Identifying Radiation-Induced Micro-SEFIs in SRAM FPGAs

Andrés Pérez-Celis, Corbin Thurlow, and Michael Wirthlin, *Senior Member, IEEE*

***Abstract*— Field-programmable gate arrays (FPGAs) are susceptible to radiation-induced effects that can affect more than one memory cell. Radiation-induced microsingle event functional interrupts (micro-SEFIs) are one of such events that can upset several bits at a time. These events need to be studied because they can overcome protection from techniques such as triple modular redundancy (TMR) and error correction codes (ECCs). Extracting these events from radiation data helps to understand if specific resources of the FPGA are more vulnerable and the extent of this vulnerability. This article presents a method based on statistics and fault injection to identify micro-SEFIs from beam-test data in the configuration memory and block RAM (BRAM) of SRAM-based FPGAs. The results show the cross section of these events for the configuration RAM (CRAM) and BRAM for three families of Xilinx SRAM FPGAs gathered throughout three neutron tests. This article also contains data from a fault injection campaign to uncover the possible CRAM source bits causing micro-SEFIs in memory look-up tables (LUTs) of Xilinx 7-series and Ultrascale devices.**

***Index Terms*— Field programmable gate arrays (FPGAs), micro single event functional interrupt (micro-SEFI), radiation testing, single event effects (SEEs).**

I. Introduction

ELECTRONIC circuits are susceptible to radiation-induced effects known as single event effects (SEEs) [1]. These events occur when a particle strikes the circuit transferring some of its energy to elements of the circuit. This energy is commonly transferred in the form of current. Depending on the location of the strike and the amount of transferred energy, SEE can have different effects in the operation of the device [2]. For example, if an SEE hits a memory cell it can change the value stored in that cell, or if an SEE happens in combinational logic it can generate a transient pulse that can potentially be latched by sequential elements. These two types of SEEs are known as single event upset (SEU) and single event transient (SET), respectively.

In SRAM field-programmable gate arrays (FPGAs), like with other SRAM-based devices, SEEs can affect more than one memory cell [3]–[5]. This could happen for one of two reasons. First, an energized particle may spread the charge over more than one memory cell, causing them to upset [6]–[8]. This event is known as a multiple-cell upset (MCU) [9]. Second, a SEE may happen in a single cell location that controls the functionality of multiple cells within the FPGA, causing more than 1 bit to upset. Examples of this include a transient event that gets latched on a reset line of multiple flip-flops or an upset in a control bit that sets or clears the content of a block RAM (BRAM). We define this event as a microsingle events functional interrupt (micro-SEFI).

Micro-SEFIs are important to study because their occurrence can overcome FPGA SEU mitigation techniques such as triple modular redundancy (TMR) and error correction codes (ECCs). Many of these mitigation techniques only protect the FPGA circuit from a single cell upset (SCU) [10]–[12], or 2-bit upsets [13]. Micro-SEFI events have been observed to impact specific resources inside of an FPGA such as look-up tables (LUT) configured as an internal memory [14]. Micro-SEFIs have also been observed to affect 2 bits in the same word of a BRAM with single error correction double error detection (SECDED). Such upsets make the data in the word unusable. Understanding micro-SEFI frequency of occurrence, shapes, and the affected resources can help provide insight and guidelines for designers to properly protect their FPGA designs from such occurrences.

Micro-SEFIs, and their effects, can be studied by examining the results from radiation test data. However, extracting micro-SEFIs from radiation test data is challenging for two main reasons. First, micro-SEFIs have a low occurrence rate. During a radiation test, most of the events will only affect one memory-cell [3]. Second, there is no information available that associates the memory cells in the configuration memory of an FPGA with the specific elements that these cells control. Such a mapping would allow the ability to identify if the affected cells are all part of the same element in the FPGA.

To address these challenges, this article proposes a method that relies on a statistical analysis of the radiation data and the emulation of faults in the FPGA configuration memory to detect micro-SEFIs. The method described in this article leverages the technique in [3] to identify MCUs and micro-SEFIs from radiation testing data. The proposed method is used to identify micro-SEFIs on Xilinx devices of three different families: 7-series (planar CMOS 28 nm), UltraScale (planar CMOS 20 nm), and UltraScale+ (FinFET 16 nm). The results estimate the cross section of these micro-SEFI events and compare them to the SCU cross section. This article also

Manuscript received April 16, 2021; revised July 15, 2021; accepted August 22, 2021. Date of publication September 10, 2021; date of current version October 18, 2021. This work was supported in part by the I/UCRC Program of the National Science Foundation under Grant 1738550 and in part by the Los Alamos Neutron Science Center (LANSCE) in December, 2018, under Grant NS-2018-8031-A.

The authors are with the NSF Center for Space, High-performance, and Resilient Computing (SHREC), Department of Electrical and Computer Engineering, Brigham Young University, Provo, UT 84602 USA (e-mail: andres.perez.celis@gmail.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNS.2021.3108572.

Digital Object Identifier 10.1109/TNS.2021.3108572

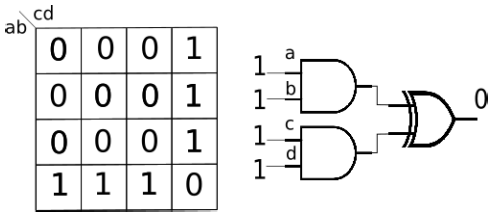

Fig. 1. Content for an four-input LUT and the resulting logic circuit.

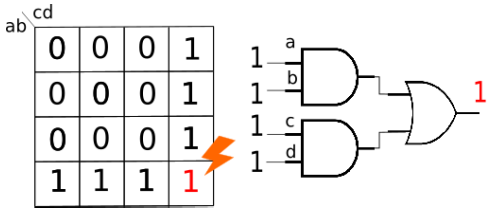

Fig. 2. Upset in a four-input LUT and the resulting logic circuit.

demonstrates that fault injection may be used to induce micro-SEFIs in a controlled environment to better understand their impact on FPGA designs.

