Ultrahigh Energy Heavy Ion Test Beam on Xilinx Kintex-7 SRAM-Based FPGA

Boyang Du, Luca Sterpone, *Member, IEEE*, Sarah Azimi, *Student Member, IEEE*, David Merodio Codinachs, Véronique Ferlet-Cavrois, Cesar Boatella Polo, Rubén García Alía, Maria Kastriotou, and Pablo Fernández-Martínez

***Abstract*— In recent years, field-programmable gate array (FPGA) devices have attracted a lot of attentions due to the increasing performance they provide thanks to technology scaling, besides their high flexibility through in-field reprogramming and/or partial reconfiguration capability. However, when such devices are to be deployed in safety- and mission-critical applications such as avionic and space applications, it is mandatory to verify the reliability of the device in the target environment where radiation effect is considered as one of the major sources of faults in the system. For static random access memory (SRAM)-based FPGA devices, the SRAM cells holding the configuration data for the circuit implemented on the devices are highly susceptible against single-event upset (SEU) induced by charged particle striking the device and one single SEU in the configuration memory may corrupt the implemented circuit design causing system misbehavior. In this paper, we present the radiation test data on Xilinx Kintex-7 SRAM-based FPGA using ultrahigh energy heavy-ion test beam for the first time available to third-party radiation test in CERN.**

***Index Terms*— Configuration memory, single-event upset (SEU), static random access memory (SRAM)-based field-programmable gate array (FPGA), ultrahigh energy (UHE) heavy-ion (HI) beam.**

I. Introduction

With technology advancements in recent years, field-programmable gate array (FPGA) devices are gaining more and more attentions due to the increasing performance and high flexibility they provide. Even in safety- and mission-critical applications, FPGA presents a flexible solution to be considered to meet the ever-increasing computational demands of certain tasks such as image and signal processing. However, the same as other electronic devices, to be deployed in safety- and mission-critical applications such as avionic and space missions, the reliability of FPGA device has to be evaluated against faults and errors induced by radiation effect.

Manuscript received November 30, 2018; revised January 22, 2019 and February 12, 2019; accepted February 25, 2019. Date of publication May 7, 2019; date of current version July 16, 2019.

B. Du, L. Sterpone, and S. Azimi are with the Department of Control and Computer Engineering, Politecnico di Torino, 10129 Turin, Italy (e-mail: boyang.du@polito.it; luca.sterpone@polito.it; sarah.azimi@polito.it).

D. M. Codinachs, V. Ferlet-Cavrois, and C. B. Polo are with European Space Agency, ESTEC, 2200 AZ Noordwijk, The Netherlands (e-mail: david.merodio.codinachs@esa.int; veronique.ferlet-cavrois@esa.int; cesar.boatella.polo@esa.int).

R. G. Alía, M. Kastriotou, and P. Fernández-Martínez are with the European Organization for Nuclear Research, CERN, 1211 Geneva, Switzerland (e-mail: ruben.garcia.alia@cern.ch; maria.kastriotou@cern.ch; pablo.fernandez.martinez@cern.ch).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNS.2019.2915207

Regarding static random access memory (SRAM)-based FPGA, the SRAM cells holding the configuration data of the circuit design implemented on the device are among the devices highly susceptible against single-event upsets (SEUs) induced by radiation effect when charged particles strike the device [1]. One SEU in the configuration memory may cause the circuit design to be altered leading to system misbehaviors depending on the bit affected by such SEU. Thus, it is mandatory to evaluate the device and design sensitivity against SEU in configuration memory and apply suitable fault tolerant strategy accordingly to guarantee a successful mission [2].

On the other hand, radiation test is one of the methods that are commonly used for such evaluation as it could emulate the space environment using accelerated particles to apply to the device under test (DUT). The single event effects (SEEs) error rate cross section could then be computed by correlating the linear energy transfer (LET) of the test environment and the LET of the real application environment, such as the LET of galactic cosmic rays (GCR). However, in facilities equipped with heavy ion (HI) beam line, it is difficult to achieve the energy level of the ions in the GCR spectra, thus the LET correlation is obtained for fitting the test LET curve upon the GCR LET curve (in case of space mission application).

In this paper, we present results from the radiation test on a Xilinx Kintex7 SRAM-based FPGA device using the first-ever available ultrahigh energy (UHE) HI beam, defined as ions in the 5–150 GeV/n range, provided in CERN. The rest of this paper is organized as follows. Section II provides some background information including previous radiation test experiments carried out on different FPGA devices and radiation beam. Section III introduces the experiment setup used in the present radiation test. Section IV presents the error rate cross-sectional analysis of the test results. Conclusions are drawn in Section V.

II. Background

As mentioned before, the high sensitivity of SRAM cells to SEEs induced by radiation and the criticality of its role itself in SRAM-based FPGA makes the SEUs in configuration memory one of the major contributors of errors and misbehaviors. As shown in Fig. 1, one bitflip (SEU) in the configuration memory, depending on whether such bit is used to configure the resources in devices such as lookup table (LUT) and programmable interconnection point (PIP), may corrupt the circuit implemented on the device.

