damage the chip permanently or put the processor in an indeterminate state. Indeed, tolerating a SEL is mainly a hardware issue, because, while it affects software, we can recover from a SEL only via hardware methods (usually by power-cycling the device). Therefore, in general, when testing the resilience of software to transient faults, SELs are not usually a desired fault type to inject.

We performed the power level exploration by uniformly scanning a region of $2\,808\mu m^2$ of the SRAM with a step of $3\mu m$ on both axes, for a total number of points of 312 per scan. Each byte was filled with a *Z pattern* (01011010)[4] as to make the results not biased by any possible difference in sensitivity to bit-flips between 0-bits and 1-bits. The result of the parameter exploration is shown in Figure 5, presented as the number of SEUs per laser shot (total number of SEUs divided by 312). From these results it is possible to draw the following conclusions:

- There is a power level below which no event occurs ($< 75pJ$ in our case)
- There is a power level above which SELs are generated ($> 500pJ$ in our case)
- The curve trend is non-uniform: in the power level range [75; 200], the derivative is 4.18 SEU/shot/pJ. Then, there is almost a plateau between power levels [200; 300], and then the trend is linear with a derivative of 0.96 SEU/shot/pJ in the range [300; 500].

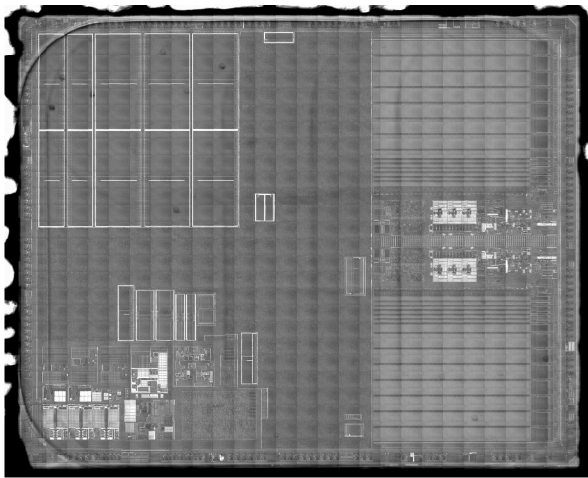
All the subsequent experiments we performed that are reported in this article were conducted with power values between 250 and 300, so that the power level is far from the dangerous SEL area, while it has a sufficient number of SEU events per shot and low variability. These data do not represent general information applicable to any DUT, since they are highly dependent on the specific design of the device and other factors that are difficult to know a priori. Therefore, a device-specific power level exploration is essential at the beginning of any experimental campaign to preliminary assess the necessary power level to use and the device cross-section.

It is worth mentioning that while such a parameter exploration is relatively easy to perform for an SRAM, the same analysis applied to the core part of the processor is much more difficult. The reasons are multiple: 1) the core part may have a different manufacturing process, leading to a different power level curve, thus the exploration needs to be redone from scratch; 2) detecting faults in the core is difficult because they may be masked by other logic in the core itself and because we may lack support to access the core memory areas; 3) identifying where the memory areas are is also a challenging process because they are not immediately recognizable, like the main memory is, in the IR picture (see Figure 4a: some peripheral registers are visible within the core because of their relatively large size, which is in the order of several *kB*, but the CPU registers are only a few bits large and are not visible). These issues render the usage of a laser to test the processing core against SEUs rather difficult.

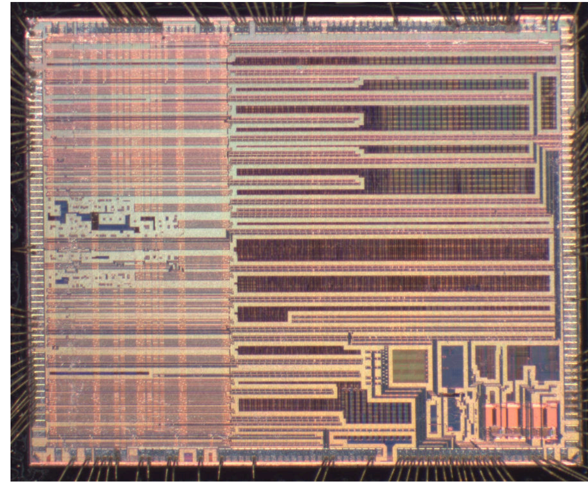
Probability of a single and multiple bit flips