II. SEUs in CRAM

An SRAM-based FPGA contains a large amount of configuration RAM (CRAM) that specifies the logical operations, routing, and modes of the FPGA. SEUs within the CRAM memory of a programmed FPGA may impact the proper operation of the design. Even a single CRAM upset can cause the implemented logic to behave differently than intended.

For example, Fig. 1 shows the 16 bits of a four-input LUT and the logic circuit it implements. These bits in the CRAM define the logic behavior of the LUT. Fig. 2 shows the resulting circuit after an upset has caused a change in the content of a single bit of the four-input LUT. The circuit changes the last logic gate from an XOR to an OR.

Some LUTs within an FPGA can be configured to act as memories that can be read and written. Such LUTs have additional signals to control the read and write process. For example, the 16 bits of a 4-bit input LUT can be used as a 16×1 memory. These LUTs can interconnect with each other to provide more flexibility and memory configurations in the designs. The bits controlling the configuration of these LUTs, and other FPGA elements, are also stored in the CRAM.

Within the configuration memory, certain bits are designated control bits for specific device resources [10]. These resources include LUTs, routing logic, and memory blocks (BRAM). SEUs that occur in these control bits can cause multiple upsets related to their associated resources. For example, a single bit upset in one of these bits can cause the entire contents of a memory LUT to reset or the contents of a word in memory to be cleared. In this article, these types of events are referred to as micro-SEFIs as they affect the correct functionality of a specific building block within the FPGA.

Fig. 3 illustrates the differences between SEUs, MCUs, and micro-SEFIs. This could represent either the CRAM, or BRAM of an FPGA, or SRAM memory of another

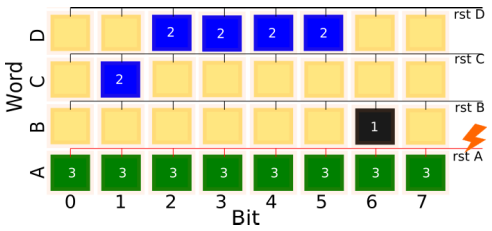

Fig. 3. Examples of the types of upsets in a memory. The black cell is an SCU, the blue cells show an MCU of size 5, and the green cells represent a micro-SEFI of size 8 that affect all the 8 bits in word A.

device [15]. For this explanation, consider the case of BRAM data in an FPGA. The black cell (1) shows an SEU that changes the content of a single cell. These events are known as SCUs. The blue cells (2) shows an event caused by a single particle that upset 5 bits. These events caused by a single particle and affect more than one memory cell are MCUs [4], [7]. Lastly, the green cells show an event that corrupted word A of the BRAM. This event, classified as a micro-SEFI, could happen due to a transient in a reset line that gets latched or an upset in a control bit that could change the memory content. The size of an MCU and a micro-SEFI refers to the number of upset bits.

III. Micro-SEFIs in SRAM FPGAs

Micro-SEFIs refer to the corresponding affected bits in the associated elements where the SCU occurs. While the resulting bits from a micro-SEFI bits are usually visible, the source SCU causing the micro-SEFI is often not visible. This is due to the lack of information on the purpose of individual CRAM bits. Micro-SEFIs caused by an SET are also not visible. As a result of this limitation and the lack of a physical to logical mapping of the CRAM, it is rarely possible to identify the source SCU for a given micro-SEFI event.

The concept of micro-SEFIs is derived from a more global SEU, commonly known as single-event functional interrupt or SEFI [16]. An FPGA SEFI often disables an entire device and requires reconfiguration or a power cycle [17]. The circuitry associated with a global SEFI could include configuration interfaces and global device control registers [10]. Several SEFIs have been observed in FPGA devices, although the cross section of these events is small.

Quinn *et al.* [10] reported on SEFIs that affect the JTAG configuration, SelectMAP configuration, scrub, and power-on reset circuitry. These events normally need a device reset to recover the device functionality. The identification of these events was performed by closely analyzing failures that happened in an FPGA. For example, it is possible to detect a scrub SEFI by counting the number of frames scrubbed in a scrub cycle. If this number is higher than a given threshold, then a scrub SEFI is detected. In contrast, micro-SEFIS are more difficult to detect.

The difference between SEFI and micro-SEFIs is that a micro-SEFI only affects a small region of the device resulting in local errors. For SRAM FPGAs, micro-SEFIs only directly affect user design logic and memory content in one region of

the FPGA while global SEFIs impact the functionality of the entire device.

Global SEFIs are relatively easy to identify as the impact of the SEFI can be observed by global device signals such as the "DONE" programming pin. Because micro-SEFIs only impact a small region of the FPGA they are more difficult to identify. All regions of the FPGA must be observed continuously for unusual behavior. Furthermore, the sensitive cross sections of these events are small requiring large amounts of radiation testing.

Bellato *et al.* [11] reported the presence of micro-SEFIs in the control logic, routing logic, and the LUTs of the FPGAs.[1] Their approach uses the information of the mapping from the bitstream to specific resources of the FPGA. With this mapping information, the authors successfully identified micro-SEFIs within routing signals.

The first step in understanding the impact of micro-SEFIs on FPGA designs is to accurately identify them in radiation test data. To this end, the proposed approach in this article describes a method to automate the identification of micro-SEFIs from raw CRAM data extracted during radiation tests.