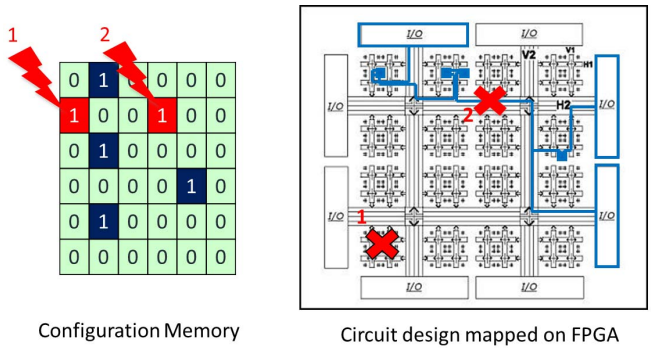

Fig. 1. SEU in configuration memory may corrupt circuit design mapped on FPGA.

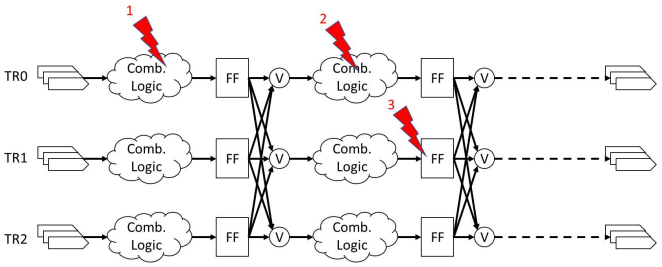

Fig. 2. SEUs in configuration memory affect different copies of logic path in XTMR implementation.

There are already several solutions for mitigating SEUs in configuration memory, such as the traditional redundancy-based solutions and various configuration memory scrubbing techniques. Triple module redundancy (TMR) as one of the most popular fault tolerant techniques has numerous implementations and variations. As for SRAM-based FPGAs manufactured by Xilinx, a Xilinx TMR (XTMR) tool [3] is available to implement TMR automatically on a design, which triplicates the logic paths and other sequential elements including flip-flops (FFs) and block memories together with automatic voter insertion, as shown in Fig. 2.

The XTMR implementation provides a fine granularity SEU mitigation in configuration memory as long as no two replicates of same logic path segment are affected by SEUs in configuration memory, the circuit design is still able to carry out normal function. For example, as shown in Fig. 2, when SEUs in configuration causing corruptions in location 1 and location 2 at the same time, the design behavior is not affected; the same goes to the situation when location 1 and location 2 are corrupted at the same time as each logic path segment can still produce correct results; however, when location 2 and location 3 are corrupted at the same time, wrong results will be generated for that segment and may be propagate through the whole circuit design causing system misbehaviors.

The second traditional solution is to refresh or rewrite the configuration memory with the correct bitstream periodically or triggered by some detection mechanism. Such techniques are called configuration memory scrubbing, whose key point is to determine when and how the rewriting is performed so that the circuit design function is not interrupted frequently while the error rate cross section could satisfy the reliability requirement posed by application constraints.

Several strategies exist already for triggering the configuration memory scrubbing. The most straightforward solution is blind scrubbing, which does not require any error detection mechanism to be implemented and it just rewrites the whole configuration memory periodically to avoid SEU accumulation, which, as demonstrated above, could be critical for solution such as XTMR. With blind scrubbing, it is critical to determine the frequency of the scrubbing, which should be calculated relying on the error rate cross section data of the device and design in the target environment [4].

More sophisticated and complicated scrubbing techniques exploit other features of the devices and circuit design itself, for example, in [5], the scrubbing is done in frame instead of the whole configuration memory by exploiting the partial reconfiguration capability of the device; in [6], scrubbing is scheduled according to the criticality of the hardware task so that a better tradeoff between the system reliability and scrubbing overhead could be achieved.

Furthermore, error detection of SEUs in configuration memory could be implemented to optimize the scrubbing rate for reducing system availability overhead. One simple error detection mechanism could be implemented by exploiting features such as the Frame error correcting code (ECC) and soft error mitigation (SEM) IP provided by Xilinx for its SRAM-based FPGA devices [7].

No matter which, the error rate cross-sectional data of DUT in the target environment is highly beneficial for better tradeoff of performance and desired system reliability against SEE induced by radiation effects. Radiation test as one of the methods for gathering the error rate cross-sectional data is quite popular among academic and commercial projects for evaluating different devices and designs to be deployed in space applications [2].

Large amount of radiation test data is already available in the literature with different types of electronic devices and different radiation beam recipes. For SRAM-based FPGA devices, for example, radiation test results have been presented in [5] using a Xilinx Virtex-5 FPGA under both neutron and proton radiation beam in different facilities, in [8] using Xilinx Kintex-7 FPGA under proton beam, in [5] using Xilinx Virtex-5 FPGA under neutron beam evaluating a frame-level redundancy scrubbing (FLR-scrubbing) technique and detection and correction hardware solutions [9]. Although heavy-ion radiation experiments on Kintex-7 have been previously presented in [8] and [10].

