Single-Event Characterization of a Stratix® 10 FPGA Using Neutron Irradiation

Andrew M. Keller, *Student Member, IEEE,* Michael J. Wirthlin, *Senior Member, IEEE*

***Abstract*—FPGAs are being used in data center applications in large quantities. Single-event upsets (SEUs) occur more frequently within large-scale deployments of SRAM-based FPGAs. This work estimates the neutron cross section for SEUs in the configuration memory and memory blocks of a 14-nm FinFET Stratix 10 FPGA. SEU data was collected using a custom SEU data collection system. The developed system takes advantage of SEU mitigation features available on the device. The New York City FIT rate for SEUs is estimated to be 3.2 FIT per Mbit for configuration memory and 7.1 FIT per Mbit for memory blocks.**

I. Introduction

SRAM-based FPGAs are susceptible to memory upsets caused by terrestrial-based radiation [1]. Radiation can corrupt values in configuration memory or any other memory element including user flip-flops, memory blocks, etc. In terrestrial environments, the likelihood of radiation inducing upsets in a *single* device may be very small, (e.g., one upset every 20 to 60 years on average); but in deployments of *many* FPGAs, the occurrence of radiation-induced upsets is more prevalent.

One FPGA that is used in large-scale in terrestrial based systems is the Stratix 10. The Stratix 10 is a high-capacity, high-performance FPGA that is deployed in data center applications [2]. It is built on a 14-nm tri-state FinFET technology node [3] and contains hundreds of millions of SRAM cells. Because this FPGA is used in large quantity and is built on a newer technology node, it provides an interesting specimen for single-event characterization using neutron irradiation.

Measuring the SEU neutron cross section of a device requires that SEUs can be observed. While the Stratix 10 FPGA does not permit full-device readback of configuration memory, it does provide a mechanism for reporting SEUs as they occur. This reporting mechanism is based on internal configuration scrubbing and error correction coding (ECC). In addition to being able to report SEUs, the Stratix 10 FPGA and vendor tools provide many other features related to SEU mitigation [4].

This work uses Stratix 10 device features to measure the SEU neutron cross section of configuration memory and memory blocks, (i.e., M20K blocks). Fault injection is used prior to neutron radiation testing to validate the developed SEU data collection system. In addition to providing neutron radiation test results, this work provides a detailed description of *how* the SEU data was collected, which may prove useful in future experiments.

The FPGA under test is the Stratix 10 1SG280LU. This device contains the equivalent of 2.8 million logic elements, (i.e., 4-input lookup-table flip-flop pairs), in logic resources. It has 96 multi-gigabit transceivers (28.3 Gbps), 229 Mbits of user memory in memory blocks, nearly twelve thousand 18x19 DSP multipliers (capable of up to 10 TeraFLOPS of single-precision floating point performance), and it has almost twelve hundred general purpose I/O pins.

Single-event characterization of this device was conducted at the Los Alamos Neutron Science Center (LANSCE) from December 6th to 11th, 2018. The SEU neutron cross section of configuration memory and memory blocks were estimated. Estimates were made using data obtained from the proposed SEU data collection system. Results are compared against other devices and scaled to a large-scale system. Power consumption was monitored and no single-event latch-ups (SELs) or anomalous power events were observed.

II. Device Architecture

To understand how the SEU neutron cross section is estimated, it helps to understand the Stratix 10 device architecture and available SEU mitigation features. The objective of this work is to use the device's architecture and SEU mitigation features to measure the SEU neutron cross section. This following two sections discuss the organization of configuration memory and each of the SEU mitigation features used in the proposed SEU data collection system.

The Stratix 10 configuration memory is divided into logical sectors. The configuration of each logical sector is governed by a local sector manager (LSM). All LSMs are connected to an on-chip configuration network. The network is driven by the secure device manager (SDM) [5], which acts as the gateway to configuration memory. This organization is shown in Fig. 1. Configuration memory is addressed by sector, frame, and bit.