IV. Micro-SEFI Extraction

This section describes the steps involved in our proposed method for the identification of micro-SEFIs from the raw CRAM radiation test data. This method is based on finding statistically anomalous events from this data. The first step is data acquisition from radiation tests. The second step is the identification of micro-SEFIs from data outliers using the Poisson distribution. The last step uses the remaining data to extract likely micro-SEFI events and emulates faults in the CRAM to verify if the event is a micro-SEFI.

A. Data Collection

The first step in identifying micro-SEFIs is to collect upset CRAM and BRAM data from beam experiments. During a beam experiment, the CRAM and BRAM contents are continuously read through a process called "configuration readback." This readback data are then compared to a golden copy of the CRAM and BRAM to identify upset memory cells. Performing a full device readback, comparing the data with the golden copy, and correcting the upsets comprise a "scrub cycle." During the radiation test, the scrub cycles occur continuously providing snapshots of the memory contents at discrete time intervals.

This approach assumes that all memory locations have the same probability of being upset. Each scrub cycle will have an average number of upsets that will follow a probability distribution. For radiation testing in SRAM FPGAs, the number of upsets follows a Poisson distribution [18]. The probability of an event that follows the Poisson distribution is:

$$p(x) = \frac{\lambda^x e^{-\lambda}}{x!} \tag{1}$$

[1]This work used the term SEFI; however, the reported events fit our definition of micro-SEFI and are unlike the global SEFIs described earlier.

TABLE I
Upset Information From LANSCE Neutron Testing

Scrub cycle	Upset ID	Frame	Word	Bit
13	1	0x0C8F36	5	16
13	2	0x1040693	44	23
13	3	0x11807C4	67	22
14	1	0x11409A0	10	16
15	1	0x1140067	61	25
15	2	0x1180B3C	70	15

where x is an integer of the number of events, $P(x)$ is the probability of exactly x events happening, and λ is the mean of the distribution or expected number of events.

Each scrub cycle is independently analyzed for the presence of micro-SEFIs. Upsets from a scrub cycle in a neutron radiation test are shown in Table I as an example. Each line of this table lists all the CRAM upsets that occurred in scrub cycles 13–15 of the test. Each CRAM upset in the table lists the logical location of the upset including the frame, word, and bit. In this example, scrub cycle #13 had three CRAM upsets, cycle #14 had one upset, and cycle #15 had two upsets.

B. Identifying Potential Micro-SEFIs

The second step is the identification of likely micro-SEFI events. This step analyzes all scrub cycles within the radiation test to find those cycles that have more upsets than would be expected from the Poisson distribution. Such anomalous cycles may suggest the presence of a micro-SEFI. This analysis step starts by computing a cutoff value, which we define as k, on the Poisson distribution for the number of upsets per scrub cycle. This value separates the scrub cycles of the distribution into two sets. The lower set contains all scrub cycles where the number of upsets is less than k. These scrub cycles are those that are statistically likely and represent conventional upsets. The upper set contains all scrub cycles where the number of upsets is greater than or equal to k. This upper set contains scrub cycles with a highly unlikely number of upsets and is a candidate for micro-SEFI events.

This parameter, k, is computed using a Poisson fit over the distribution of the number of upsets per scrub cycle. The λ of the Poisson fit is then used to calculate $P(x)$, the probability that x upsets occur in a scrub cycle. For this article, we chose the cutoff value of x to be where $P(x) \leq 10^{-10}$. This probably means that the chances that an event after the cutoff occurs are less than 10^{-10}. The resulting value is chosen as the cutoff, and scrub cycles are flagged accordingly.

To choose this value for other datasets, researchers can follow the same procedure and choose a probability value based on the uncertainty they are willing to tolerate. To check if the chosen value is suitable for the dataset, the cutoff value should split the distribution into two sets. The lower set must have a Poisson distribution that follows the Poisson fit. The upper set must have some events that may appear intermittently along the histogram. For example, in Fig. 4, the upper set has a Poisson distribution that contains almost all the number of upsets per scrub cycle (13 upsets per scrub

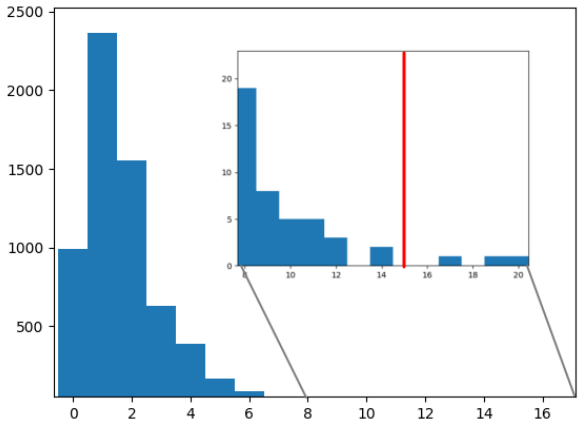

Fig. 4. Histogram showing the frequency of the number of upsets across all scrub cycles in a neutron test for the XC7A200T device.

cycle is missing). After the cutoff, events of some sizes are presented while events of other sizes are missing. We found this to be a suitable indicator that separates micro-SEFIs from other events in the distribution.

The very low cutoff probability chosen will reduce the likelihood that groups of upsets that are not micro-SEFIs will be classified as micro-SEFI. This low probability cutoff, however, may result in smaller micro-SEFI events being improperly classified as conventional MCU upsets. The next steps of the proposed method will introduce a method for identifying extracting some micro-SEFI events that have fewer upsets than the cutoff value.

Fig. 4 shows a histogram of scrub cycle upset data collected from a single neutron radiation test for a Xilinx Artix 7-series device. The x-axis represents the number of upsets detected in a single scrub cycle. The y-axis shows the number of scrub cycles containing the given number of upsets for the duration of the beam test. Fig. 4 contains a zoomed-in section focusing on the number of scrub cycles containing eight or more upsets. The red line in the figure represents the cutoff value, k, calculated with the process previously described. The cutoff value for these data is 15, meaning the remaining scrub cycles with more than 15 upsets are flagged as micro-SEFIs.