On average, it is possible to generate a single-bit flip by selecting a power level (Figure 5) so that the average SEUs/shot is close to 1. However, the number of bit flips per shot has a very high variance, as can be immediately seen from Figure 6a, which represents the spatial distribution of the laser shots, colored according to the number of bit-flips that they cause, on a small

[4] The pattern is called "Z" as 01011010 represents the binary encoding of the character `Z` in ASCII-8. This pattern contains all possible two-bit digrams in the same byte: 00, 01, 10, 11.



(a) Bottom (IR)



(b) Top (Visibile)

FIGURE 4: The two figures show the same DUT when decapped from the bottom (via IR-light microscope) and when decapped from the top (via a visible-light microscope). The two images are mirrored with respect to each other on the horizontal axis. In sub-figure (a), it is possible to notice the SRAM memories on the top-left corner, the two flash memories on the right, the analogue peripherals in the bottom-left corner, other isolated memories, and the rest dark area is the die logic. The same components are visible (mirrored) in sub-figure (b) even if the view is obstructed by the presence of the metallic substrate.

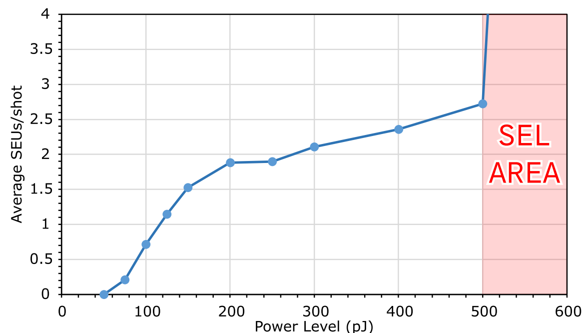


FIGURE 5: Number of average SEUs per laser shot depending on the power level.

portion of SRAM with all bits set to “1”. The reason behind this particular appearance lies in the topology of the memory transistors arrangement on the die. The blue band in the figure corresponds to the locations that are either ineffective for switching the cell’s state, or that are dedicated to switching the cell’s value to “1” and, thus, have no observable effect in this case, as the cells are already set to “1”. As for the other colored areas, the SRAM cells are so close that a laser shot of 1 μm either hits more than one cell, or charges the device in such a way that Mutiple Cell Upsets (MCUs) easily occur. As a consequence, the number of actual single SEUs is indeed very limited. To better characterize this effect, we portray in Figure 6b the flip count histogram of a uniform portion of SRAM, stimulated with 300 pJ of energy per shot. We expected an average value of about 2 SEU/shot, according to the results shown in Figure 5), however, the distribution has a high variability and double flips have actually a low occurrence. In conclusion, while it is possible to generate 1-bit events *on average*, it is very difficult to generate actual SEUs by laser injection at this technology scale, as multiple cells are very easily perturbed and, therefore, MCUs are easy to trigger.

The importance of die cleanliness, focus, and stabilization

Die cleanliness. After backside opening of the component and removing the die attach either mechanically or chemically, the component can be cleaned with deionized water, acetone, or special PCB cleaner liquids in an ultrasonic bath. After soldering the component on the BUT, if there are more contaminations on the component, then the daughter board can be cleaned again in the ultrasonic bath. The previous Figure 4a shows a clean die with very few dirty spots. We decided to analyze the effect of one of these remaining spots on SEU susceptibility. We uniformly explored the small region with a 3×3 μm grid, checking the number of SEUs that occurred after each laser shot for every point of the grid. The result of the analysis is depicted in Figure 7b in the same style as

