Neutron Radiation Testing of RISC-V TMR Soft Processors on SRAM-based FPGAs

Andrew E. Wilson, Michael Wirthlin, and Nathan G. Baker

***Abstract*—Soft, configurable processors within FPGA designs are susceptible to SEUs. SEU mitigation techniques such as TMR and configuration memory scrubbing can be used to improve the reliability of soft processor designs. This paper presents the improvements in reliability of five different TMR soft processors within a neutron radiation environment. The TMR processors achieved up to a 75× improvement in reliability at the cost of potentially 4.8× resource utilization and an average 12.4% decrease in maximum frequency compared to the unmitigated designs. This work compares the metrics of reliability, power consumption, and performance among the default unmitigated processors and their TMR variations.**

***Keywords*-RISC-V; Fault tolerance; redundancy; Triple Modular Redundancy (TMR); Single Event Upset (SEU); fault injection; radiation testing; FPGA; soft processor; Radiation hardening by design**

I. Introduction

Static random access memory (SRAM)-based field programmable gate arrays (FPGAs) often include soft processor implementations to provide software programmability. 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). A variety of soft processor cores are available including open source RISC-V processor implementations utilizing an open instruction set architecture (ISA). Integrating soft processors in an FPGA can be beneficial for applications in both terrestrial and space environments.

Soft processors are susceptible to single event effects (SEEs) induced by radiation found in both space and terrestrial environments. Single event upsets (SEUs) are of special concern within FPGAs as they contain a large amount of configuration RAM (CRAM) and internal block RAM [1]. SEUs can cause functional failures in the design within the FPGA by corrupting the state and circuit configuration, potentially leading to a critical failure of the system. The use of SEU mitigation techniques to mask these failures may be required to provide sufficient reliability in the system [2].

Triple modular redundancy (TMR) is an effective mitigation technique capable of masking single point failures within FPGA soft processors [3]. Although TMR provides additional reliability through redundancy, this mitigation technique results in greater power consumption, higher resource utilization, and slower maximum frequency. To prevent an accumulation of SEUs causing a multi-point failure, a repair mechanism known as CRAM scrubbing dynamically corrects any upsets within configuration memory without disrupting operation [4].

This paper investigates the reliability of four different implementations of the RISC-V ISA and measures the improvement in reliability achieved through TMR mitigation. This work also includes the Xilinx MicroBlaze processor as a vendor provided reference example. Previous works have investigated TMR RISC-V processors running a Dhrystone application in both a bare metal environment as well as within a Linux operating system (OS). The stand-alone Taiga processor achieved a 33× improvement in mean-time to failure for both fault injection [5] and neutron radiation testing [6]. The Linux VexRiscv System on Chip (SoC) achieved a 10× improvement with fault injection and neutron radiation testing [7]. Along with these implementations, this paper introduces three additional soft processors (MicroBlaze, PicoRV32, and NOEL-V) on the Xilinx Kintex UltraScale.

Several improvements have been made over previous soft processor testing approaches [8]. First, the processors evaluated in this paper use a more robust CoreMark benchmark over the more outdated Dhrystone benchmark. Second, triplication of the processor designs is performed by the SpyDrNet Python API [9] and TMR tools. Third, configuration scrubbing is performed by utilizing the high-speed SelectMap configuration interface with a 32-bit wide bus at higher speeds compared to previous serial JTAG methods. The TMR processors in this experiment achieved up to a 75× improvement in reliability at the cost of potentially 4.8× resource utilization with an average 12.4% decrease in maximum frequency.

This paper describes the implementation of each experimental design and measures the cost of TMR in terms of both logic utilization and clock frequency reduction. This paper explains the setup for the neutron radiation testing and the fault injection campaign as well as the results of these tests. Using these results, as well as performance and dynamic power consumption, these soft processors are compared to each other and their TMR implementations using single value metrics as well as a composite metric that combines the metrics together.

II. Fault Tolerant RISC-V Soft Processors

RISC-V is an open instruction set architecture (ISA) maintained by the RISC-V Foundation that is used in academia, research, and industry [10]. RISC-V has specifications for 32-bit, 64-bit, and 128-bit processors with established extensions and opportunity for custom application-specific instructions.

This work was supported by the I/UCRC Program of the National Science Foundation under Grant No. 1738550.

