Neutron Radiation Testing of a TMR VexRiscv Soft Processor on SRAM-Based FPGAs

Andrew E. Wilson E>, Sam Larsen, Christopher Wilson, Corbin Thurlow, and Michael Wirthlin E>, *Senior Membe,; IEEE*

Abstract-Soft processors are often used within field-programmable gate array **(FPGA)** designs in radiation hazardous environments. These systems are susceptible to single-event upsets (SEUs) that can corrupt both the hardware configuration and software implementation. Mitigation of these SEUs can be accomplished by applying triple modular redundancy (TMR) techniques to the processor. This article presents fault injection and neutron radiation results of a Linux-capable TMR VexRiscv processor. The TMR processor achieved a IOx improvement in SEU-induced mean fluence to failure with a cost of 4 x resource utilization. To further understand the TiVffi system failures, additional post-radiation**fault injection was performed with targets generated from the radiation data. This analysis showed that not all the failures were due to single-bit upsets, but potentially caused by multibit upsets, nontriplicated IO, and unmonitored nonconfiguration RAM (CRAM) SEUs.**

***Index* Terms-Fault injection, fault tolerance, field-programmable gate array (FPGA), radiation hardening by design, radiation testing, redundancy, RISC-V, single-event upset (SEU), soft processor, triple modular redundancy (TMR).**

I. INTRODUCTION

SRAM-BASED field-programmable gate arrays (FPGAs) often include soft processor implementations within their digital designs. These soft processors are implemented using the FPGA's reprogrammable resources such as lookup tables (LUTs), flip-flops (FFs), digital signal processing (DSP) units, and block RAM (BRAM). The configurable soft processors provide a softw are platform coupled with the custom FPGA solution. Using a processor such as the VexRiscv, an open source RISC-V processor using an open instruction set architecture (ISA), allows for the integration of established software tools and libraries [1]. Implementing soft processors in FPGAs can be beneficial for applicati ons in both terrestrial and space envi ronments.

To provide a sufficientl y reliable sys tem, the use of mitigation may be required to improve the functional reliability of the digital design [2] . S ingle-point failure s can be masked with triple modular redundancy (TMR) [3]. This effective mitigation technique implements redundant logic and triplicated voters to mask error s that would cause functional errors within the syste m. The improvement in reliability provided by **TMR** comes at a cost of greater power consumption , higher resource utiliz ation, and slower maximum operational frequency.

Radiation found in space and terrestrial environments can prove hazardous to static random-access memory (SRAM)-based FPGAs and the soft processors implemented within. Energized particles can cause single-event upsets (SEUs) that flip bits in configuration RAM (CRAM) and BRAM [4]. SEUs can cause functional failures in the design within the FPGA by corrupting the state and circuit configu ration. SEUs can produce unpredictable and unwanted results that may lead to a critical failure of the system . TMR can mask the single-point failures cau sed by SEUs and improve the reliability of the system.

This article inve s tigates the reliability of the VexRiscv processor system and the improvement in reliability achieved through TMR mitigation. Previous work has tested TMR mitigation technique s for hare metal app lications running on the Taiga RISC-V processor [5]. [6]. This article utilizes an existing open source Linux-capable RISC-V system on chip (SoC) implementing the VexRiscv processor. To suppo rt the Linux operating system, additional circuitry was added to the memory management unit (MMU) and the double data rate (DDR) contro ller. The results of this investigation characterize the potential effecti venes s of this system in radiation hazardous envi ronments.

The presented test data consist of radiation testing and fault injection results for the RISC -V soft-c ore running Linux application s. To better understand the differences of this article's results compared to previous work , additional post-radiation fault injection was performed using the CRAM up sets observed at the radiation test. This targeted fault injec- tion provided a better analysis of the sys tem failures observed during the neutron radiation test and identified failures not ma sked by **TMR.** The se potential failure modes included multibit upsets. nontriplic ated IO devices, and un observed faults.

The remainder of thi s article is organized as follows. Secti on II presents background information on fault-tolerant **RISC- V** soft processors. Sect ion III describes th e design under test (DUT) used in the experimen ts. The initi al fault injection test and result s are detailed in Section IV . Th e following Sec tion V state s the setup of the neut ron radiati on test and its result s. Section VI expound s on the analysis from the

Manusc ri pt recei ved Janu ary 15 , 202 l ; revise d February 25. 202 l. March 9. 2021 , and Marc h 1 5, 2021 ; ac c ept e d Mar ch I 6 , 20 21. Date of publ ic a t io n Mar ch 24 , 20 21 ; date of cu rre nt ve rs io n May 20, 202 1 . T his work was su pporte d by th e Na tio nal Sc ie nce Found a ti o n th rough the Indust ry- Unive rs it y Coo perat ive Researc h Ce nters (1/UC RC) Program under Grant I 7385 50 .

T he a uth o rs are with t he NSF Cente r for Space. High- perfo1111a nce. and Res i l ie nt Co mput in g (S HR EC) , Brig ham Youn g Univer s it y, Provo, UT 84602 USA (e- mai l: a ndr ew.e.w il so n @byu.ed u: sa mla rs @by u.ed u; c hr is to phe r.w ilson@by u.edu ; corbin.thurl ow@ byu.edu; wi rth lin @by u.ed u).

Co lo r vers io ns of one or more fig u res i n t hi s articl e are avail able at https://do i.o rg/ I O. I I ll9ff NS.2 02 l.306X835.