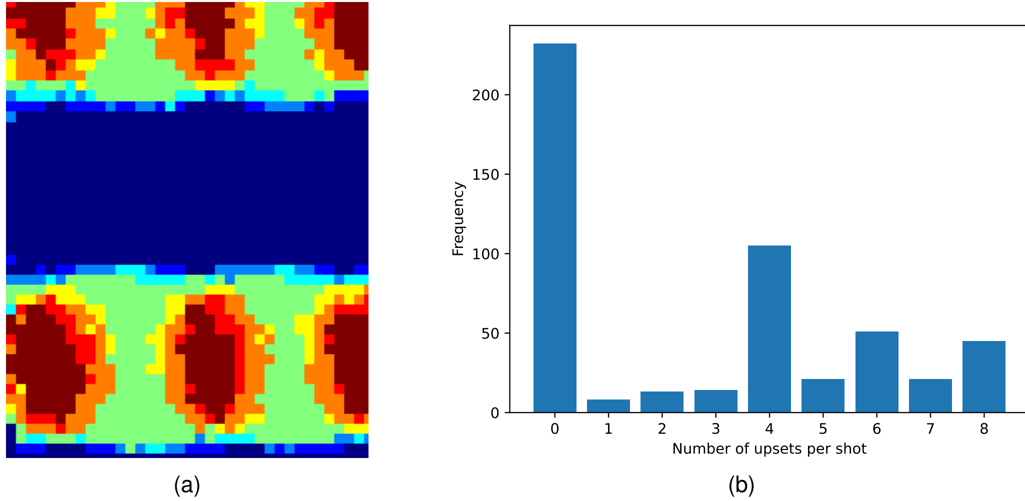


FIGURE 6: Sensitivity test of a small $100nm \times 100nm$ area of the SRAM, explored with the finest possible step of $0.1nm$ (color scale is blue=0 events – dark red=8 events).

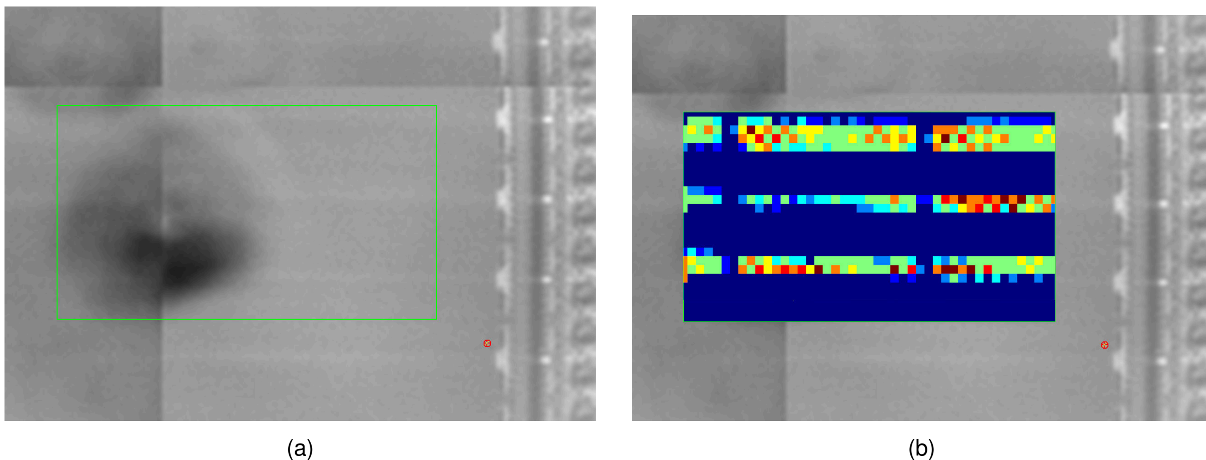


FIGURE 7: The exploration of a small portion of the SRAM where a spot was present. The heat map in sub-figure (b) shows the number of bit-flips when the laser is shot at that point (color scale is blue=0 events – dark red=8 events).