The only way to access configuration memory is through the SDM. Commands are sent to the SDM though JTAG or through the SDM mailbox. The mailbox conveys SDM requests and responses between the SDM and mailbox clients instanced on the FPGA fabric. JTAG is used by the vendor tools and an on-chip mailbox client is used by the proposed

Manuscript submitted August 26th, 2019 for publication in the 2019 IEEE Radiation Effects Data Workshop (REDW) Workshop Record; revised August 29th, 2019. This work was supported by the Utah Space Grant Consortium, the I/UCRC Program of the National Science Foundation under Grant No. 1738550, and by LANSCE under proposal NS-2018-7895-A.

A. M. Keller and M. J. Wirthlin are with the NSF Center for Space, High-Performance, and Resilient Computing (SHREC) in the Electrical and Computer Engineering Department of Brigham Young University, Provo, UT 84602 USA (e-mail {andrewmkeller,wirthlin}@byu.edu).

978-1-7281-3828-2/19/$31.00 ©2019 IEEE

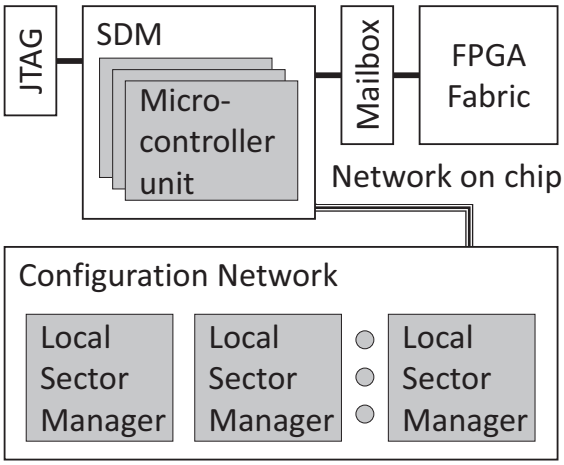

Fig. 1. Stratix 10 Device Architecture: Secure Device Manager

SEU data collection system. The SDM provides many security features and configuration options; this study uses the SDM for internal configuration memory scrubbing, SEU detection, and fault injection.

III. SEU Mitigation Features

Internal configuration memory scrubbing is a mechanism built into the device that can detect and correct SEUs. Configuration scrubbing continuously monitors for SEUs and restores correct values when corrupted values are encountered [6]. Internal configuration memory scrubbing is based on ECC. Within a frame of configuration memory, scrubbing can correct single bit upsets and detect multi-bit upsets [4]. The interval of correction can be adjusted by the end user, and the end user can enable or disable correction. If correction is disabled, then only the first SEU in a sector will be reported.

When an SEU is detected, it is reported by a local sector manager to the SDM and the SEU data is loaded into an error message queue. Internal scrubbing is potentially conducted by all LSMs in parallel. The error message queue holds up to four entries, so it must be emptied quickly so that it does not overflow during neutron radiation testing. Reading entries out of the error message queue is how SEU data is collected for configuration memory in this study. Upsets occur at a greatly accelerated rate in neutron radiation testing, so being able to empty the error message queue quickly is critical.

Fault injection is the purposeful corruption of configuration memory [7]. It is used in this study to verify the proposed SEU data collection system. To collect the SEU data necessary for this study, many features and functions need to be integrated together. Their harmony and coordination are verified through fault injection.

Fault injection in the Stratix 10 is performed using the fault injection debugger (FID). The FID is a software tool available from the vendor. It requires an additional license to use, but it is included with Quartus Prime (the vendor's FPGA development software). The FID can inject faults randomly or in specific locations. It can also read SEU data from the error message queue. Completing an operation request using the FID can take a several seconds. This tool provides the functionality necessary to validate the proposed approach for collecting SEU data.

SEU data can also be obtained using the Advanced SEU Detection IP Core [4]. This IP core communicates with SDM to send commands and receive responses concerning SEUs. It provides the same functionality as the FID for obtaining SEU data, but it is able to respond much more quickly. This IP core makes it possible to response to SEUs in real time. The core sends a notice when an SEU is detected, reports the specific bits affected, and it can lookup sensitivity information.