Digit a l Object Ide nt ifie r IO. I I0 9 ff NS. 202 l.306R835

additional fault injection results using targets generated from the radiation data. Section VIII concludes the article.

II. FAULT-TOLERANT RISC-V SOFT PROCESSORS

RISC-V is an open ISA that is used in academia, research, and industry. There are many open source **RISC- V** processors available for implementation within FPGAs with the support of existing software tool chains and libraries . The inclusion of a processor adds a software-defined subsystem to the digital design. The available RISC-V processors range widely in features. performance, and utilization of FPGA resources. Some of these processors are optimized for SRAM-based FPGA implementations, while others are not.

One study has shown the difference in performance and maximum frequency achieved within a selection of RISC-V soft processors [7]. This study's results showed the Taiga and VexRiscv among the best performing processors for SRAM-based FPGAs. Space applications have tight constraints that require the best performance for the FPGA resources utilized and power consumed .

Previous work [5], [6] performed fault injection and radiation testing on the Taiga RISC-V Processor. Taiga, a 32-bit RISC-V processor, was chosen for its optimized performance for Intel and Xilinx SRAM-based FPGAs [8]. The pipelined processor implements multiple independent execution units allowing for variable execution latencies. The Taiga processor design used approximately 33% fewer slices while clocking 39% faster than a LEON3-based system built on a Xilinx Zynq X7CZ020 [9].

This work expands the study by targeting the VexRiscv RISC-V soft processor [l]. The VexRiscv is a pipelined 32-bit processor developed with the high-level language SpinalHDL. This processor was chosen for its support of a Buildroot Linux image and performance of 1.21 Dhrystone million of instructions per second (DMIPS)/MHz and 2.27 Coremark/MHz. The processor also takes advantage of a large ecosystem of open source IP for the quick integration of SoCs within FPGA digital designs [IO].

A. TMR RISC-V

FPGA-implemented RISC-V soft-cores can mitigate against SEUs by using TMR techniques to provide redundancy to the design. The TMR soft processor includes three redundant domains and triplicated voters capable of masking a failure of a single redundant domain (see Fig. l). The three redundant domains process the same input stimulus into equal outputs during correct operation . When one domain out of the three fails, the output does not match the other two domains. The triplicated majority voters mask this erroneous output and use the majority output of the other two domains [11].

The Brigham Young University (BYU) and Los Alamos National Laboratory (LANL) BL-TMR tool provides an automatic process of triplicating the design and inserting triplicated voters [12J. The tool has achieved up to lOOx improvement in mean time to failure for some nonprocessor designs . It performs fine-grained TMR on the FPGA primitives at the netlist level by triplicating all FFs, LUTs, BRAMs, and DSPs, and

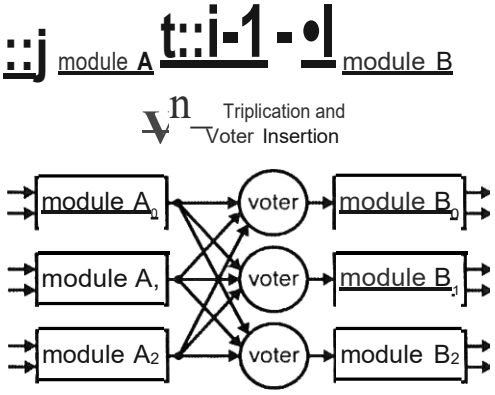

Fig. I. TMR with triplicated voters.

inserting triplicated voters between these primitives. The tool's input is a vendor-independent electronic design interchange format (EDIF) file that can be exported from Xilinx Vivado. The finished TMR EDIF file can be imported into Xilinx Vivado as a post-synthesis file and be used to produce a placed and routed bitstream.

B. Related Works

Many different soft-core processors have been targeted for space application. Several studies have modified the LEON2 and LEON3 processors for improved reliability with different mitigation schemes including TMR [12]-[16]. Other work has applied TMR to the Picoblaze [17], [18J, a free 8-bit soft processor provided by Xilinx. Xilinx also offers a Microb!aze TMR subsystem for use within their FPGAs [19]. This subsystem includes the TMR Microblaze with a soft error mitigation (SEM) core to perform single en-or correction (SEC) on the configuration memory.

The RISC-V ISA is relatively new compared to these other systems, and few works have characterized the fault tolerance of the available processors. The main target for these studies has been the Rocket Chip, the officially supported by RISC-V foundation and fully featured hardware description language (HDL) implementation f20J. One study characterized this processor against SEUs using fault injection with Xilinx 's SEM IP [21]. It was compared to other processors from previous works, showing that the Rocket Chip was more sensitive than other soft processors. This article targets more FPGA-optimized soft processors that run at higher frequencies and require fewer FPGA resources.

Other works have applied TMR to the Rocket Chip processor and validated the mitigation technique. One work used the Mentor Precision Hi-Rel tool to apply fine-grain TMR to the Rocket Chip and performed fault injection with the SEM IP [22]. Their study achieved a substantial reduction in sensitive bits of up to 11.5 x. Another work used Cadence's electronic design automation (EDA) [23] flow to apply fine-grained TMR to the Rocket Chip and used custom HDL to inject faults through the internal configuration access port (ICAP) [24J. That work also performed heavy-ion testing on the TMR design, only achieving a 3 x reduction in the cross section.