Figure 6a. By looking at the heat map, it is easy to notice that the area in the center of the spot on the left is less sensitive than the clean area on the right. To provide a rough metric, we detected 492 SEUs in the left half of the area, and 606 SEUs in the right one, i.e., the spot caused a decrease of -23% of the sensitivity. Additionally, the mean value of SEUs per shot (excluding the points where no event occurred) changes from 4.03 to 4.36 with the standard deviation changing from 0.69 to 0.81. While the difference may look small, this can have a significant effect on a large exploration, if the number of dirty spots (or their size) is large enough. This result confirms that even an almost transparent spot can deviate or attenuate the laser beam in a significant way, hiding sensitive components, and possibly reducing their sensitivity. Therefore, the cleaning of the die plays a crucial role in obtaining valid and uniform data.

Focus and stabilization. Before starting the experiments, the DUT must be aligned on the XY plane so that it is perfectly perpendicular to the laser. With our equipment, the alignment is made possible by a set of gears that allows a sub-micrometer fine-grain regulation of the inclination of the *holder*. The optical focus of the laser lens is then adjusted by varying its Z position. This regulation is essential for having a precise focus set at a fixed depth within the silicon die. It is crucial that the DUT is placed perpendicular to the Z axis. Otherwise, during a planar scan, different parts

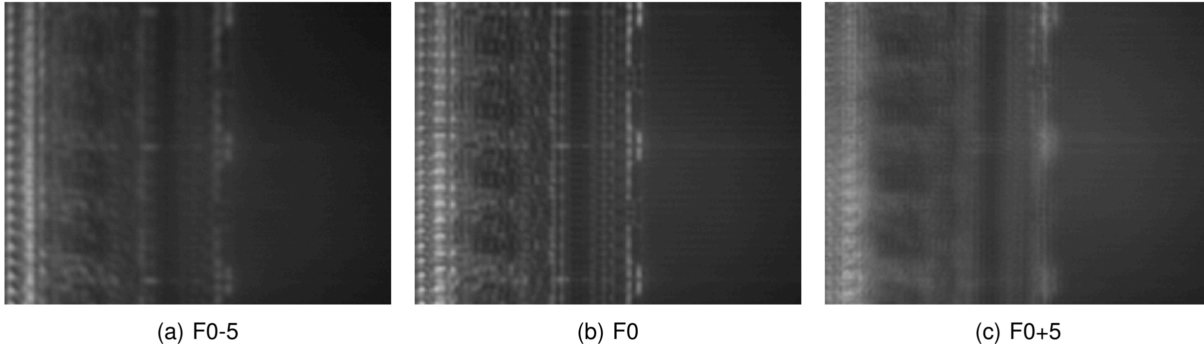


FIGURE 8: A particular of the SRAM memory as seen by the IR microscope with the three considered focus.

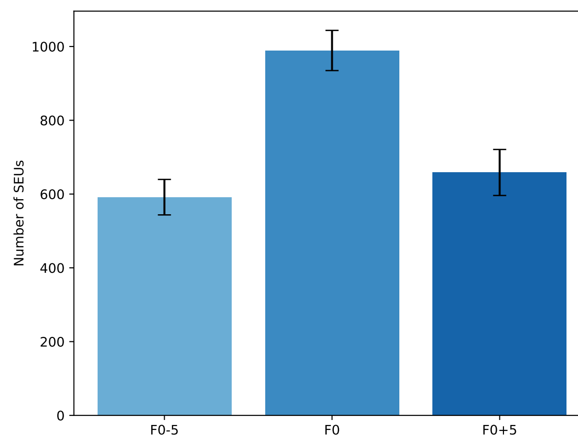


FIGURE 9: Number of SEUs in the same region when uniformly exploring it for 702 laser shots and varying the focus. The laser was on optimal focus (F0) and then manually shifted off-focus (F0+5: Z position $5\mu m$ higher than optimal, F0-5: Z position $5\mu m$ lower than optimal).

of the DUT would be at a different focus, resulting in a gradual change in the amount of deposited charges by the laser. This irregularity either enhances or attenuates the effect of the laser depending on the location, possibly leading to incoherent results.