C. Verification of Micro-SEFI Events With Fault Injection

The final step is to extract micro-SEFIs from the scrub cycles that contain fewer than k events as this set of data may contain additional micro-SEFIs. To extract these events, the MCU technique described in [3] is used. Although this technique extracts both MCUs and micro-SEFIs, it does not distinguish between these two types of events.

Fault injection was used to distinguish micro-SEFIs from MCUs from the scrub cycles with fewer than k upsets. Fault injection artificially injects upsets into the CRAM memory to mimic the upset behavior seen in a radiation environment. This testing method offers a lower cost approach in testing the reliability of FPGA devices, as well as the ability to implement constrained testing parameters such as targeting specific bits or elements in the device [19]. The fault injection used in this work emulates radiation-induced upsets by changing the content of user-accessible CRAM bits of the FPGA using an external JTAG controller [20]. The two approaches to classify micro-SEFIs take advantage of the flexibility offered by this fault injection method.

The first fault injection approach selects scrub cycles that report at least one event and injects each bit reported from the scrub cycle individually. For each injected bit, the entire CRAM content is read to detect if additional bits have been upset as a consequence of this single-bit upset. If the original injected bit causes additional upsets, these bits are classified as a micro-SEFI.

The second fault injection approach targets *essential* CRAM bits within a design.[2] Essential bits are those that have some impact on the functionality of the design (most bits are not essential). The goal is to isolate the potential bits from the essential bits file that control a targeted resource. Repeating this process for different instances of the same resource can provide enough information to learn the footprint of the micro-SEFI and can also reveal the bit causing the micro-SEFI. With this information, it is possible to identify remaining micro-SEFIs that were not identified previously from the lower set of the data, i.e., scrub cycles with fewer upsets than k.

V. Radiation Testing

This section describes three neutron radiation tests that were performed to extract micro-SEFIs. A summary of each test will be given along with an overview of the raw data. All tests were performed on Xilinx devices. Except for the data gathered from the Los Alamos Neutron Science Center (LANSCE) 2018 test in the Artix-7 (XC7A200T) device containing a finite state machine (FSM) design, all other data correspond to static tests.

Three FPGAs from Xilinx were tested at the LANSCE in December 2018: Artix-7 (XC7A200T), Kintex Ultrascale (XCKU040), and an MPSOC featuring the Zynq Ultrascale+ (XCZU9EG). Fig. 5 shows the KCU105 board set up featuring the XCKU040 inside the facility. The arrow illustrates the neutron particle trajectory which is normal to the FPGA. CRAM data was read and scrubbed using the JTAG Configuration Manager (JCM) [20]. The JCM performed continuous reads and scrubs of the CRAM, logging every detected upset. On average, the JCM was capable of performing a readback and scrub cycle every 2.5 s for the Artix-7 (XC7A200T), 4 s for the Kintex Ultrascale (XCKU040), and 9 s for the Zynq Ultrascale+ (XCZU9EG). The average flux was 8.22×10^5 (n/cm^{2}s^{-1}). The expected number of upsets per scrub cycle is shown in Table II.

Table III summarizes the BRAM data for the 7-series device. The BRAMs were loaded with three different patterns divided evenly into the 365 BRAMs in the device. The patterns have BRAMs with all their bits set to 1, all their bits set to 0, and the third pattern is a checkerboard pattern where ones and zeros are alternated throughout all the BRAM content.

The second test was also conducted at LANSCE in October of 2019. The experiment tested the XC7K325T, XCKU040, and XCKU9P devices from Xilinx. All the devices

[2]These bits can be generated through optional settings using Xilinx design tools.

TABLE II

EXPECTED NUMBER OF UPSETS PER SCRUB CYCLE FOR THE THREE DIFFERENT DEVICES AT LANSCE 2018

	XCZU9EG	XCKU040	XC7A200T
Technology	FinFET16nm	CMOS 20nm	CMOS 28nm
Avg. Flux (n/cm^2s^{-1})	8.22×10^5	8.22×10^5	8.22×10^5
Fluence (n/cm^2)	1.04×10^{11}	2.34×10^{11}	1.59×10^{11}
Bits in CRAM	142,693,248	102,800,448	59,145,600
Scrub style	JTAG	JTAG	SelectMAP
Scrub cycle time (s)	9	4	0.25
CRAM Cross Section[3] (cm^2/bit)	2.67×10^{-16}	2.55×10^{-15}	6.99×10^{-15}
Expected upsets per scrub cycle	0.282	0.862	0.0850

TABLE III

SPECIFICATIONS FOR BRAM EXPERIMENT

Device	XC7A200T
BRAM Bits	13,455,360
BRAM σ (cm^2)	6.32×10^{-15}
Fluence (n/cm^2)	3.47×10^{10}
Upsets per scrub cycle	0.0175

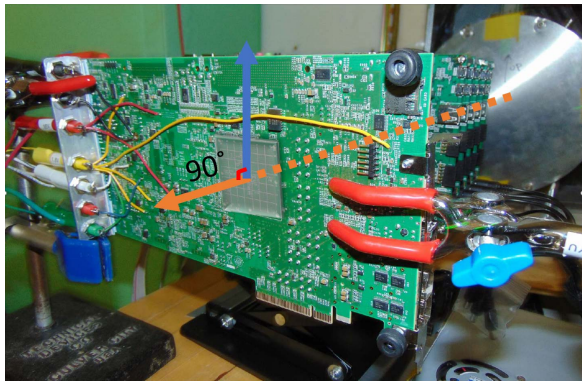

Fig. 5. XCKU040 aligned with the neutron beam.