Other work has also targeted the VexRiscv processor for fault injection over the JTAG interface (25]. That work used Synplify to automate the application of fine-grained TMR to the DUT and achieved a 1.5 x improvement in the meantime to failure (MTTF). Using the same JTAG tool, the BYU JTAG Configuration Manager (JC M), this m1icle provides additional fault injection results but differs by using a different TMR tool and comparing those results to neutron radiation data [26).

This micle builds on previous experiments targeting a bare metal application on isolated Taiga RISC-V processors using local BRAM memory (5), (6). The bare metal application requires no operating system to execute. Two Taiga experimental designs were implemented on the Kintex UltraScale KCU105 development board with the XCKU040-2FFVA l 156E FPGA, one containing 20 unmitigated processors and the other containing 20 TMR processors. The Xilinx UltraScale devices are produced with 20-nm FinFET technology (27). Along with a fault injection campaign, these two experimental designs were exposed to a neutron radiation beam at the Los Alamos Neutron Science Center (LANSCE). There was a 33 x reduction in the neutron cross section between the unmitigated and TMR designs. matching closely to the 32.5 x improvement seen in the fault injection results. With this reduction in neutron cross section and the 27% decrease in operational frequency, the TMR Taiga soft-core achieved a 24 x improvement of the mean work to failure.

This article uses the more complex VexRiscv SoC rnnning benchmarks within a Linux operating system. This system requires additional MMU and DDR controller digital hardware. This micle also provides further analysis to identify all the different failure modes observed during the radiation test.

Ⅲ. DESIGN UNDER TEST

Two VexRiscv experimental designs were implemented on the Digilent Nexys Video development board with the XC7A200T- l SBG484C FPGA. one containing the unmitigated design and the other containing the TMR design . The Xilinx Series 7 devices are produced with 28-nm planar technology compared to the 20-nm technology of previous work. These designs utilized the preconfigured Linux SoC from "Linux on LiteX-VexRiscv" open-source project [1). This included a specific design for the Nexys Video board and an operational Linux image. The Linux images were loaded over the Ethernet connec tion , and universal asynchronous receiver-transmitter (UART) was used to monitor the status of the device. To verify the functionality of the processor during the experiments. the Buildroot-provided Dhrystone benchmark was executed and the results were reported over a UART connection during the tests [28].

The utilizations for the LUT, look-up table random-access memory (LUTRAM), FF, and BRAM resources of these designs are reported in Table I. The VexRiscv processor is more complex and requires more FPGA resources to implement. To generate the TMR processor design. all digital logics were targeted by the BL TMR tools , including the memory interface generator (MlG) controller, Ethernet interface, and UART module. The TMR design required 4.1 x more resources to implement. The default system clock of 100 MHz was used for both designs, though with the addition of TMR voters the slack for timing closure was reduced.

TABLE I
VEXRISCV SOC DESIGN UTILIZATION

Design	LUT	LUTRAM	FF	BRAM
Unmitigated	6791 (5.0%)	319 (0.7%)	5506 (2.1%)	43 (11.8%)
TMR	27916 (20.9%)	957 (2.1 %)	16512 (6.2%)	129 (35.3%)
Cost Ratio	4.11 x	3x	3x	3x

TABLE II
FAULT INJECTION UN YEXRISC'V-L!NUX

Design	Upsets	Failures	Sensitivity	Improvement
Unmitigated	41206	319	0.774%	I.OOx
TMR Processor	504258	280	0.055%	13.94 x

JV. PREIRRADIATION FAULT INJEC'TION

Fault injection is a technique that can be used to emulate CRAM upsets within an SRAM-based FPGA [29) . By interfacing with the FPGA configuration manager, configuration frames can be read, modified, and written back to the device during operation. After the emulated upset, the FPGA design can be te sted for any functional failures. This type of fault injection is only capable of emulating the CRAM SEU subset of the possible single-event effects (SEEs) and does not take into account any BRAM SEUs or single-effect functional interrn pts (SEFls). These tests provide a deterministic, scalable method that can potentially inject every CRAM bit within the device.

The BYU JCM was able to inject random faults directly into the FPGA's configuration memory consisting of 77 845 216 bits. Each injection is an independent event resulting in either a normal working system or a functional failure. Each failure represents a single sensitive bit in the configuration of the FPGA design. The sensitivity of the fault injection campaign is the ratio of failures to the total emulated upsets . These sensitive bits are not necessarily deterministic and depend upon the state of the design as shown in Section VI. which details the post-radiation fault injection tests. There are also additional failure modes of multibit CRAM upsets, BRAM upsets, and SEFls that these fault injection tests did not emulate.

After injecting a fault, the system allows this fault to propagate for 5 ms (approximately 500000 clock cycles with a 100-MHz clock). The processor rep011s the Dhrystone self-test results over JTAG and the JCM scrubbed the injected fault by writing the correct configuration frame to the FPGA. The following conditions are considered errors : failure to boot Linux, an incorrect Dhrystone output, and failure to respond to UART communication. Each of these errors triggers a local reconfiguration. If the processor continued to perform incorrectly after a repaired fault injection , the entire device was reconfigured.