The laser was equipped with auto-focus software, but the results were not satisfactory and the focus was almost always performed manually. We performed a small experimental campaign to determine the importance of having a good focus on a small region of SRAM, which showed that a variation of $\pm 5\mu m$ from the expert-determined focus resulted in a reduction in the number of injected SEUs of $30-40\%$ (Figure 9). The variability (standard deviation) in the number of SEUs was, instead, not impacted.

From this result, we learned the importance of having a good and stable focus for reproducible experiments and uniform long-run explorations. Indeed, when performing long testing campaigns that endure for many hours without supervision, the focus and alignment tend to drift due to temperature cycles and other environmental effects. Empirically, we determined that the focus drift in our setup amounts to roughly 5 to $10\mu m$ every 24 hours, which forced us to perform re-alignment and refocus every 12 hours to get consistent data. However, this value is not consistent over several days (the focus drift was often lower than expected). We did not investigate this problem further since any solution would require an automatic auto-focus system, which is not currently available on the market.

Exploration patterns

Most of the experiments consist in exploring an area on the DUT, usually rectangular, on the XY plane. The exploration can be performed mainly in three ways:

1) *Grid-based*: the exploration follows a predefined grid. The laser moves to a point of the grid, stops, shoots, and then moves again. This process is repeated until all points in the grid have been explored. In this scanning mode, additional

parameters must be set, most importantly the waiting time before and after each laser shot, which is essential to let the mechanical parts stabilize before the next shot.

2) *Free-scan*: in this mode, the laser moves continuously in a predefined primary axis and never stops. When the laser arrives at the area boundary, it moves for a configurable distance in the secondary axis and then inverts the direction on the primary axis. With free-scan, the shots are triggered by a configurable clock signal and are not position-dependent. It must be noted that the inversion of movement takes time, so the deceleration and acceleration time must be taken into account when performing a uniform exploration.
3) *Random-walk*: the laser moves in a random direction and jumps back in another random direction when the area boundaries are hit. The laser shot may happen per single point (like grid-based) or randomly, assuming a continuous movement (like when free-scanning).

There is no unique answer on which of these three options is preferable because it depends on the experimental parameters. For instance, a very small DUT may be scanned with a very fine grid-based approach in a reasonable time frame, while a more complex device needs to resort to free-scan or random-walk.

Software testing considerations

Auxiliary software

The DUT can usually run in two macro-configurations: the software testing mode, in which it executes the actual software, and the command-based mode, in which is waits and execute the commands provided by the testing equipment (such as read/write from the memory). Both of them required the device to have some kind of software mechanism (often part of the operating system) to run a set of basic functions, including:

- Reprogramming the DUT, making it possible to switch between different testing configurations.
- Reading specific memory addresses of the DUT.
- Writing specific memory addresses of the DUT.
- Notifying the testing equipment that an error occurred on the DUT, useful mostly in the software testing mode.
- Resetting the state of the DUT (in a soft-mode, compared to the hard-mode provided by the electrical signal previously described).

The memory used by this set of functions must be isolated in a well-known address space, to avoid hitting them with the laser and create spurious errors in the testing software itself.

Device exploration

The first data obtained after preparing the setup are usually needed to configure the laser power and the other parameters that we discussed in the previous sections. In order to do these tests, the command-mode explained above is sufficient. This exploration is mostly straightforward for memories: we fill the memory with a known pattern, the laser shot is triggered, and then we check if and which addresses of the memory have been affected. Reading/Writing from/to the DUT memory is a time consuming operation and this delay may slow down the exploration if at each shot the DUT memory must be completely read. The optimization of this read/write must be take into account.