Additional sensitivity information can be obtained from what is called a sensitivity mapping header (SMH) file. An SMH file contains information that classifies all configuration bits within an FPGA design as potentially used or not. It designates potentially used bits as belonging to a specific hierarchical partition of the design, which comes from the user tagging design partitions with a specific advanced SEU detection (ASD) tag. The SMH file also indicates which configuration bits are addressable but physically non-existent or otherwise excluded from SEU observation, which is important for determining the total number of configuration bits included in measuring the SEU neutron cross section.

In addition to estimating the SEU neutron cross section of configuration memory this study also estimates the SEU neutron cross section of user memory blocks. The user memory block studied is the M20K. Each M20K instance provides 20 Kbits of memory and consists of 512 words that are 40 bits wide. Eight bits of every word can be used to protect the memory from SEUs through ECC. When using this feature, single-bit, double-adjacent, and triple-adjacent bit upsets can be detected and corrected [4]. The estimate made in this paper is made *without* the use of ECC, which is disabled by default.

IV. SEU Data Collection System

The SEU mitigation features available for the Stratix 10 FPGA need organized into a system that can observe SEUs during an accelerated radiation test. During such a neutron test, SEU occur at a much accelerated rate compared to terrestrial environments. As such, the test will likely need to run long periods of time without user intervention. The developed SEU data collection system needs to collect SEU data quickly and run unattended. Fig. 2 shows the main components of the SEU data collection system. The Stratix 10 SEU mitigation features and other features are used within this system to collect SEU data.

The system consists of: the FPGA development board; a network power switch for remote power cycling; a power management bus (PMBus) monitor for recording temperature, voltage, and current; serial communication links for interfacing with the FPGA; and a host computer that orchestrates the test flow. Fig. 3 provides a simplified view of the connectivity between components in the system. The network switch and host computer are connected via Ethernet over a local area network, and the host computer is connected to the PMBus monitor and JTAG connection via USB.

On top of the connected system, a hardware and software stack is added to perform the actual collection of SEU data.

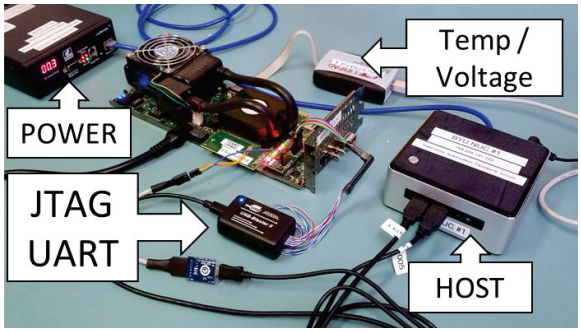

Fig. 2. SEU Data Collection System Setup

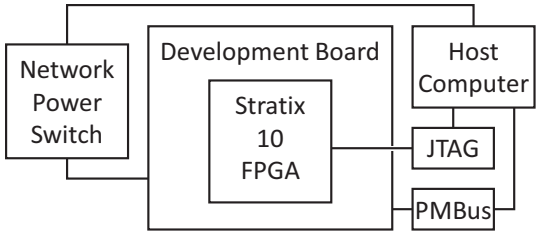

Fig. 3. SEU Data Collection System Connectivity

Fig. 4 lays out the hardware design that was built for this test. Fig. 5 lays out the software stack that was built for this test. The software stack consists of a series of Python modules and some simple TCL scripts for communicating with the FPGA though System Console (a useful debug tool available with Quartus Prime). The hardware stack consists of a JTAG-to-bus bridge IP core, a memory mapped SDM mailbox client IP core, and a memory block array made up of 1024 M20K memory blocks. All SEU data was collected through this hardware and software setup.