The fault injection results in Table II shows the recorded failures and failure rate for both the unmitigated and TMR design . The TMR design showed a l 3.94 x improvement in sen sitivity of single-bit CRAM upsets over the unmitigated design. The reduction in sensitive bits achieved with the TMR

TABLE Ill
N EUTRON RADIATION TEST DATA

Design	LUT Utilization	Normalized Fl U T H 1zahon	(n/ 2 uence cm	Observed CRAMU	F ·1 psets a1 ures	Cross Section (cm 2	+95% Confidence -95% Confidence	R d e	f uc 10n
Taiga Core Unmitigated	43350 (17.9%)*	6.38 X *	1.15 X 10^{10}*	2527*	52*	2.27 X 10·10	2.44 X 10- 2.10 X 10-10		Ix
Taiga Core **TMR**	222029 (91.6%)*	32.69x*	2.QQ X 10^{11}*	52139*	27*	6.76 X 10-11	9.8] X 10- - 4.45 X 10-I		33x
VexRiscv Unmitigated	6791 (5.0%)	I X	1.92 X 10"	46046	306	1.59 X 10^{-9}	1.6] X 10- 1.57 X 10^{-9}		Ix
VexRiscv **TMR**	27916 (20.9%)	4.11 X	4.34 X 10^{11}	100371	69	1.59 X 10- 10	J.68 X 10- 1.50 X IQ-10		!0 x

* Results for experimental designs containing 20 Taiga cores.
One Taiga processor is comprised of 2178 LUTs which is 0.23% of the VexRiscv utilization.

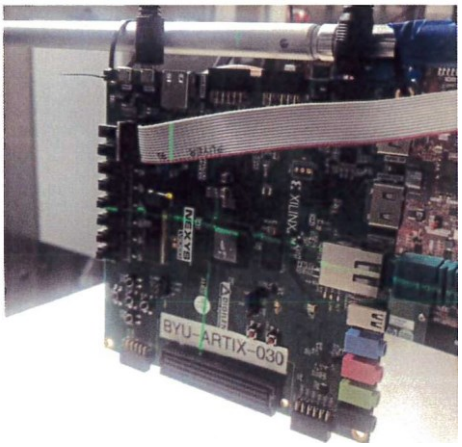

Fig. 2. Y xRiscv experiment at Chiplr

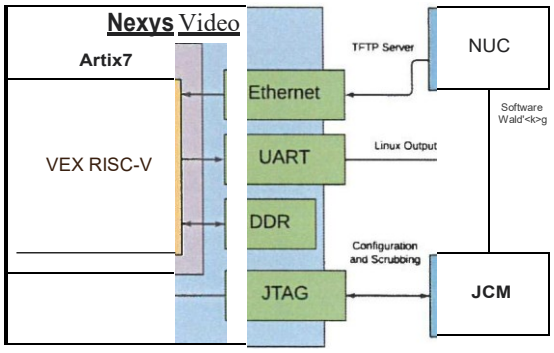

Fig. 3. Ve,,_R_iscv system tesl diagram.

VexRiscv processor was less than the 33 x reduction that was observed in previous work regarding the TMR Taiga Processor [6].

V. RADLATION TESTING

The two YexRiscv processor designs were placed in a high-energy neutron radiation beamline at the Chiplr facility at the Rutherford Appleton Laboratory, U.K. [30). This neutron beam is commonly used for testing integrated circuits to estimate circuit sensitivity to atmospheric neutrons [31]. The board was placed normal to the beam source and operated at room temperature. A collimator was used to restrict the beam to only the FPGA device on the board (the DOR memory used in this system was not in the beam path). Fig. 2 shows the development board positioned for the radiation test.

The organization of the test system is shown in Fig. 3 . The test system includes the Nexys Video FPGA board. a host computer next unit of computing (NUC), and the embedded JCM JTAG controller. The Nexys Video board includes the FPGA device under tes t. the DOR memory that provides main memory for the Linux system, an Ethernet interface for loading the Linux image, and a UART interface to monitor the system operation. The host NUC computer performs seve ral function s during the test. First. it upload s the Linux image to the DOR memory of the Nexys board using the trivial file transfer protocol (TFTP). Second. it monitor s the output of the VexRi scv UART to detect processor hang s or incorrect Dhrystone execution. Third. it logs the operation of the **JCM** scrubber system (described below). An en-or in the system is recorded when the system failed to hoot Linux. produces an incorrect Dhrystone output , or detects no UART communication from the VexRiscv processor.

A. Configuration Scrubbing

During the radiation test. there is accumulation of SEUs within the configuration logic. A repair mechanism known as CRAM scrubbing is employed to actively correct the upset bits within the CRAM. When implementing TMR mitiga ti on, a repair mechanism is essential in preventing multiple TMR domain failure and drastically improving the effectiveness of TMR [32]. This experiment uses the BYU JCM with a 25 Mbps data rate to perform continual CRAM scrubbing, record any upsets within the FPGA fabric. and reconfigure the FPGA when the system needed to recover.

The continuous scrubb ing process involves a series of discrete *scrub cycles* in which the contents of the entire CRAM memory are read and compared against the golden CRAM state. Any errors that are found during a scrub cycle are repaired through partial reconfiguration over JTAG. The average time for each scrub cycle is 5 s. Placed in its position at the Chiplr facility, there were an average of 6.0 CRAM upsets detected during each scrub cycle. The actual number of up sets varies from scrub cycle to scrub cycle and follows a Poisson distributi on as demon strated in Fig. 4. All CRAM upsets identified and repaired by th e JCM were logged with the scrub cycle time stamp.