Exploring core and peripheral components is instead challenging. Firstly, it is not easy to identify each component on the die by visual inspection via microscope, making difficult to verify after the shot which register or peripheral is affected. Then, the outcome of a laser shot may be modified or masked by other (non-targeted) logic circuits. For instance, let us consider the internal temperature sensor of a microcontroller: hitting it with a single laser pulse makes difficult to distinguish this event from a laser shot that hits the Analog-to-Digital Converter (ADC) that reads the temperature analog signal. Other problems appear if we target external interfaces – like UART, Ethernet or I2C –, because the layers of complexity and the possible interaction with external components make the analysis of the results even more challenging.

Time problem of an exhaustive exploration Exploring the whole die using a grid-base exploration method with a small step is impractical. In our case, the area of Figure 4a is approximately 5×4 mm; exploring such space by using a grid pattern of $100nm \times 100nm$ means a total of 2×10^9 points to explore. Even if each point takes only 10 ms – which is very optimistic, because the time required by the mechanical laser assembly to stabilize is longer than that –, it would require approximately 115 days for just one single scan. Moreover, as explained in the previous section, to map the entire memory, we may need to run a data collection script after each laser shot to check whether a SEU occurred. Even when optimized, this operation still takes some milliseconds to complete, exacerbating the problem. It is important to remark that selecting a

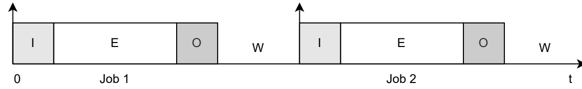


FIGURE 10: Modeling of the execution of a task.

larger grid size may lead to biased result, depending on the technology node. Let us consider Figure 6a where a very small SRAM area is scanned with the 100×100 nm grid size. If we suppose to increase this step to 2×2 μm, it is possible that we inadvertently align the grid to the insensitive rows (blue) or the sensitive rows (colored) – sized approximately 2 μm in the vertical axis – and bias the results. In conclusions, if grid-based exploration is planned, its parameters must be carefully selected to avoid unfeasible exploration runs in term of required time and/or biased results. Based on our experience, grid exploration for SEUs should be used only for testing very small portions of the die and not to perform an exhaustive analysis, which may take too much time or introduce biases.

Alternative solutions to perform an uniform exploration
In order to overcome the problems of the grid-like solution, the other two alternatives (free-scan and random-walk) should be taken into consideration for performing an exploration that can, statistically, cover the whole die. When configuring the laser to shoot at a constant, time-triggered rate while moving the laser head at a constant speed, the shots effectively appear as uniformly distributed across the whole surface. This method reduces the scanning time, as the laser assembly does not have to stop, stabilize, and then shoot, as it can deliver pulses while moving at a high speed (even 1 mm/s or more). In addition, by repeating the random scan of the same area multiple times, this method more closely simulates the random impact of ionizing particles over the entire area being covered. The downside of this technique resides in the difficulty of assessing whether a statistically sufficient number of samples have been acquired.

Testing the actual software

The software on the DUT runs one or more finite-time tasks repeatedly, where each instance of a task is called *job*. The execution timeline of a task is depicted in Figure 10. The *testing equipment* should be equipped with a mechanism that checks the correctness of the output for each job. There are multiple ways of doing so. For instance, the *testing equipment* could run a parallel copy of the code that runs on the DUT, checking the correctness of the the output in real-time. Another approach is based on the concept of a *golden run*, in which the DUT runs fault-free (i.e., with the laser turned off) a pre-defined set of jobs that will necessarily result in a correct reference output. The computed results are then saved by the *testing equipment* as golden runs, and used as a baseline for checking the correctness of the outputs while injecting faults into the running DUT.

With Figure 10 as a reference, let us discuss on what happens when a laser shot causes a SEU. This event may happen in any of the highlighted time regions:

- E: it may happen during the actual execution of the time. This is the *normal* case and it is surely something we would like to test.
- I or O: during these phases, the *testing equipment* is writing/reading the the input/output to/from the DUT, respectively. Whether errors occurring due to faults injected within these intervals have to be considered valid is a choice that depends on the specific scenario. Indeed, the *testing equipment* may be slower to provide I/O than a real application would do, therefore increasing the exposure time of the task and making the results more pessimistic.
- W: in this case, the laser shot will not likely produce any visible effect because the DUT is not running any job (it is still possible that it hits some OS routines or other system code).