TABLE IV

EXPECTED NUMBER OF UPSETS PER SCRUB CYCLE FOR THE THREE DIFFERENT DEVICES AT LANSCE 2019

	XCKU9P	XCKU040	XC7K325T
Technology	FinFET16nm	CMOS 20nm	CMOS 28nm
Avg. Flux (n/cm^2s^{-1})	1.11×10^6	1.11×10^6	1.11×10^6
Fluence (n/cm^2)	3.20×10^{11}	3.11×10^{11}	2.93×10^{11}
Bits in CRAM	142,657,536	102,800,448	72,823,424
Scrub style	JTAG	JTAG	JTAG
Scrub cycle time (s)	9	4	5
CRAM Cross Section (cm^2/bit)	2.67×10^{-16}	2.55×10^{-15}	5.69×10^{-15}
Expected upsets per scrub cycle	0.381	1.164	2.299

used the JCM to read and scrub the memories via JTAG. The expected number of upsets per scrub cycle and details of the experiment are shown in Table IV.

One last experiment was performed for the XCKU040 device at chip irradiation (CHIPIR) in 2018. The experiment consisted of a static design that had all the BRAMs instantiated with preconfigured content. This experiment also used the JCM to configure, read, and scrub the data via JTAG. The specifications of the expected upsets per scrub cycle and the experiment details are listed in Table V.

[3]As reported by Xilinx in the Device Reliability Report UG116 (v10.12).

TABLE V

EXPECTED NUMBER OF UPSETS PER SCRUB CYCLE FOR THE XCKU040 AT CHIPIR 2018

	XCKU040
Technology	CMOS 20nm
Avg. Flux (n/cm^2s^{-1})	5.00×10^6
Fluence (n/cm^2)	2.38×10^{11}
Bits in CRAM	102,800,448
Scrub style	JTAG
Scrub cycle time (s)	4
CRAM Cross Section (cm^2/bit)	2.55×10^{-15}
Expected upsets per scrub cycle	5.243

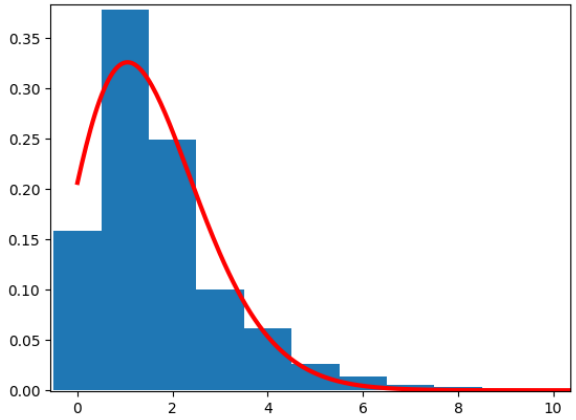

Fig. 6. Histogram of upsets in scrub cycles. The Poisson fit with $\lambda = 1.57$ is plotted in red.

VI. RESULTS

This section presents the results of the different experiments conducted during the three radiation tests. This section starts with the discussion of CRAM micro-SEFIs in each of the three experiments. It then describes one example of using synthetic designs to understand the source bit of micro-SEFIs in memory LUTs. Lastly, the section ends with results of micro-SEFIs that occur inside type 1 frames, i.e., the addresses that contain the BRAM data.

A. CRAM Micro-SEFIs

1) LANSCE 2018: Our technique identified several micro-SEFI events in the three devices tested in LANSCE 2018. The cutoff values for the XC7A200T, XCKU040, and XCZU9EG, respectively, are 15, 15, and 13. The mean λ for the Poisson fit over the scrub-cycle histogram is 1.57, 1.61, and 1.17 upsets per scrub-cycle, respectively. Fig. 6 shows the Poisson fit over the distribution of scrub cycles for the XC7A200T device.

Table VI shows the number of micro-SEFIs for different sizes identified in the CRAM using the cutoff approach. The size is the number of bits that are affected by the event. The Ultrascale+ device (XCZU9EG) had the largest number of events of size 40 or greater, while the XCKU040 experienced a higher count of small micro-SEFIs. The cross section of

TABLE VI

IDENTIFIED MICRO-SEFIs OF DIFFERENT SIZES FOR EACH DEVICE AT LANSCE 2018

Device	Size 10-19	Size 20-29	Size 30-39	Size >40
XCZU9EG	0	0	0	32
XCKU040	2	7	7	0
XC7A200T	2	1	0	1

TABLE VII

MICRO-SEFIs CROSS SECTION FOR THREE FPGA FAMILIES AT LANSCE 2018

Device	Cross section (cm^2)			
	size 10-19	size 20-29	size 30-39	size >40
XCZU9EG	$> 9.62 \times 10^{-12}$	$> 9.62 \times 10^{-12}$	$> 9.62 \times 10^{-12}$	3.08×10^{-10}
XCKU040	8.55×10^{-12}	2.99×10^{-11}	2.99×10^{-11}	$> 4.27 \times 10^{-12}$
XC7A200T	1.26×10^{-11}	6.29×10^{-12}	$> 6.29 \times 10^{-12}$	6.29×10^{-12}

these events is presented in Table VII. A single event was assumed to compute the cross section of micro-SEFIs with zero occurrences [18].

The lower set of the data for the XC7A200T of the LANSCE 2018 test is used to perform an additional step consisting of the injection of each bit of the micro-SEFIs. Interestingly, not all of the identified micro-SEFIs were repeatable through this process. About 13% where repeatable through fault injection. There are two possible explanations we have for these results. The first explanation is related to upsets in CRAM bits. The bit causing the micro-SEFIs is not readable, so it is not present on the scrub cycle upsets. Moreover, this bit may not be addressable, so even performing an exhaustive fault injection would not uncover the source bit causing the micro-SEFI. The second explanation is not related to CRAM upsets, this could be, for example, a transient event in some control circuitry, e.g., a reset line, that could modify a set of bits at once.