Although, in this paper, we present the radiation test results on using a Xilinx Kintex-7 under the first ever available UHE HI radiation beam provided by CERN.

III. Experimental Setup

A. Device and Design Under Test

During the radiation test, a Xilinx Kintex7 FPGA KC705 Evaluation Kit equipped with a Kintex7 XC7K325T SRAM-based FPGA was used as DUT. Similar to the radiation test we performed on Xilinx Virtex-5 [7], an ARM-based SoC,

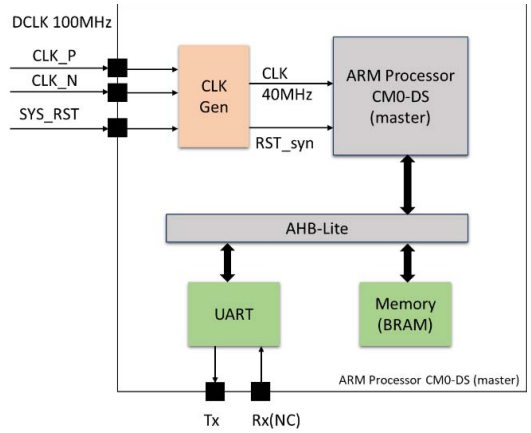

Fig. 3. Original ARM-based SoC used as benchmark circuit.

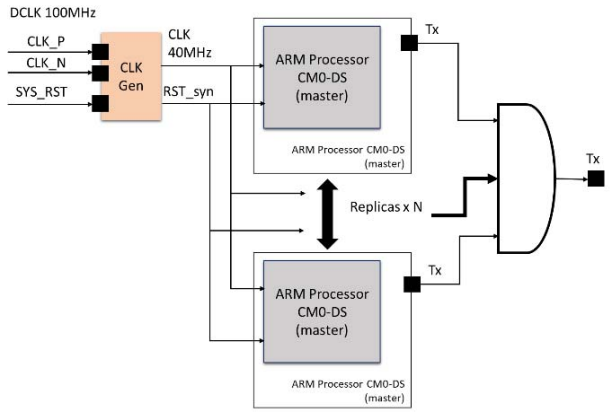

Fig. 4. Replication scheme of ARM-based SoC for increasing the device utilization.

as illustrated in Fig. 3, was used as benchmark circuit, which contains an ARM Cortex-M0 processor provided by ARM as flattened netlist through University Program, an universal asynchronous receiver-transmitter (UART) peripheral as input and output device, a block memory implementation via Xilinx block RAM (BRAM) IP for holding software code and data, and a clock generator to convert the differential clock source on board to a single end clock signal in system. The UART and BRAM components are attached to an AMBA high-performance bus (AHB)-Lite bus as same as the Cortex-M0 processor.

However, comparing to the Virtex-5 device used in previous radiation test, Kintex-7 has much larger amount of resources [11], so in order to maintain the utilization of the resource to capture as many effects as possible during the test, the ARM-based SoC was replicated multiple times as shown in Fig. 4 [12].

Furthermore, two versions of the ARM-based SoC have been prepared.

1) Plain version: the original ARM-based SoC as shown in Fig. 3, replicated for 50 copies on Kintex-7 device.
2) XTMR version: based on Plain version with XTMR applied, replicated for ten copies.

The hardware resource utilizations for both versions are reported in Table I. As could be noted that the resource

TABLE I
RESOURCE UTILIZATION FOR PLAIN AND XTMR VERSION OF ARM-BASED SoC ON KINTEX-7

Version	LUT [#]	LUT [%]	FF [#]	FF [%]	BRAM [#]	BRAM [%]
Plain_SoC*	~+3,907	~+1.91	1,189	0.29	4	0.89
Plain_x50**	195,074	95.72	59,460	14.59	200	44.94
XTMR_SoC*	~+18,760	~+9.20	9,057	2.22	12	2.70
XTMR_x10**	187,557	92.03	90,572	22.22	120	26.97

*_SoC: utilization data is for one copy in the final design
**_x50 and _x10 means the original ARM SoC is replicated 50 and 10 times for Plain and XTMR version respectively
+Utilization of LUTs for each copy in the final design has a small difference among them due to later place and route stage.

overhead induced by XTMR implementation is as high as 400% (for LUT), which is due to the extra voters inserted and more complicated routing caused by the three replicated logic paths and voter connections.

In particular, the overhead introduced, in general, by a redundancy technique is impacting not only the number of individual sequential and combinational logic resource replications but also on a given portion of extra resources, which are introduced to allow the functionality of the redundancy approach, especially when the utilization goes up, the level of optimization for place and route suffers negative impact, which could, in turn, results in even higher overhead.