From this analysis, it becomes evident that, for each job, we need to know during which phase of the computation the laser pulse was delivered, so that we can decide which of the measurements has to be discarded in post-processing. Moreover, the *testing equipment* should minimize the W time, to avoid getting too many laser shots outside of the actual execution interval of the job, wasting experimental time.

In a real system, it is possible to have multiple tasks running on the DUT and scheduled by an operating system. If the experimenter would like to test this software scenario with the above presented model, a solution is to consider the task as the hyper-period of the software system, i.e., the time frame in which the scheduling decisions of the set of jobs spawned by the tasks repeat. Thus *E* becomes the length of the hyper-period. In such a way, the testing occurs in a well-defined time frame that can be safely scaled to the traditional reliability hourly metrics. It should be noted that, theoretically, the hyper-period may be extremely large that an exploration would be unfeasible because too slow, however, this case is not common in industrial applications.

Conclusions

The use of laser testing is still experimental and presents many challenges, but it is advantageous in many scenarios, especially for software testing. The following two paragraphs recap the advantages and disadvantages of this technique identified in this experimental campaign.

We identified the following **benefits** in using a laser to test software:

- Performing experiments on specific memory regions and internal components. It is possible to test only one peripheral or exclude a memory area (for instance, because it contains some auxiliary code to perform the experiment).
- Injecting faults at specific memory locations with a good level of accuracy. With some degree of uncertainty, it is possible to hit a specific memory cell and to cause a SEU in a precise direction ($1 \rightarrow 0$ or $0 \rightarrow 1$).
- Ease in changing power level and, therefore, testing the DUT under different conditions. Correlation between laser power and radiation energy exists, allowing the experimenter to obtain a rough and preliminary idea of the device cross-section.
- It is, in general, less expensive and less risky compared to radiation sources and allows interactive testing with short prototype develop-test-evaluate cycles.

Conversely, we identified the following **challenges** in using a laser to test software:

- Due to statistical effects, it is very difficult to consistently get only SEUs, while MCUs are the most easily-obtainable type of SEE.
- The exploration time is problematic if a grid-based exploration is used and becomes easily unfeasible if the region of interest is too large. Obtaining uniformly distributed events requires a careful planning and it is non-trivial.
- Sample preparation and laser parameters (e.g., focus) play a crucial role in the final measurements, hindering the reproducibility of the experiment and its validity.
- Due to the previous disadvantage, the testing of the core is more difficult, also because it is difficult to identify a power level to generate events in the core.
- It is easy to introduce biases in the distribution of events over space and time.

ACKNOWLEDGMENTS

This work has received funding by: the European Space Agency (OSIP no. 4000133770 / 21 / NL / MH / hm), the National Resilience and Recovery Plan (PNRR) through the National Center for HPC, Big Data and Quantum Computing, and the Italian project PMDI C-ETSI (Bando MISE CUP B49J24001210005).