However, in some cases, a micro-SEFI can be directly mapped to a specific bit and thus it is straightforward to reproduce the event. This is of special interest because even though micro-SEFIs have a low cross section, mapping them to a specific resource within an FPGA provides engineers with additional knowledge to leverage the reliability of the design by limiting the use of the affected resource.

The next step to further identify micro-SEFIs in the lower set of scrub cycles is to inject each bit in the scrub cycles with the presence of a supposed MCU. The goal of the replay is to separate the MCU and micro-SEFIs apart from the LANSCE 2018 data. The results show that micro-SEFIs are rare events that are at least an order of magnitude less likely to happen than an MCU. Consider the case of the XCKU040 that presented 7628 MCUs and only had 16 micro-SEFIs.

Micro-SEFIs affecting 6 bits were observed in the XC7A200T when performing the replay for the lower set of the data. It is worth pointing out that these data correspond to a b13 design [21] replicated to utilize most of the device logic. micro-SEFIs are related to the specific design because some resources in the FPGA may not be vulnerable to this event, and the ones that are will have different cross sections. For this particular case, 120 micro-SEFIs were discovered with this

TABLE VIII

MICRO-SEFIs COUNT AND CROSS SECTION FOR THE CHIPIR TEST ON THE XCKU040 DEVICE

	Size 10-19	Size 20-29	Size 30-39	Size >40
Count	0	19	21	5
Cross Section (cm^2)	$> 4.20 \times 10^{-12}$	7.98×10^{-11}	8.82×10^{-11}	2.10×10^{-11}

TABLE IX

MICRO-SEFIs COUNT FOR EACH DEVICE DURING LANSCE 2019 EXPERIMENT

Device	Size 10-19	Size 20-29	Size 30-39	Size >40
XCKU9P	0	0	0	2
XCKU040	0	0	11	0
XC7K325T	2	1	0	1

TABLE X

MICRO-SEFIs CROSS SECTION FOR LANSCE 2019

Device	Cross section (cm^2)			
	size 10-19	size 20-29	size 30-39	size >40
XCKU9P	$> 3.13 \times 10^{-12}$	$> 3.13 \times 10^{-12}$	$> 3.13 \times 10^{-12}$	6.25×10^{-12}
XCKU040	$> 3.22 \times 10^{-12}$	$> 3.22 \times 10^{-12}$	3.54×10^{-11}	3.22×10^{-12}
XC7K325T	6.83×10^{-12}	3.41×10^{-12}	$> 3.41 \times 10^{-12}$	3.41×10^{-12}

step. These results show the importance of performing this fine-grain filtering to distinguish between MCUs and micro-SEFIs. For this particular test, the ratio of MCUs to micro-SEFIs affecting 6 bits is 2:1.

2) CHIPIR 2019: For the CHIPIR 2019 test, the cutoff value to divide the histogram detailing the number of upsets per scrub cycle is 22 for the XCKU040 device. The mean for the Poisson fit over the distribution is 3.63. The number of events and their cross sections is presented in Table VIII.

3) LANSCE 2019: For the LANSCE 2019 test, the cutoff value for the histogram with the number of upsets per scrub cycle is 14, 17, and 19 for the XCKU9P, XCKU040, and XC7K325T, respectively. The mean for the Poisson fit over the distribution in the same order is 1.43, 2.09, and 2.60. The number of events is shown in Table IX, their cross section is specified in Table X.

B. Micro-SEFI in Memory LUTs

Given that micro-SEFIs are events with a low cross section, we inject faults into a synthetic design on the XC7A200T to gather more micro-SEFI data and study their footprint. The design consists of instantiating 500 memory LUTs initialized to the desired value. With the essential bit file, we perform a targeted fault injection on several designs with the 64-bit memory content set to hexadecimal values within the range of $0x0000000000000000$ and $0xFFFFFFFFFFFFFFFF$. The injections were only single-bit injections where the content of a single bit was changed at a time. In the case where no additional upsets happened, the fault was scrubbed. In the interesting case where additional upsets were detected, the device was reconfigured and the bit causing the event was flagged as a micro-SEFI source bit. The results of this experiment identify the bit, causing the micro-SEFI in the memory LUTs. Interestingly, the size of the micro-SEFI varies depending on

TABLE XI
PATTERNS OF MEMORY LUT MICRO-SEFIS

Memory LUT Initialization	micro-SEFI Pattern	Number Affected Words
0000_2	none	none
0001_2 0010_2	$AAAA0000_{16}$	2
0011_2	$AAAA0000_{16}$	4
0100_2 1000_2	55550000_{16}	2
0111_2	$AAAA0000_{16}$ $FFFF0000_{16}$	4
1010_2	$FFFF0000_{16}$	2
1100_2	55550000_{16}	4
1101_2	55550000_{16} $FFFF0000_{16}$	4
1111_2	$FFFF0000_{16}$	4

the initialization content of the memory LUT; this suggests that the size of a micro-SEFI is design-dependent.

Table XI shows the memory LUT content and the patterns read after causing a micro-SEFI in the LUTs. The first column specifies the values in binary loaded in all nibbles of the LUT. The second column shows the read pattern after causing the micro-SEFI in the LUTs. The pattern specified in the second column repeats affecting two or four 32-bit words, with the number of affected 32-bit words specified in column 3. From these experiments, it is possible to notice that the number of bits in the micro-SEFI depends on the initial value loaded into the memory-LUT. Also, it can be noted that the effect of the micro-SEFI has a more involved effect than simply clearing all the bits. Without the physical layout for the implementation of these LUTs is hard to point out what the source bit of the micro-SEFI is exactly doing.