Note that due to the fact that Kintex-7 as a relative new device is not supported by Xilinx integrated system environment (ISE) tool, instead Xilinx Vivado design suite has to be used. However, the XTMR tool we used is a legacy version, which can apply the TMR solution on Xilinx NGC netlist file exported by Xilinx ISE tool. Therefore, a workaround has been used for preparing the DUT: first, implement the single copy of the ARM-based SoC in Xilinx ISE tool with the target device set to Virtex-5, while keeping the BRAM component as black box; second, apply the XTMR tool on the NGC exported from ISE tool and the netlist with TMR is then exported as EDIF file; finally, in Xilinx Vivado suite, the EDIF file is imported as a component, instantiated, and replicated as shown in Fig. 4 to implement the final design. The black box in the first step of the workaround is regenerated using Vivado and the EDIF file has to be manually patched due to format change across tools.

The process of switching between tools, in particular, the legacy tool XTMR, blocks the opportunity for Vivado to optimize the internal structure of the ARM SoC as it is imported as an EDIF component. Furthermore, the XTMR process reserved all the output connections even in the final design (_x10 version) they are not connected. Last but not least, the mismatch of cell library (Virtex-5 in XTMR tool and Kintex-7 in Vivado) caused extra resources used to implement basic logic in the EDIF component. These drawbacks caused the reported resource utilization of XTMR version shown in Table I to be much higher than expected as in typical TMR implementation (particularly FF utilization), for example, the one reported in [7], which is around 3.3× overhead.

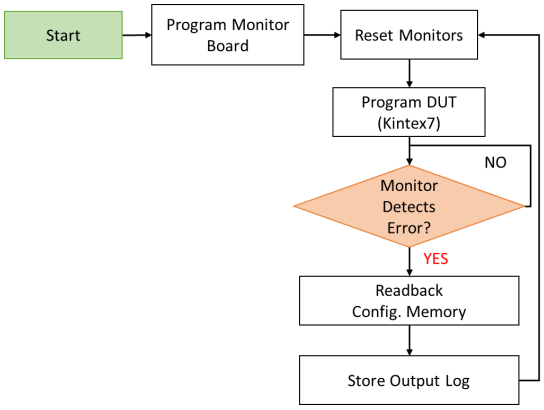

Fig. 5. Monitor flow with the Host PC application.

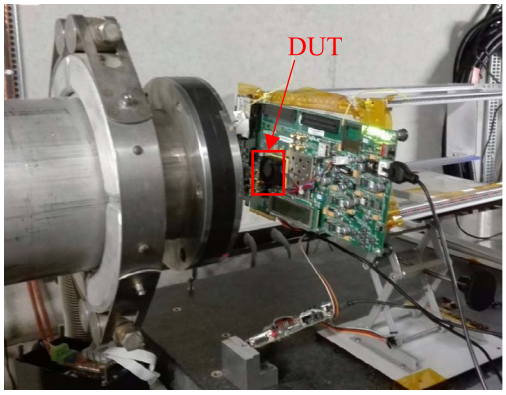

Fig. 6. Board and beam setup for alignment.

For the final design in DUT, as shown in Fig. 4, due to a limited number of pins could be used and monitored on the board during the radiation test, both the Plain and the XTMR version have their replicas' outputs ANDed together to reduce the number of pins in the final design. In this way, as long as one of the replicas generates an error in its output signal, the error will propagate to the output of the top-level design and captured by the monitor during the test.

For the program running on the ARM processor, a bubble sort application was implemented. In the bubble sort program, an array of 100 integers is generated using the pseudonumber generator, and as the seed provided is a constant, so the sequence of numbers in the array is always the same across different runs. The sorting is performed two times in ascending and descending orders and results are sent to UART as output. The application is executed in a deadloop to continuously generate output to be monitored from outside. The code is compiled without any optimization (-O0), which results in a binary code of 1.1 kB.

B. Monitor

Besides the Kintex-7 board as DUT, another board, namely, Zybo Board [14], was used to monitor the DUT outputs along with a custom designed Host PC application. When the test starts, the Host PC application will program the monitor board, i.e., Zybo board, to initialize the UART component to continuously monitor the outputs from DUT and program the DUT to start the run loop. During the run, the monitor board keeps checking the UART outputs from DUT and compares them with the golden copy captured and stored before the radiation test (fault-free output). In case of a mismatch, Host PC application will be notified by monitor board and starts the configuration memory readback procedure of DUT and after the UART log data transferred from monitor board to Host PC and stored, the UART monitors in monitor board are reset and a new run starts as illustrated in Fig. 5. During each run, the Kintex-7 DUT is reprogrammed to clear out any SEU in configuration memory accumulated in the previous run.

C. UHE Heavy-Ion Beam

The Xe HI beam was used for the radiation test with energy level set to 40 GeV/n, and the effective LET is 3.7 MeV·cm^2/mg obtained by FLUKA [15] while considering the volume around 1 μm^3 (which is compatible with SEU sensitive regions). The particles with such high energy level are capable to penetrate the device with package, and possible to generate single-event multiple upsets (SEMUs) in the configuration memory as shown in Section IV.