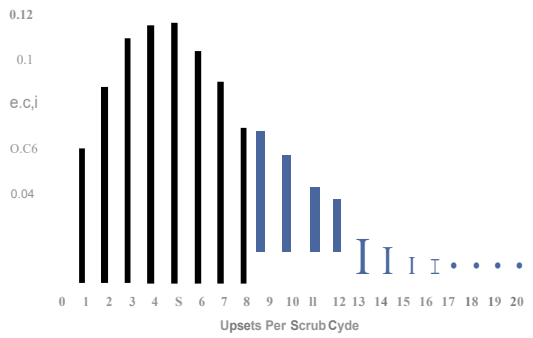

Fig. 4. Distribution of CRAM upsets in scrubbing cycles.

B. Radiation Test Results

The effectiveness of the TMR is represented hy the reduction in the cross section, which is the ratio between the failures and the total fluence. The radiation test results in Table III shows LUT utilization, the total fluence each design was exposed to, and the observed CRAM upsets, along with two top rows reporting results for the previous work 20 individual Taiga RISC-V soft-core [5]. The estimated cross section for each design is reported with 95% confidence intervals [33]. The TMR VexRiscv processor achieved a l0x reduction in the neutron cross section of system failures.

Dming the radiation test, the JCM observed multiple flipped bits between scrub cycles as a result of the accelerated induced upset rate. This suggests that some failures could have been caused by multibit CRAM upsets. Failures could also be caused by upsets in FFs, BRAMs, and other FPGA primitives not observed by the JCM. These additional failure modes can account for the TMR design's lower improvement rate of lOx compared to the 15 x improvement found with the fault injection campaigns. The TMR VexRiscv achieved a lower improvement in SEU mitigation than the isolated **TMR** Taiga soft processor [5]. Unlike the Taiga, the VexRiscv SoC is a complicated system with layers of volatile memory (some off-chip) that is not being actively repaired during the operation and various single-point interfaces that are critical to the operation of the processor.

VI. POSTRADIATION FAULT INJECTION

Additional fault injection experiments were conducted for the **TMR** VexRiscv system after the radiation test to better understand the behavior of the TMR system in the beam and to identify single-point failures in the design . The post-radiation fault injection approach was performed using the radiation CRAM upset data to playback every observed upset individually. This section summarizes the results from these experiments and describes the additional insights into the behavior of this system.

This post-radiation fault injection approach involves the *playba ck* of observed CRAM upsets. This test will inject each of the CRAM upsets observed at the beam into the design and observe the behavior of the TMR VexRiscv sys tem . These CRAM bits are injected one at a time and the configuration memory is scrubbed in between CRAM fault injection to emulate single-bit CRAM upsets. The purpose of the test is to see which of the CRAM bits upset during the radiation beam test cause system failures in the system.

The address of all 100 371 CRAM upsets observed in the beam were logged along with their timestamp and configuration scrub cycle. These logs were organized into a "playlist" and the fault injection tool was modified to inject these faults from the playlist in sequential order. Like the preitrndiation fault injection approach, the system executed the full duration of the Dhrystone benchmark for each injected fault to allow for error propagation. If an error was observed in the behavior of the system, the corresponding CRAM bit was tagged as *sensitive*.

Several iterations of this playback fault injection were performed for each CRAM bit, and the results indicated that the sensitive CRAM bits do not always cause the system to fail. Because of the dynamic nature of this processor system, a CRAM fault inserted into the system in one iteration may cause a system error but not during another iteration. Because of the temporal, probabilistic nature of this fault injection, each CRAM bit was injected in the system multiple times to estimate a "sensitivity rate," for each sensitive CRAM bit. The sensitivity rate, s;, of a specific CRAM bit i is the probability that the given CRAM cell will cause a system error when upset. The sensitivity rates of each sensitive CRAM bit will differ and the multiple playback iterations were performed to estimate the distribution of CRAM sensitivity rates among the upset CRAM bits observed at the beam test.

The full postradiation fault injection playback was performed 20 times [1] to estimate a discrete distribution of CRAM sensitivity rate. After completing this playback 20 times, 54 of the l00 371 CRAM bits observed in the radiation beam test were observed to cause a failure in at least one playback iteration. The distribution of CRAM bits and their sensitivity rate is summa1ized in Table IV. The first column indicates the number of times, *i,* the given CRAM bit caused a design error out of 20 trials. The second column is the sensitivity rate and is computed by dividing i by 20 (s; = i / 20). The third column indicates the number of CRAM hits, *11.* out of the total 54 sensitive bits that caused *i* errors during the 20 iterations . The final column is the estimated probability of a sensitive bit having the given sensitivity rate and is computed by dividing *n* by 54. Note that the table does not include rows for the sensitivity rates that were not seen in this playback (i.e., there were no CRAM bits that failed in exactly three of the 20 trials).

The expected value of the sensitivity rate distribution for sensitive CRAM bits is computed as follows:

$$/l = L,S; \cdot p;.$$

The expected sensitivity rate for the sensitive bits in this experiment was calculated as *11* = 73.2 %. The expected number of errors caused by these 54 **CRAM** upsets during the radiation test can be estimated by multiplying the estimated sensitivity rate by 54, or 73.2% x 54 = 39.6. This estimate suggests that on average, the number of errors observed at the radiation test due to single CRAM sensitive bits is ~40.

[1] Thi s fault injection experim ent took ov er t wo wee ks to perfonn the 2 million CRAM fault in jection s.