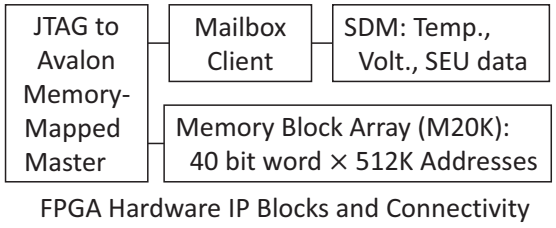

Fig. 4. Internal Components of the Design Being Tested

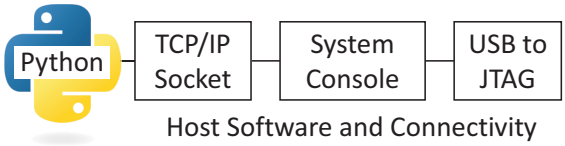

Fig. 5. Software Setup for Collecting SEU Data

Custom Python scripts were used to orchestrate all SEU data collection tasks. Temperature, voltage, and current data collection on the PMBus was performed in the background using 3rd party software, but all other tasks were performed using Python scripts. A standard stream wrapper was written in Python to automate and integrate the use of the FID for validating the system through fault injection. Data requests to the FPGA were all made through Python calls that sent commands and received responses through a TCP/IP socket server established in System Console. This approach greatly simplified test development and execution.

Custom hardware could be developed to collect SEU data from the Advanced SEU Detection IP Core and identify SEUs in an array of memory blocks, but this study took a more simplistic approach. In this study, the hardware on the FPGA consists of only: the JTAG-to-Avalon Memory Mapped Master IP Core, the SDM Mailbox Client IP Core, and a simple wrapper around an Embedded Memory IP Core instance that contains 1024 M20K memory blocks. The mailbox client and memory block array are memory mapped to an Avalon Memory Mapped Bus, (i.e., a vendor provided bus protocol), and all data transactions are handled by supporting functions in the System Console.

V. Test Flows

With the system in place, two different test flows were implemented: one for detecting upsets in configuration memory, another for detecting upsets in memory blocks. The flows are designed to be robust enough to support continuous testing in neutron testing without operator intervention. The configuration memory test flow is verified through fault injection prior to radiation testing, and proper readback of configuration memory was also verified outside of the neutron beam to validate the SEU data collection system for both configuration memory and memory blocks. Fig. 6 diagrams the two separate test flows; it highlights commonality and difference.

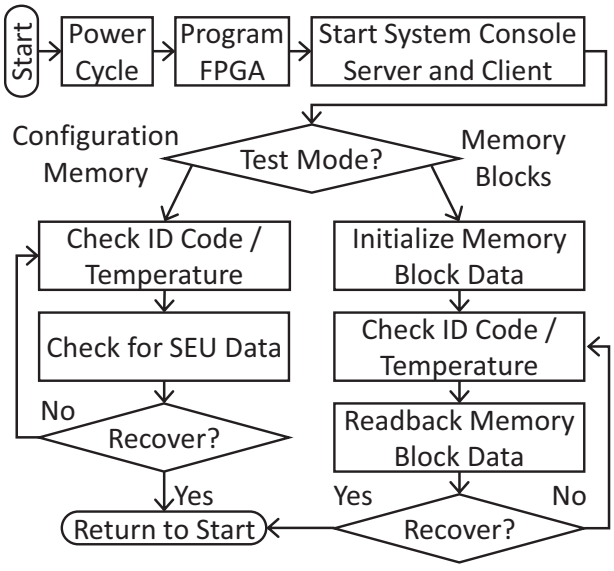

Fig. 6. SEU Data Collection Test Flow

Both flows begin in the same way. First, the FPGA development board is power cycled, (i.e., powered off for 10 seconds and then turned back on). This clears the FPGA of any upsets and begins the test in a clean state. Second,

the FPGA is programmed. This loads a clean copy of the supporting hardware onto the FPGA. It takes approximately 7 seconds to program the device plus an additional 43 seconds of software overhead, (i.e., 50 seconds total to program the device). The design can become corrupted by accelerated neutron radiation during this time, which is taken into account. Next, a TCP/IP socket server in System Console is started and an accompanying client connection is opened. With the FPGA programmed and the System Console client on-line, the system is now ready to for use by either flow.