When UHE is used, high penetration capability eases the effort for DUT preparation since there is no need to delid or thin the DUT as shown in Fig. 6 with fan still attached. Furthermore, the physical reaction between the beam and DUT, as the secondary electrons generated when HIs pass through matter further produce ionization (delta rays), is different with the energy transfer with lower energy particles (<10 MeV/n), meaning the UHE could provide more close effects of SEE and more accurate SEE error rate evaluation than using lower energy beam when GCR spectra are considered where particle energies peak in the range of 500 MeV/n–1 GeV/n. The delta ray generated by high energy ion could deposit energy in a larger distance range around the ion trajectory comparing to the effects with ion of lower energy, SEMUs could be more probable using UHE than using lower energy beam.

Unfortunately, as we mentioned that it is the first time UHE beam in CERN is open for third-party testing, we only had the chance for testing the DUT for one LET value. However, though the LET is quite constant due to high penetration, there is still high-LET fragment production up to ~12 MeV·cm^2/mg. More tests with different LET values are being considered as further investigation of current work.

A logbook containing the beam information regarding a number of particles hitting the devices along with timestamps during the radiation test is provided so that it is possible to correlate the error detected during the test and number of particles afterward for calculating the error rate cross section presented in Section IV.

IV. Experimental Results

A. Error Rate Analysis With VERI-Place Tool

The VERsatIle (VERI)-Place [13] Tool is a tool developed in house, which is capable of: 1) performing design application error rate prediction for SRAM-based FPGA with respect to SEU in configuration memory induced by radiation effects and

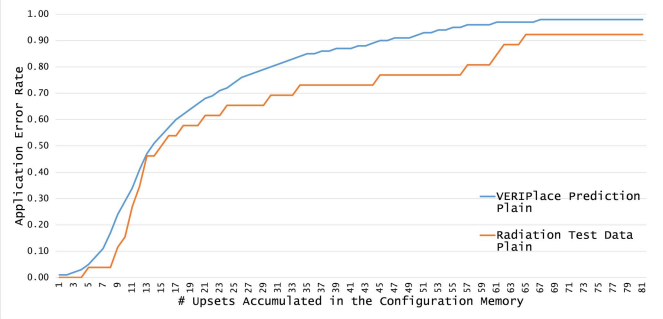

Fig. 7. VERI-Place error rate comparison with radiation test data for Plain version.

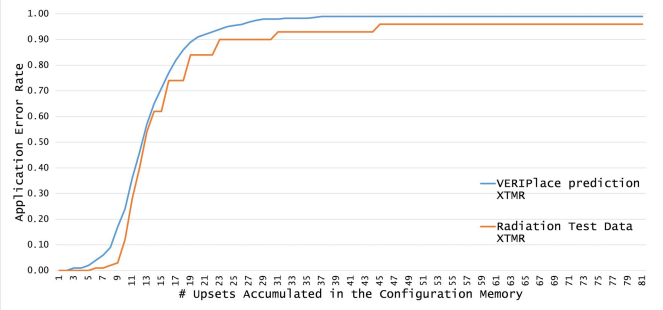

Fig. 8. VERI-Place error rate comparison with radiation test data for XTMR version.

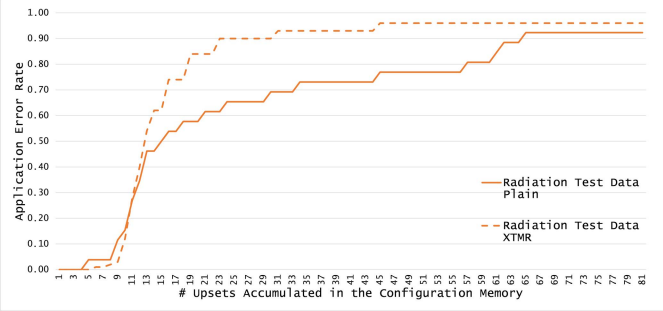

Fig. 9. Error rate comparison between the Plain and XTMR version collected during the radiation test.

2) improving design reliability without introducing hardware resource overhead. It has been used in previous radiation test for error rate prediction and the effectiveness of its hardening strategy has also been verified [7]. The application error rate is calculated as the probability of application generating an error at the output with respect to a certain number of SEU accumulated in the configuration memory. For the purpose of this paper, we used the VERI-Place tool to exclusively predict the expected application error rate with respect to the SEUs accumulated within the Kintex-7 FPGA configuration memory. The comparison between the radiation test data and VERI-Place prediction is shown in Fig. 7 for the Plain version and Fig. 8 for the XTMR version. Note that the data reported in Figs. 7–9 are related to the full device configuration memory and the application error rate is not normalized with respect to the number of design copies mapped on the FPGA.