TABLE IV
NUMBER OF FAILURES CAUSED BY EACH CRAM BIT DURING FAULT INJECTION PLAYBACK

# of Failures (i)	Sensitivity Rate (s;)	# of CRAM bits (n)	Probability p;
1	.05	7	12.9%
2	.10	1	1.9%
7	.35	1	1.9%
8	.40	1	1.9%
9	.45	2	3.7%
10	.50	9	16.7%
19	.95	1	1.9%
20	1.00	32	59.2%
		54	100%

The overall sensitivity of the TMR design to single-bit CRAM upsets in this playback experiment is 40/100371 = 0.040%. This is lower than the design sensitivity estimate of 0.055% shown in the preirradiation fault injection experiment (see Table II).

These results suggest that not all errors seen in the radiation test can be attributed by single-bit CRAM upsets. Multibit CRAM upsets and unobserved upsets with in FFs, BRAMs, and other FPGA primitives could account for the remainder of the errors. Multibit CRAM upsets have proved difficult to identify with fault injection using the radiation data. Future work will perform further fault injection to properly identify potential multibit upsets.

VII. POINT OF FAILURE NETLIST ANALYSIS

The final activity performed to understand the behavior of this system in the radiation test is to attempt to identify the location of the single-bit upsets that caused failures in the system. The approach used to find these single-point failures is to determine the purpose of the 54 sensitive CRAM bits identified in the playback approach of Section **VI.** The tile location of each of these bits was determined to identify nets and logic that may be susceptible to failures. Thirty-eight unique tile locations were identified from these 54 sensitive bits.[2] Once the tile locations were identified, a Vivado TCL script generated a list of nets that use the given tile and represent potential single-point failures in the design.

Table V shows the breakdown of how many tiles contained each type of net. Most of the affected tiles included nets to the synchronous dynamic random-access memory (SDRAM) DDR and the OSERDESE/ISERDESE primitives used for its interface. Some tiles contained two or more of the triplicated system clocks and resets where potentially one upset could affect multiple TMR domains and compromise the mitigation scheme. The remaining nets identified are related to logic for the processor and its system cache.

Fig. 5 shows the floorplan for the DUT with the highlighted sensitive tiles of the playback fault injection data. Many of the highlighted tiles on the right side contained nets for the SDRAM DDR interface. Though the TMR tools triplicated the logic for this interface, they could not triplicate the primitives associated with the I/O pins such as OSERDESE/ISERDESE.

[2]A more thorough random fault injection identified 175 tiles that may result in system failure.

TABLE V
NUMBER OF CLASSIFIED NETS IN SENSITIVE TILES

	Random Injection Tiles	Playback Tiles
Total Tiles	**175**	**38**
sdram	147 (84.0%)	31 (81.6 %)
OSERDESE	80 (45.7%)	18 (47.4%)
ISERDESE	122 (69.7%)	26 (73.7%)
TMR sys clks	27 (15.43%)	10 (26.32%)
TMR sys rsts	19 (10.86%)	3 (7.89%)
dataCache	18 (10.29%)	4 (10.53%)
dBusWishbone	25 (14.29%)	5 (13.16%)
decode_to_execute	18 (10.29%)	5 (13.16%)
execute_to_memory	17 (9.71%)	4 (1053%)
memory_to_writeBack	16 (9.14%)	2 (5.26%)
ethmac	19 (10.86%)	1 (2.63%)
IBusCache	18 (10.29%)	5 (13.16%)
storage (Cache)	52 (29.71%)	5 (13.16%)

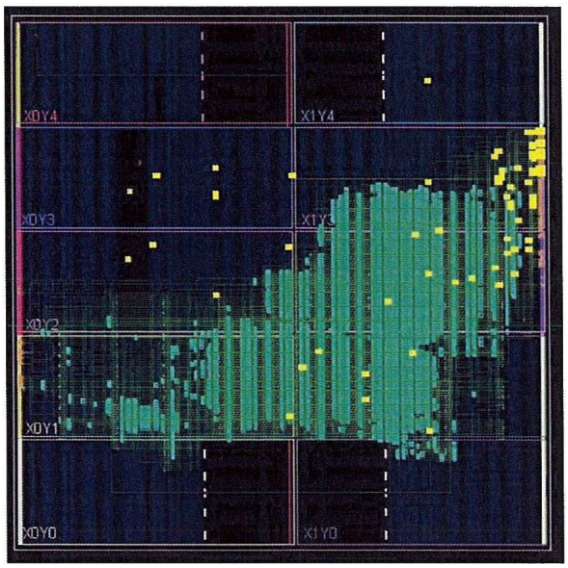

Fig. 5. Floorplan with highlighted playback sensitive tiles.

Nontriplicated physical IO pins are single points of failure that cannot be mitigated by any digital hardware method, but may have to rely on protocol level error detection and correction. Even if error correction code (ECC) was employed to mitigate any single errors within the data lines. that mitigation could not be extended to any of the address or control lines.

VIII. CONCLUSION

TMR is proved to be an effective tool to mitigate the effects of SEUs within the FPGA fabric. A TMR soft-core processor can see up to a 33x improvement in reliability at the cost of potentially 5x resource utilization and decreased operation frequency. As soft-processors are integrated into more complex systems with on-board memory and communication interfaces, there are more events which can cause a failure in a radiation environment. The **TMR** Linux-capable. soft-core processor only demonstrated a 10x improvement. A **TMR** soft-processor can provide effective improvement to the reliability of the digital logic, but to operate on potentially corrupt instructions/data and deliver protected results, the processor may need to rely on additional hardware and software mitigation techniques.