The configuration memory test mode follows System Console initialization with an ID code check and a temperature reading. The successful completion of these tasks acts as an indicator that the system is still responsive, (i.e., a heartbeat). If these tasks cannot be completed successfully, the test flow is restarted. The configuration memory test follows this step with a check for SEU data.

For system verification purposes, faults can be injected into configuration memory prior to checking for SEU data. Faults are injected using the FID, which is a separate process maintained and controlled through a custom Python wrapper. The FID has an interactive command-line mode that makes this mode of operation possible [4]. Injected faults are detected by the development system and their locations are identified in the SEU data returned from the SDM. During neutron radiation testing, fault injection is not performed and the FID is not invoked; but system verification through fault injection proved to be a valuable preparation for neutron radiation.

Checking for SEU data from configuration scrubbing polls for entries in the error message queue by sending a command to the SDM and receiving the response. The command for this action is include in the Advanced SEU Detection IP Core. Most of the time, the request for SEU data behaved as expected, but two unexpected responses were observed in both fault injection and neutron testing. First, sometimes repeated requests for SEU data would return with no SEU data even though SEU data was expected (because a fault was injected, or the device was exposed to accelerated neutron radiation). Second, sometimes too many upsets were seen in too short a window of time, (e.g., thirty or more in a ten second period). The cause of this behavior is not clear. These behaviors were identified and filtered out of the neutron radiation test results. In this step, a check for SEU data is made back-to-back (approximately 23 ms apart) for one full second.

Checking for SEU data is followed by a recovery step. The recovery step is a failsafe that is present in both flows. If anything goes wrong in either flow, the recovery step allows the system to return to a working state. Things that will trigger a return to start include unresponsiveness to System Console requests, not receiving SEU data within a certain time period, (e.g., two minutes), receiving too much SEU data within a certain time period, or any other behavior that prevents the system from continuing in its flow. If no error is encountered, then the flow continues back to the check ID and temperature step to collect more SEU data.

The memory blocks test flow is very similar to the configuration memory test flow but it looks for SEUs in memory blocks instead of in configuration memory. This test mode follows System Console initialization with an initialization of memory block values. There are 524,288 40-bit words in the memory mapped memory block array, (i.e., 1024 M20K instances or 20 Mbits of memory block memory). A pattern of all-ones, all-zeros, one-zeros, and zero-ones are written to every four words. Any changes from the initial all-zero values prior to initialization are recorded. Initialization takes about 6.5 minutes. It is followed by check ID and temperature, which is then followed by a full memory block array readback. All detected changes are recorded and an off-chip golden copy is updated with the current values. Unless recovery is needed, this test flow then loops back to check ID and temperature and continues to collect memory block SEU data by observing upsets in the memory block array.

With the test flows working correctly, the system is ready for neutron radiation testing. The only things that change between the desktop setup in Fig. 2 and the neutron test setup in Fig. 7 are the placement of FPGA development board and the lack of fault injection. At this point, the system is fully developed and tested. It uses SEU mitigation features and other features available in the device and vendor software to support the collection of SEU data. With all of the preparation and validation, the system is ready for neutron radiation testing.

VI. Neutron Radiation Testing

Neutron radiation tests of the Stratix 10 SEU memory cross sections were conducted at the Los Alamos Neutron Science Center (LANSCE) in December of 2018. For this experiment, the Stratix 10 FPGA was aligned perpendicular to the neutron beam path such that the two inch collimated neutron beam would pass directly through the FPGA. Several other boards were placed between this experiment and the beam aperture. Degradation of neutron fluence based on the distance of the device from the source was taken into account. In preparation for the test, the liquid cooling unit on the device was removed and an external fan was put in place to keep the device cool. Fig. 7 shows the setup of evaluation board in the beam path.