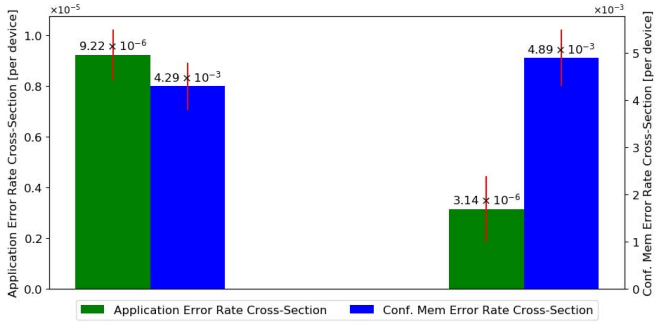

Fig. 10. Application and configuration memory error-rate cross section comparison for Plain and XTMR versions.

As can be seen from Figs. 7 and 8, the prediction made by the VERI-Place tool is accurate as the small offset is due to the factor that the high energy particles hitting the device may generate SEMUs and the beam was operating as spilling mode (periodically, a burst of particles is directed to strike the device) leading to overcounting of SEUs in configuration memory during the radiation test.

When comparing to radiation test data of Plain and XTMR versions together, as shown in Fig. 9, it can be observed that the XTMR has lower error rate when the number of SEUs accumulated in the configuration memory is relatively low, however, when the number of SEUs increases, the error rate of XTMR version goes higher than the plain version for which we believe is due to the large resource overhead in XTMR version (i.e., larger sensitive area/bits in configuration memory). On the other hand, it indicates the fact that when XTMR is complimented with other techniques such as configuration memory scrubbing to avoid SEU accumulation in configuration memory, XTMR version can achieve quite low error rate.

Furthermore, the application error rate cross section which is calculated as the probability of a particle striking the device generates an error in the output, along with the configuration memory error rate cross section which is the probability of a particle striking the device generates an SEU in the configuration memory are reported in Fig. 10. The application error rate cross section obviously depends on the circuit design, and in this case, the software running on the soft-core Cortex-M0 processor also, while the configuration memory error rate cross section is device dependent and, as could be seen in Fig. 10, similar between the Plain and XTMR version. Also, XTMR version is able to achieve the application error rate cross section 65.9% lower than the Plain version.

B. Observation of SEMU

After the radiation test, readback data files of the configuration memory were collected for each run of both Plain and XTMR version. Frame data have been extracted from the binary readback file to find the actual SEUs occurred in the configuration memory during the radiation test. As configuration memory is the main focus here, the mask file was used to remove the dynamic content, e.g., BRAM in the readback file as described in [16] and [17].

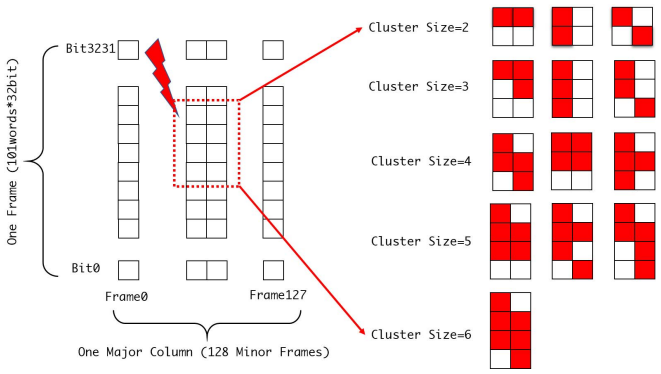

Fig. 11. Cluster (SEMU) patterns observed during the radiation test.

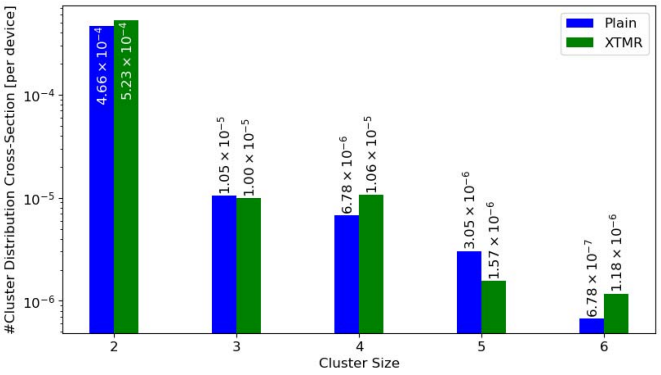

Fig. 12. Cluster distribution cross section of different cluster sizes.

By analyzing the location of SEUs in the configuration memory, several patterns have been found as multiple upsets occurred close to each other forming a cluster, which we claim to be actual SEMU occurrences. Two bits in the configuration memory are marked as close when they reside in the same major column and the distance calculated according to Formula 1 is less than $\sqrt{2}$. Similar multiple bit upsets (MBU) have been seen in reports of previous radiation tests with lower energy [9], [18]

$$\text{dist}(a,b) = \sqrt{(\text{LFA}(a)-\text{LFA}(b))^2 + (\text{BitOffset}(a)-\text{BitOffset}(b))^2} \quad (1)$$

where LFA is the linear frame address in readback data, BitOffset is the bit offset within the frame.