In addition to the size, the results provide valuable information to automate the process of detecting a memory-LUT micro-SEFI. The data suggest that an upset in a frame address with minor[4] 30 or 31 consistently produces a memory-LUT micro-SEFI in the XC7A200T device.

A similar experiment was performed on the LUT memories of the XCKU040 device. The results of this experiment showed that the patterns of the micro-SEFIs are not regular, as in the case of the Artix 7 device. We were unable to map the number of bits in the micro-SEFI with the number of set bits in the LUT. However, the results show that the size of the LUT micro-SEFIs can range from 3 to 32 bits affecting at least two words and four at most. The results also provide information on the source bits that cause the micro-SEFI. For the XCKU040, the micro-SEFI presented in minors 6, 7, and less frequently, appeared in minor 8.

A final experiment was performed using the b13 design [21]. To utilize a significant amount of resources in the device, the B13 state machine was replicated 256 times for a single design. The experiment injected each of the essential bits individually at least three times. The results showed that additionally to the micro-SEFIs in minors 6, 7, and 8, there were some micro-SEFIs in minors 12, 24, 26, and 27. These micro-SEFIs affected two words of two different frames.

[4]For Xilinx devices, the term minor refers to the five least significant bits of a frame address.

TABLE XII
CROSS SECTION FOR MICRO-SEFI EVENTS ON THE BRAMS OF THE XC7A200T

Event size	16	128	1024
Cross section (cm^2)	1.07×10^{-17}	2.14×10^{-18}	4.28×10^{-18}

TABLE XIII
MICRO-SEFIS COUNT AND CROSS SECTION FOR BRAMS AT THE CHIPIR TEST ON THE XCKU040 DEVICE

Size	128	512	1280	1410-1415	1536-1548	>4962
Count	19	1	1	5	20	2
Cross Section (cm^2)	7.98×10^{-11}	4.20×10^{-12}	4.20×10^{-12}	2.10×10^{-11}	8.40×10^{-11}	8.40×10^{-12}

Again, it is hard to point out exactly the affected resources and the exact mechanism that causes which causes the micro-SEFIs without more information on the physical layout or the relationship between the resources controlled by each bit in the CRAM. The latter has been partially addressed with project X-Ray. This project has been used to understand the relationship of each bit in the CRAM with the resource it controls in [22]. However, project X-ray only supports a single Xilinx FPGA family, thus, is not a scalable solution to understand micro-SEFIs.

C. BRAM Micro-SEFIs

For BRAM resources, the proposed technique identified three different sized micro-SEFI events for the XC7A200T: one 16-bit, five 128-bit, one 1024-bit, and one 1025-bit event. Their cross sections are shown in Table XII. We assumed that the 1025-bit event is the same as the 1024-bit event when computing the cross section. It is worth mentioning that the 128-bit events are not in the BRAM data itself but rather happened in type 1 frames. These events could affect features of the BRAM that were not exercised on these tests. For the experiments presented in this article, the BRAMs were used as single block 36Kb memories and tested with ECC enable and disable.

The results for the CHIPIR 2019 test shows similar micro-SEFI events in the XCKU040 device. These micro-SEFIs have a larger range, and the largest event affected 5311 bits of data. The count and their cross section are presented in Table XIII.

An experiment was performed with all the BRAMs instantiated on the XC7A200T to analyze further the BRAM micro-SEFIs. After injecting all the essential bits, no micro-SEFI was detected. This result is expected as not all of the CRAM bits are addressable, and the type 1 frames where BRAM data and other configuration is located are part of those inaccessible bits. However, interestingly six bits in minor 27 caused an SCU in a single type-1 frame. Although the effect of this upset is beyond the scope of this article, it would be interesting to inject those bits in designs that use BRAMs.

However, the 1024-bit events happened in the BRAM data. These events affected the same 2 bits in each of the 512 words in the BRAM. With the SECDED code that can be enabled in 7-series BRAMs [23], the 1024-bit event would

cause SECDED to report all the words as uncorrectable. Furthermore, the 1025-bit event would affect 511 words with a double error, and the remaining word would have three upsets. Given the functionality of SECDED, this 3-bit event would be treated as a single-bit error. SECDED would attempt to correct this error but instead, introduce another error into this word.

VII. Conclusion

The technique presented in this work successfully identified micro-SEFIs from radiation data on three different FPGA families: 7-series (28 nm), UltraScale (20 nm), and UltraScale+ (16 nm). The technique used statistics to compute the cutoff value to divide the Poisson distribution into two sets. The resulting upper set isolates possible micro-SEFI events. For the lower set of data, the use of fault injection provides information to distinguish between MCUs and micro-SEFIs.

Performing fault injection in the devices revealed additional information about micro-SEFIs. The experiment on the memory LUTs uncovered the specific bit causing a micro-SEFI for the XC7A200T and XCKU040 devices. It also showed an estimate of the bits that are upset with the event. Additionally, the experiment on the b13 showed that micro-SEFIs are not always reproducible for two main reasons. First, a micro-SEFI event could be triggered during a beam test when a particle upsets a bit that is not user accessible for fault injection testing. Second, injecting a specific bit may not always trigger the micro-SEFI event in the design. The inconsistency of reproducing micro-SEFI events during fault injection requires further research to understand what causes this behavior.

The overall results show that micro-SEFIs are rare events that have a small cross section. Micro-SEFIs can overcome protection techniques like SECDED, as shown by the BRAM results where 2 bits in each word got corrupted making the BRAM data unusable. This behavior drove us to generate a synthetic design to investigate the footprint of micro-SEFIs. Results from performing essential bit fault injection show that some micro-SEFIs can be mapped to a specific bit within the CRAM. To further increase our understanding of micro-SEFIs it is possible to use projects like X-Ray. This will be explored in future work.