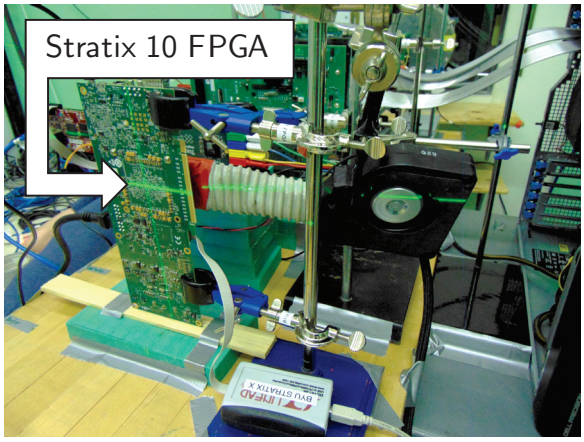

Fig. 7. Stratix 10 Neutron Radiation Test

Throughout the test, the temperature and power consumption of the FPGA were observed. Temperature data was collected from the SDM. During the neutron test, the FPGA's

temperature ranged from thirty-eight to seventy-six degrees C with an average temperature of forty-eight degrees C. Voltage and current data were collected over the PMBus using software from the power regulator vendor. No anomalies were observed in device voltage or current draw. No single-event latch-ups were observed.

During this test a detailed log was recorded of the beam fluence over time. The total neutron fluence was recorded once per second using a separate micro-controller that observed the dosimetry signal provided by the test facility. Timestamps of this log were aligned with timestamps in the SEU data collection log. Aligning the timestamps of the two logs made it possible to exclude fluence that occurred during periods of time when valid SEU data was not being collected, (e.g., during a power cycle, periods of no SEU data, or excess SEU data as seen in fault injection). Results presented in Table I used the beam log and SEU data collection log to filter the total amount of fluence exposure.

VII. Results

Results from the two neutron radiation tests are presented in Table I. A column is dedicated to each test. Total bits indicates the number of bits included in SEU observation. Fluence is the amount of high-energy neutron exposure, (i.e., greater than 10 MeV), in neutrons per cm^2 of area. Upsets are the number of observed upsets. Cross section per bit is,

$$\sigma_{\text{bit}} = \frac{\text{Upsets}}{\text{Fluence} \times \text{Total Bits}}. \quad (1)$$

The 95% confidence interval is calculated using a standard method for each cross section measurement based on the upset count and total fluence [8]. Finally, the SEU neutron cross section per bit is converted to failures in time (FIT, or failures per billion hours of operation) per Mbit. The FIT rate is based on a 13 n cm^{-2} h^{-1} high-energy neutron flux found in New York City (NYC) at sea level. This metric is included for convenience [9] and will be used to estimate the soft error rate of a hypothetical large-scale FPGA cloud computing platform.

TABLE I
Neutron Radiation Test Results for Stratix 10

Bit Type	Configuration Memory	Memory Blocks (1024 M20Ks)
Fluence (n/cm^2)	1.86×10^{11}	7.50×10^{10}
Upsets	28,681	860
Cross Section / Bit (cm^2)	$2.45\times10^{-16}\pm1\%$	$5.47\times10^{-16}\pm7\%$
FIT / Mbit	3.2	7.1

Comparing these finding against SEU neutron cross section measurements of different devices is insightful. Table II contains the SEU neutron cross section measurements of the configuration memory and memory blocks in three different FPGA devices: Intel's Stratix V (28-nm CMOS) [10] , Xilinx's Kintex 7 (28-nm CMOS) [11], and Xilinx's UltraScale+ (16-nm FinFET) [11]. Here the SEU neutron cross section of the Stratix 10 configuration memory is estimated to be 20× smaller than its predecessor, the Stratix V.