The patterns found by analyzing readback file are listed in Fig. 11. As can be noted that, even though the size of cluster can go up to 6 and up to 3 adjacent bits in the same frame may be corrupted at the same time, no cluster across three frames has been observed.

The distribution of the clusters of different sizes including the isolated bitflip (i.e., cluster size is 1) for both Plain and XTMR version is reported in Fig. 12. The cross section is calculated as the number of clusters of certain sizes divided by the number of particles passed through the device across all the runs for both Plain and XTMR version during the radiation test. Trivially, the cross section decreases with the size of the cluster. Table II reports the comparison between the data reported in [19], data with Si ion are included as it has similar LET with the one presented in this paper; however, the Xe ion beam has much higher LET leading to much higher bitflip cross section.

TABLE II
COMPARISON WITH TEST RESULT OF LOWER ENERGY BEAM ON KINTEX-7

	Si*	Xe*	**Xe (UHE)**
LET (MeV·cm²/mg)	4.35	49.3	**3.7**
CMem Bitflip Cross Section (#bitflip/particle)	$3.8\cdot10^{-2}$	$3.19\cdot10^{-1}$	$\mathbf{4.67\cdot10^{-3}}$
% Cluster Size = 1	90.1	64.0	**88.66**
% Cluster Size = 2	8.7	23.0	**10.85**
% Cluster Size = 3	0.6	4.1	**0.22**
% Cluster Size = 4	0.2	3	**0.20**
% Cluster Size = 5	-	-	**0.05**
% Cluster Size = 6	-	-	**0.02**

* data reported in [14]

As could be noted from the table, in our experiment, we did observed larger clusters (size of 5 and 6), which is not reported in the experiment with lower energy level HI beam. When comparing with Si beam, the distribution of larger cluster is higher with the Xe UHE beam even though LET is lower; when comparing with Xe beam of lower energy, it is important to note that the UHE beam's LET is much lower, we argue that with the same LET, UHE beam will induce higher portion of SEMUs which as for now is being planned as future work to verify. Thus, UHE beam present characeteristics regarding SEMU effects, which may not be trivial using lower energy beam for acclerated radiation test and unfold data later for GCR spectra, i.e., the result from UHE beam could be closer to the real scenario when the application under GCR radiation environment is considered.

The information regarding the occurrence of SEMU (cluster) and the cross section (probability) is critical for the accurate evaluation of system reliability against SEEs in configuration memory induced by radiation effects, especially when certain fault tolerant technique is to be applied and evaluated. For instance, the configuration memory scrubbing techniques, such as the ones reported in [20] and [21] and ECC-based techniques, such as the built-in FrameECC in Xilinx Kintex-7 FPGA devices [3] may not be sufficient alone when large cluster occurs corrupting multiple bits across the different frame.

Also, when XTMR solution is considered, it is important to mention that the SEMU poses further obstacles as if the place and route of three logic path replications did not take into account the possibility of SEMU, there is a chance two bits in the configuration memory close to each other control two of logic paths of the same segment, which means that one single particle hitting the device causes such cluster corrupt will cause an error in the circuit design.

The aforementioned VERI-Place tool is able to improve the design reliability on this regard, by acting on the place and route stage of design implementation so that the final configuration memory contains a reduced number of sensitive bits considering also SEMU. Unfortunately, as the radiation test presented in this paper is the first time, the UHE beam line available for the third party in CERN, we did not have time for testing another version with VERI-Place mitigation applied based on XTMR version as we did in previous radiation test on Xilinx Virtex-5 FPGA.

V. Conclusion

In this paper, we presented the radiation test results using Xilinx Kintex-7 SRAM-based FPGA under UHE HI beams made available in CERN. The error rate analysis and comparison with error rate prediction performed by the VERI-Place tool indicate that the XTMR version of design is able to achieve a well-reduced sensitivity against SEUs in configuration memory induced by radiation effects. The overall application error rate cross section of XTMR is 65.9% lower than the plain version.

Furthermore, SEMUs have been observed as clusters of different sizes in the configuration memory readback file, which means that further actions may need to be taken to cope with the possibility of multiple upsets in configuration memory corrupting multiple resources in the circuit design at the same time, for instance, two logic path replicas in XTMR implementation.

As for future work, the first comparison with data on Xilinx Virtex-5 FPGA from previous radiation tests shall be performed with more detailed analysis; second, new version of the design with VERI-Place mitigation applied based on XTMR version is being prepared and planned for future radiation test for verifying mitigation solution's effectiveness on Kintex-7 device as well as on Virtex-5 FPGA device and for more complete analysis regarding the effects of SEMU in configuration memory.

References

[1] S. Mitra, N. Seifert, M. Zhang, Q. Shi, and K. S. Kim, "Robust system design with built-in soft-error resilience," *Computer*, vol. 38, no. 2, pp. 43–52, Feb. 2005.