References

[1] M. Ceschia *et al.*, "Identification and classification of single-event upsets in the configuration memory of SRAM-based FPGAs," *IEEE Trans. Nucl. Sci.*, vol. 50, no. 6, pp. 2088–2094, Dec. 2003.

[2] R. Gaillard, "Single event effects: Mechanisms and classification," in *Soft Errors in Modern Electronic Systems*. Boston, MA, USA: Springer, 2011, pp. 27–54.

[3] A. Perez-Celis and M. J. Wirthlin, "Statistical method to extract radiation-induced multiple-cell upsets in SRAM-based FPGAs," *IEEE Trans. Nucl. Sci.*, vol. 67, no. 1, pp. 50–56, Jan. 2020.

[4] G. Tsiligiannis *et al.*, "Multiple cell upset classification in commercial SRAMs," *IEEE Trans. Nucl. Sci.*, vol. 61, no. 4, pp. 1747–1754, Aug. 2014.

[5] B. Gill, M. Nicolaidis, and C. Papachristou, "Radiation induced single-word multiple-bit upsets correction in SRAM," in *Proc. 11th IEEE Int. Line Test. Symp.*, Jul. 2005, pp. 266–271.

[6] L. T. Clark and S. Shambhulingaiah, "Methodical design approaches to radiation effects analysis and mitigation in flip-flop circuits," in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI*, Jul. 2014, pp. 595–600.

[7] A. Bosser *et al.*, "Investigation on MCU clustering methodologies for cross-section estimation of RAMs," *IEEE Trans. Nucl. Sci.*, vol. 62, no. 6, pp. 2620–2626, Dec. 2015.

[8] R. Reed *et al.*, "Heavy ion and proton-induced single event multiple upset," *IEEE Trans. Nucl. Sci.*, vol. 44, no. 6, pp. 2224–2229, Dec. 1997.

[9] 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 Trans. Nucl. Sci.*, vol. 61, no. 6, pp. 3080–3087, Dec. 2014.

[10] H. Quinn, P. Graham, J. Krone, M. Caffrey, and S. Rezgui, "Radiation-induced multi-bit upsets in SRAM-based FPGAs," *IEEE Trans. Nucl. Sci.*, vol. 52, no. 6, pp. 2455–2461, Dec. 2005.

[11] M. Bellato *et al.*, "Evaluating the effects of SEUs affecting the configuration memory of an SRAM-based FPGA," in *Proc. Design, Autom. Test Eur. Conf. Exhibit.*, vol. 1, 2004, pp. 584–589.

[12] M. J. Cannon *et al.*, "Strategies for removing common mode failures from TMR designs deployed on SRAM FPGAs," *IEEE Trans. Nucl. Sci.*, vol. 66, no. 1, pp. 207–215, Jan. 2019.

[13] J. Tonfat *et al.*, "Analyzing the influence of the angles of incidence on SEU and MBU events induced by low LET heavy ions in a 28-nm SRAM-based FPGA," in *Proc. 16th Eur. Conf. Radiat. Effects Compon. Syst. (RADECS)*, 2016, pp. 407–412.

[14] M. Cannon, A. Pérez-Celis, G. Swift, R. Wong, S.-J. Wen, and M. Wirthlin, "Move the laser spot, not the DUT: Investigating the new micro-mirror capability and challenges for localizing SEE sites on large modern ICs," in *Proc. 17th Eur. Conf. Radiat. Effects Compon. Syst. (RADECS)*, Oct. 2017, pp. 126–129.

[15] C. Poivey, B. Doucin, T. Carriere, R. Marec, and P. Calvel, "Heavy ion induced gigantic multiple errors in state of the art memories," in *Proc. Eur. Space Compon. Conf. (ESCCON)*, vol. 439, 2000, p. 13.

[16] T. S. Nidhin, A. Bhattacharyya, R. P. Behera, T. Jayanthi, and K. Velusamy, "Understanding radiation effects in SRAM-based field programmable gate arrays for implementing instrumentation and control systems of nuclear power plants," *Nucl. Eng. Technol.*, vol. 49, pp. 1589–1599, Dec. 2017.

[17] R. Koga, S. H. Penzin, K. B. Crawford, and W. R. Crain, "Single event functional interrupt (SEFI) sensitivity in microcircuits," in *Proc. 4th Eur. Conf. Radiat. Effects Compon. Syst. (RADECS)*, Sep. 1997, pp. 311–318.

[18] H. Quinn, "Challenges in testing complex systems," *IEEE Trans. Nucl. Sci.*, vol. 61, no. 2, pp. 766–786, Apr. 2014.

[19] C. Thurlow, H. Rowberry, and M. Wirthlin, "TURTLE: A low-cost fault injection platform for SRAM-based FPGAs," in *Proc. Int. Conf. ReConFigurable Comput. FPGAs (ReConFig)*, Dec. 2019, pp. 238–245.

[20] A. Gruwell, P. Zabriskie, and M. Wirthlin, "High-speed programmable FPGA configuration through JTAG," in *Proc. 26th Int. Conf. Field Program. Log. Appl. (FPL)*, Aug. 2016, pp. 257–260.

[21] F. Corno, M. S. Reorda, and G. Squillero, "RT-level ITC'99 benchmarks and first ATPG results," *IEEE Des. Test. Comput.*, vol. 17, no. 3, pp. 44–53, Sep. 2000.

[22] H. Yu, H.-M. Lee, Y. Shin, and Y. Kim, "FPGA reverse engineering in vivado design suite based on X-ray project," in *Proc. Int. SoC Design Conf. (ISOCC)*, Oct. 2019, pp. 239–240.

[23] *7 Series FPGAs Memory Interface Solutions*, Xilinx, San Jose, CA, USA, 2012, p. 91.