TABLE II
Cross Section Comparison with Other Devices
(cm^2 per bit, and comparison ratio)

Device	Configuration Memory		Memory Blocks	
Stratix V (28-nm) [10]	4.84×10^{-15}	20×	–	
Kintex 7 (28-nm) [11]	5.69×10^{-15}	23×	5.57×10^{-15}	10×
UltraScale+ (16-nm) [11]	2.67×10^{-15}	1.1×	9.82×10^{-16}	1.8×

A previous study examined the impact of SEUs on several data-center-like applications running on a hypothetical large-scale FPGA cloud computing platform [12]. Since the Stratix 10 FPGA is used in large-scale data center applications [2], it is helpful to scale the results of this study to reflect the SEU rate of a hypothetical large-scale system. The hypothetical system selected consists of 100,000 FPGAs deployed in Denver, Colorado. In Denver, the high-energy neutron flux is 3.76× higher on average than the reference NYC flux [9]. This system would experience one upset in configuration memory every 1.4 hours on average approximately. Only a fraction of these upsets would actually affect an active design operating on the device.

VIII. Conclusion

SEU neutron cross section data was collected on a Stratix 10 FPGA using the device's SEU mitigation features. The SEU data collection system and data analysis approach were presented. No single event latch-ups or other anomalous power events were observed. The neutron SEU cross section of a single bit in configuration memory is estimated to be 2.45×10^{-16} cm^2, which is approximately 20× smaller than the Stratix V. The neutron SEU cross section of a single bit in block memories (i.e., M20K) is estimated to be 5.47×10^{-16} cm^2.

References

[1] H. Quinn and P. Graham, "Terrestrial-based radiation upsets: a cautionary tale," in *13th Annual IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM'05)*, 2005, pp. 193–202.

[2] J. Fowers *et al.*, "A configurable cloud-scale DNN processor for real-time AI," in *Proceedings of the 45th Annual International Symposium on Computer Architecture*, ser. ISCA '18. Piscataway, NJ, USA: IEEE Press, 2018, pp. 1–14.

[3] *Intel Stratix 10 GX/SX Device Overview*, Intel Corp., February 2019. [Online]. Available: https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/hb/stratix-10/s10-overview.pdf

[4] *Intel Stratix 10 SEU Mitigation User Guide*, Intel Corp., October 2018. [Online]. Available: https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/hb/stratix-10/ug-s10-seu.pdf

[5] T. Lu *et al.*, "Secure device manager for Intel Stratix 10 devices provides FPGA and SoC security," Intel Corp. [Online]. Available: https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/wp/wp-01252-secure-device-manager-for-fpga-soc-security.pdf

[6] A. Stoddard *et al.*, "A hybrid approach to FPGA configuration scrubbing," *IEEE Trans. on Nuclear Science*, vol. 64, no. 1, pp. 497 – 503, 2016.

[7] H. Quinn *et al.*, "Fault simulation and emulation tools to augment radiation-hardness assurance testing," *IEEE Trans. on Nuclear Science*, vol. 60, no. 3, pp. 2119–2142, 2013.

[8] H. Quinn, "Challenges in testing complex systems," *IEEE Trans. on Nuclear Science*, vol. 61, no. 2, pp. 766–786, April 2014.

[9] J. JEDEC, "Measurement and Reporting of Alpha Particles and Terrestrial Cosmic Ray-Induced Soft Errors in Semiconductor Devices: JESD89A," pp. 1–85, 2006. [Online]. Available: https://www.jedec.org/sites/default/files/docs/JESD89A.pdf
[10] A. Keller *et al.*, "Dynamic SEU Sensitivity of Designs on Two 28-nm SRAM-Based FPGA Architectures," *IEEE Trans. on Nuclear Science*, vol. 65, no. 1, pp. 280–287, 2018.
[11] "Device reliability report, second half 2018," Xilinx. [Online]. Available: https://www.xilinx.com/support/documentation/user_guides/ug116.pdf
[12] A. Keller and M. Wirthlin, "Impact of soft errors on large-scale FPGA cloud computing," in *Proceedings of the 2019 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, ser. FPGA '19. New York, NY, USA: ACM, 2019, pp. 272–281.