The post-radiation fault injection proved successful in understanding the different failure modes for a complex soft-core processor system that extends beyond the FPGA

with various off-chip memory and interfaces. The operation of the processor is defined by the software, thus a BRAM or DDR SEU may lead to a critical failure. This article introduced further analysis of the neutron radiation data by using targeted fault injection. This analysis presented several concerns introduced by the greater complexity. Failures caused by SEUs can be time-dependent and difficult to replicate even with a deterministic approach such as fault injection. BRAM and DDR SEUs require additional hardware to monitor, and the specifics of a SEFI may be impossible to monitor. A placed and routed TMR design may have single-point failures introduced by how the tools route critical clock and reset lines. TMR digital logic does not mitigate against any SEUs affecting external nontriplicated IO such as the DDR interface.

In future works, through additional monitoring and targeted fault injection, sources for failures within these complex designs will be investigated and categorized. Additional techniques targeting hardware, memory, and software could be tested to mitigate these failures and provide a more reliable system for space applications targeting Xilinx SRAM-based FPGAs.

REFERENCES

[I] Spina!HDL. (2019). *VexRiscv*. [Onlinej. Available: https://github.com/SpinalHDUVexRiscv

[2] H. Quinn, P. S. Graham. **K.** Morgan. J. Krone, M. P. Caffrey. and M. J. Wirthlin, "An introduction to radiation-induced failure modes and related mitigation methods for Xilinx SRAM FPGAS." in *Proc. Int. Conj. Eng. ReC(}nfigurablc Syst. Algorithms (£ RSA),* T. P. Plaks, Ed. Las Vegas, NY, USA: CSREA Press, 2008. pp. 139-145.

[3J Y. lchinomiya. S. Tanoue, M. Amagasaki, M. Iida, M. Kuga. and T. Sueyoshi, ·'Improving the robustness of a softcore processor against SEUs by using TMR and partial reconfiguration." in *Proc. 18th IEEE Annu. bi/. Symp. Field-Program. Custom Comput. Mach.,* May 2010, pp. 47-54.

[4J P. Graham. l Caffrey, J. Zimmerman, D. E. Johnson. P. Sundararajan. and C. Patterson, "Conseq uences and categories of SRAM FPGA configuration SEUs ," in *Proc. 5th A11n11. Int. Conf Mil. Aerosp. Program. Log. Del'ices (MAPLD),* 2003. [Online]. Available: http://c iteseer x. is t.psu.edu/v ie wdoclsummary "do i= l0.1.l.484.9371

15) A. E. Wilson and M. Wirthlin, "Neutron radiation testing of fault tolerant RISC-V soft processor on Xilinx SRAM-based FPGAs ," in *Proc. IEEE Space Comput. Conf (SCC).* Jul. 2019 , pp. 25-32.

[6] A. E. Wilson. C. Thurlow. and M. Wirthlin. "Fault injection testing of fault tolerant RISC-Y soft processors on Xilinx SRAM-based FPGAs.'' *J. Radial. £/feels Res. Eng.*. vol. 39, no. I. pp 356-36I, Apr 2021.

[7] C. Heinz, Y. Lavan. J. Hofmann, and A. Koch, "A catalog and in-hardware evaluation of open-source drop-in compatible RISC-V soft-core processors," in *Proc. Int. Conj. Reconfigurable Comput. FPGAs (ReCnnFig),* Dec. 20I9, pp. 1- 8 .

[8] E. Matthews. Z. Aguila, and L. Shannon, "Evaluating the performance efficiency of a soft-processor, variable-length. parallel-execution-unit architecture for FPGAs using the RISC-V ISA ." in *Proc. IEEE 26th Ann11. Int. Symp. Field-Program. Custom Comp111. Mach. (FCCM),* Apr. 2018. pp. J-8 .

[9J E. Matthews and L. Shannon, ·T AIG A: A new RISC-V soft-processor framework enabling high performance CPU architectural features," in *Proc. 27th Int. Cnnf Field Program. Log. Appl. (FPL).* Sep. 2017. pp. 24-27.

r10J (Oct. 2019). *Limn on LiteX \lexRiscv*. [O nline] . Available: https://g it hub .co m/l ite x-hu b/linux-on-li te x-vexr isc v

[l lj J. M. John son and **M.** J. Wirthlin , " Voter insertion algorithms for FPGA designs using triple modular redundancy ." in *Proc. 18th* A 111111. *ACMISIGDA Int. Symp. Field Program. Gate Arrays (FPGA).* New York . NY. USA ACM, 2010, pp. 249- 58, doi I0 .11 45 / 1723 11 2 . 1 72 3 1 54 .