[2] L. Sterpone, M. Porrmann, and J. Hagemeyer, "A novel fault tolerant and runtime reconfigurable platform for satellite payload processing," *IEEE Trans. Comput.*, vol. 62, no. 8, pp. 1508–1525, Aug. 2013.

[3] *Xilinx XTMRTool User Guide, UG156 (v2.1)*, Xilinx Inc., San Jose, CA, USA, 2006.

[4] H. Quinn *et al.*, "Using benchmarks for radiation testing of microprocessors and FPGAs," *IEEE Trans. Nucl. Sci.*, vol. 62, no. 6, pp. 2547–2554, Dec. 2015.

[5] J. Tonfat, F. L. Kastensmidt, P. Rech, R. Reis, and H. M. Quinn, "Analyzing the effectiveness of a frame-level redundancy scrubbing technique for SRAM-based FPGAs," *IEEE Trans. Nucl. Sci.*, vol. 62, no. 6, pp. 3080–3087, Dec. 2015.

[6] R. Santos, S. Venkataraman, A. Das, and A. Kumar, "Criticality-aware scrubbing mechanism for SRAM-based FPGAs," in *Proc. 24th Int. Conf. Field Program. Logic Appl. (FPL)*, Munich, Germany, Sep. 2014, pp. 1–8.

[7] B. Du, M. Desogus, and L. Sterpone, "Analysis and mitigation of SEUs in ARM-based SoC on Xilinx Virtex-V SRAM-based FPGAs," in *Proc. 11th Conf. Ph.D. Res. Microelectron. Electron. (PRIME)*, Glasgow, U.K., Jun. 2015, pp. 236–239.

[8] R. Giordano, S. Perrella, V. Izzo, G. Milluzzo, and A. Aloisio, "Redundant-configuration scrubbing of SRAM-based FPGAs," *IEEE Trans. Nucl. Sci.*, vol. 64, no. 9, pp. 2497–2504, Sep. 2017.

[9] M. S. Reorda, L. Sterpone, and A. Ullah, "An error-detection and self-repairing method for dynamically and partially reconfigurable systems," *IEEE Trans. Comput.*, vol. 66, no. 6, pp. 1022–1033, Jun. 2017.

[10] D. S. Lee, M. Wirthlin, G. Swift, and A. C. Le, "Single-event characterization of the 28 nm Xilinx Kintex-7 field-programmable gate array under heavy ion irradiation," in *Proc. IEEE Radiat. Effects Data Workshop (REDW)*, Paris, France, Jul. 2014, pp. 1–5.

[11] M. Wirthlin, H. Takai, and A. Harding, "Soft error rate estimations of the Kintex-7 FPGA within the atlas liquid argon (LAr) Calorimeter," *J. Instrum.*, vol. 9, no. 1, Jan. 2014, Art. no. C01025.

[12] *7 Series FPGAs Data Sheet: Overview-Product Specification*, Xilinx Inc., San Jose, CA, USA, 2017.

[13] M. Desogus, L. Sterpone, and D. M. Codinachs, "Validation of a tool for estimating the effects of soft-errors on modern SRAM-based FPGAs," in *Proc. IEEE 20th Int. On-Line Test. Symp. (IOLTS)*, Platja d'Aro, Girona, Jul. 2014, pp. 111–115.

[14] Digilent. *ZYBO Reference Manual*. Accessed: 2014. [Online]. Available: https://reference.digilentinc.com/reference/programmable-logic/zybo/reference-manual

[15] A. Ferrari, P. R. Sala, A. Fasso, and J. Ranft, *FLUKA: A Multi-Particle Transport Code (Program Version 2005)*. Meyrin, Switzerland: CERN, 2005.

[16] *7 Series FPGAs Configuration User Guide-UG470*, Xilinx Inc., San Jose, CA, USA, 2018.

[17] 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.

[18] A. M. Keller, T. A. Whiting, K. B. Sawyer, and M. J. Wirthlin, "Dynamic SEU Sensitivity of Designs on Two 28-nm SRAM-Based FPGA Architectures," *IEEE Trans. Nucl. Sci.*, vol. 65, no. 1, pp. 280–287, Jan. 2018.

[19] M. Berg *et al.*, "Effectiveness of Internal versus external SEU scrubbing mitigation strategies in a Xilinx FPGA: Design, test, and analysis," *IEEE Trans. Nucl. Sci.*, vol. 55, no. 4, pp. 2259–2266, Aug. 2008.

[20] I. Herrera-Alzu and M. Lopez-Vallejo, "Design techniques for Xilinx Virtex FPGA configuration memory scrubbers," *IEEE Trans. Nucl. Sci.*, vol. 60, no. 1, pp. 376–385, Jan. 2013.

[21] Xilinx Inc. *PG036 Soft Error Mitigation Controller v4.1 Product Guide*. Accessed: 2014. [Online]. Available: https://www.xilinx.com/support/documentation/ip_documentation/sem/v4_1/pg036_sem.pdf