REFERENCES

1. Rubén García Alía, Andrea Coronetti, Kacper Bilko, Matteo Cecchetto, Gerd Datzmann, Salvatore Fiore, and Sylvain Girard. Heavy ion energy deposition and see intercomparison within the radnext irradiation facility network. *IEEE Transactions on Nuclear Science*, 70(8):1596–1605, 2023.
2. Davide Baroffio, Federico Reghenzani, and William Fornaciari. Enhanced compiler technology for software-based hardware fault detection. *ACM Trans. Des. Autom. Electron. Syst.*, 29(5), September 2024.
3. A. Bougerol, F. Miller, N. Guibbaud, R. Leveugle, T. Carriere, and N. Buard. Experimental demonstration of pattern influence on dram seu and sefi radiation sensitivities. *IEEE Transactions on Nuclear Science*, 58(3):1032–1039, 2011.
4. Jakub Breier, Dirmanto Jap, and Chien-Ning Chen. Laser profiling for the back-side fault attacks: With a practical laser skip instruction attack on aes. In *Proceedings of the 1st ACM Workshop on Cyber-Physical System Security*, CPSS '15, page 99–103, New York, NY, USA, 2015. Association for Computing Machinery.
5. Brice Colombier, Paul Grandamme, Julien Vernay, Émilie Chanavat, Lilian Bossuet, Lucie de Laulanié, and Bruno Chassagne. Multi-spot laser fault injection setup: New possibilities for fault injection attacks. In Vincent Grosso and Thomas Pöppelmann, editors, *Smart Card Research and Advanced Applications*, pages 151–166, Cham, 2022. Springer International Publishing.
6. S. Duzellier, D. Falguere, L. Guibert, V. Pouget, P. Fouillat, and R. Ecoffet. Application of laser testing in study of see mechanisms in 16-mbit drams. *IEEE Transactions on Nuclear Science*, 47(6):2392–2399, 2000.
7. Pascal Fouillat, Vincent Pouget, Dale McMorrow, Frédéric Darracq, Stephen Buchner, and Dean LEWIS. *Fundamentals of the Pulsed Laser Technique for Single-Event Upset Testing*, pages 121–141. Springer Netherlands, Dordrecht, 2007.
8. Robert F Hodson, Yuan Chen, John E Pandolf, Kuok Ling, Kristen T Boomer, Christopher M Green,

Jesse A Leitner, Peter Majewicz, Scott H Gore, Carlton S Faller, et al. Recommendations on use of commercial-off-the-shelf (cots) electrical, electronic, and electromechanical (eee) parts for nasa missions. Technical Report 20205011579, NASA, 12 2020.

9. Fernanda Lima Kastensmidt, Lucas Tambara, Dmitry V. Bobrovsky, Alexander A. Pechenkin, and Alexander Y. Nikiforov. Laser testing methodology for diagnosing diverse soft errors in a nanoscale sram-based fpga. *IEEE Transactions on Nuclear Science*, 61(6):3130–3137, 2014.
10. D. McMorrow, J.S. Melinger, S. Buchner, T. Scott, R.D. Brown, and N.F. Haddad. Application of a pulsed laser for evaluation and optimization of seu-hard designs. In *1999 Fifth European Conference on Radiation and Its Effects on Components and Systems. RADECS 99 (Cat. No.99TH8471)*, pages 198–204, Fontevraud, France, 1999. IEEE.
11. Mikko Nikulainen and Ferdinando Tonicello. Utilization of cots in esa missions, 6 2021.
12. F. Reghenzani and W. Fornaciari. Towards certifiable software-implemented hardware fault tolerance. In *Proceedings of IEEE 14th International Symposium on Industrial Embedded Systems (SIES)*, page 7, 2024.
13. Federico Reghenzani. Enabling software technologies for critical cots-based spacecraft systems. In *Proceedings of the 20th ACM International Conference on Computing Frontiers*, CF '23, page 236–242, New York, NY, USA, 2023. Association for Computing Machinery.
14. A. K. Richter and I. Arimura. Simulation of heavy charged particle tracks using focused laser beams. *IEEE Transactions on Nuclear Science*, 34(6):1234–1239, 1987.
15. Cyril Roscian, Alexandre Sarafianos, Jean-Max Dutertre, and Assia Tria. Fault model analysis of laser-induced faults in sram memory cells. In *2013 Workshop on Fault Diagnosis and Tolerance in Cryptography*, pages 89–98, Los Alamitos, CA, USA, 2013. IEEE.
16. Haonan Tian, Younis Ibrahim, Rui Chen, Chen Jin, Shuting Shi, Jiesi Xing, Jianjun Li, and Li Chen. Evaluation of seu impact on convolutional neural networks based on bram and cram in fpgas. *Microelectronics Reliability*, 144:114974, 2023.