[12] A. **M.** Keller and **M** . J. Wirthlin, "Benefits of complementary SEU mitigation for the LEON3 soft processor on SRAM-based FPGAs." *IEEE Trans. Nuc/. Sci., vol.* 64, no. I. pp. 519-528. Jan. 2tl17 .

l13] M. J. Wirthlin. A. M. Keller, C. McCloskey. P. Ridd, D. Lee, and J. Draper. "SEU mitigation and validation of the LE0 3 soft processor using triple modular redundancy for space processing." in *Proc. ACM/SIGDA Int. Symp. Field-Program. Gate Arrays* . New York. NY, USA: ACM , Feb. 20 16 , pp. 205-214. doi: I0 . 11 5/28472 63 .2847278 .

l l4J A. Lindoso. L. Entrena, M. Garcia-Valderas, and L. Parra, "A hybrid fault-tolerant LEON3 soft core processor implemented in low-end SRAM FPGA," *IEEE Trans. Nucl. Sci., vol.* 64, no . I. pp. 374-381, Jan. 2017.

[l5J M. Psarakis. A. Yavousis. C. Bokhini, and A. Miele. "Design and implementation of a self-healing processor on SRAM-based FPGAs," in *Proc. IEEE Int. Symp. Defect Fault Tolerance VLSI Nanorechnol. Syst. (DFT),* Oct. 2014, pp. 165-170.

[16] N. H. Rollins, "Hardware and software fault-tolerance of softcore processors implemented in SRAM-based FPGAs." Ph.D. dissertation, Dept. Elect. Comput. Eng., Brigham Young Univ., Provo. UT. USA , 2012.

[l 7] C. Hong , **K.** Benkr id, X. Iturbe , and A. Ebrahim. " Design and implementation of fault-tolerant soft processors on FPGAs." in *Proc. 22nd Int. Conj Field Program. Log Appl. (FPL).* Aug. 20I*2,* pp. 683-686.

[l 81 I. M. Safarulla and K. Manila] , "Design of soft error tolerance technique for FPGA based soft core processors." in *Proc. IEEE Int. Con{ Adi·. Commun.. Control Comput. Technol.,* May 2014, pp. 10 36- 1040.

l I 9] Xilin x. (Oct. 2018). *Microblaze Triple Modular Redundanc y (TMR) S11bsvs1em vl.V.* [Online]. Available: https://www.xilinx.com/support/documentation/ip documentation/tmr/vI 0/pg268-tmr.pdf

[20] ChipsAlliance. (2020). *Rocket-Chip*. [Online]. Available: https://github.com/chipsalliance/rocket-chip

[2I J A. Ramos , J. A. Maestro, and P. Reviriego, "Characterizing a RISC-V SRAM-based FPGA implementation against single event upsets using fault injection ." *Microelectron. Rel.*. vol. 78, pp. 205-21 I , Nov. 2017. doi: I 0 . I 016/j .mic rore l.20 17 .09. ()()7 .

[22J L. A. Aranda *et al.*. "Analysis of the critical bits of a RISC-Y processor implemented in an SRAM-ba ed FPGA for space applications," *Electro11ics, vol. 9.* no . I , p. 175 , Ja n. 2020.

[23] L. A. C. Benites and F. L. Ka tensmidt, "Automated design flow for applying triple modular redundancy **(TMR)** in complex digital ci rcuits ." in *Proc. IEEE 19th Latin-Ama Test Symp. (LA.TS),* Mar. 2018, pp. 230-233.

[24] A. B. D. Oliveira *et al.,* "Evaluating soft core RISC-V processor in SRAM-based FPGA under radiation effects," *IEEE Trans. Nucl. Sci.*. vol. 67, no. 7, pp. 1503-1510, Jul. 2020.

[251 F. Minnella. "Protection and characterization of an open source soft core against radiation effects." Polytech. Univ. Turin, Turin, Italy. Tech. Rep. CERN-THESIS-2018-028. Apr. 20 I 8. [Onl ine] . Available: http://cds.cern.ch/record/23I3417

[26] A. Gruwe ll, P. Zabrisk ie , and M. Wi rth li n, "High-speed FPGA configuration and testing through JTAG," in *Proc. IEEE AUTOTESTCON,* Sep . 2016. pp. 218-225.

[27J *Delivering a Generation Ahead at 20 nm and 16 nm.* Accessed: 2020. [Online]. Available: https://www .xilinx.com/abou t/gene ratio n-ahead-J 6nm.html

[28] *B11ildroot Dhrystone Package.* Accessed: 2020. (Online"]. Available: hrtps://github.com/buildroot/buildroot/tree/master/package/dhrystone

[29] C. Thurlow. H. Rowberry , and M. Wirthlin, "TURTLE: A low-cost fault injection platform for SRAM-based FPGAs." in *Proc. Int. Conf ReConFigurable Co\ll pl\lt. FPGAs (ReConFigJ,* Dec. 2tl19, pp. 238- 245 .

[30] (2019) . *!SIS Chiplr Technical lnfom1atio11.* [Online] . Available: https://www. isis.stfc.ac.u k/Pages/Chi pir-technical -information.aspx

[31 j C. Cazzaniga and C. D. Frost, "Progress of the scientific commissioning of a fast neutron beamline for chip irradiation." *J. Ph ys., Conf. Ser.*. vol. 1021. May 20 18 , Art. no . 0 1 2037 .

[32] M. Berg *et al.,* '•E ffect ivene ss of internal vs. External SEU scrubbing mitigation strategies in a Xilinx FPG A: Des ign, test. and analysis.'· in *Proc. 9th Eur. Conj. Radial . Its £{Jeers Compon. Syst.,* Sep. 2007, pp. 459-466.

[33l H. Quinn. "Challenges in te sting complex systems." *IEEE Trans. Nu cl. Sci.*. vol. 61. no. 2. pp. 766-786, Apr. 2014 .