The authors are associated with the NSF Center for Space, High-performance, and Resilient Computing (SHREC), and Brigham Young University, Provo, Utah, USA (email: {andrew.e.wilson, wirthlin, nathangary-baker}@byu.edu)

There are many soft RISC-V processors available for implementation within FPGAs with the support of existing software tool-chains and libraries.

The available RISC-V processors range widely in features, performance, efficiency, and utilization of FPGA resources. The TaPaSCo RISC-V repository has compared the difference in performance and maximum frequency achieved within a selection of RISC-V soft processors in Xilinx SRAM-based FPGAs [11]. Another comparison of RISC-V SoCs in the Litex ecosystem was performed [12]. These results suggest a wide variety of implementation styles that include processors optimized for SRAM-based FPGAs and others optimized for an ASIC and ported to FPGAs for prototyping.

A. TMR RISC-V

Soft processors can tolerate SEUs by spatial redundancy such as triple modular redundancy. A triplicated design includes three redundant domains and triplicated voters capable of masking a fault at any single location. The triplicated majority voters mask erroneous output of a single domain and use the majority output of the other two domains [13].

This paper uses SpyDrNet, a Python-based netlist tool, to generate the TMR designs [9]. SpyDrNet performs fine-grained TMR on the FPGA primitives at the netlist level by triplicating all FFs, LUTs, BRAMs, and DSPs. Triplicated voters are inserted at carefully selected locations between these primitives. The input to SpyDrNet is a vendor-independent Electronic Design Interchange Format (EDIF) file that can be exported from Xilinx Vivado. The generated TMR EDIF file can be imported back into Xilinx Vivado as a post-synthesis file, and placed and routed in the final design (see Figure 1). This TMR tool has achieved a 20× to 500× reduction in CRAM sensitivity fault injection results for complex processor designs in previous works [8].

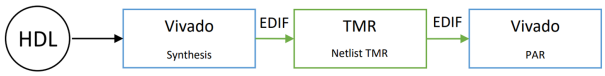

Fig. 1. TMR with triplicated voters.

The TMR RISC-V processor needs a repair mechanism to continually recover the system from the masked errors in order to prevent multiple TMR domains from failing. Without a repair mechanism, SEUs will collect within the CRAM frames until TMR is unable to mask failures. To repair SEU upsets in the CRAM, scrubbing of the configuration memory is deployed to correct corrupt bits within the configuration frame and prevent the accumulation of upsets. CRAM scrubbing is performed by rewriting configuration memory with the correct FPGA bitstream data in order scrub any incorrect values [14]. For Xilinx FPGAs, this configuration scrubbing can be performed over JTAG, the high-speed SelectMap interface, or the internal ICAP interface. Configuration scrubbing does not affect the operation of the FPGA or any memory within.

B. Related Works

Many different soft processors have been targeted for space applications. Several studies have modified the LEON2 and LEON3 processors for improved reliability with different mitigation schemes including TMR [15]–[19]. 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 fully featured HDL implementation officially supported by the RISC-V foundation [20]. One work used the Mentor Precision Hi-Rel tool to apply fine-grain TMR to the Rocket Chip and performed fault injection [21]. Their study achieved a reduction in sensitive bits of up to 11.5×. Another work used Cadence's EDA [22] flow to apply fine-grain TMR to the Rocket Chip and used custom HDL to inject faults through the internal configuration access port (ICAP) [23]. That work also performed heavy-ion testing on the TMR design, achieving a 3× reduction in the cross section. This paper targets more FPGA-optimized soft processors that run at higher frequencies and require fewer FPGA resources.

Other work has also targeted the VexRiscv processor for fault injection over the JTAG interface [24]. That work used Synplify to automate the application of fine-grain TMR to the DUT and achieved a 1.5× improvement in the mean time to failure (MTTF). Using the same JTAG tool, the JTAG configuration manager (JCM), this paper provides additional fault injection results but differs by using a different TMR tool and comparing those results to neutron radiation data [25].

Previous experiments implemented the Taiga RISC-V processor on the Kintex UltraScale KCU105 development board [5], [6]. 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× reduction in the neutron cross section between the unmitigated and TMR designs, matching closely to the 32.5× 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 processor achieved a 24× improvement of the mean work to failure.

This paper compares four soft RISC-V processors including the Taiga, VexRiscv, and two additional soft processors (PicoRV32 and NOEL-V) on the Xilinx Kintex UltraScale. The MicroBlaze processor is included as a vendor provided study reference. Related work has investigated the MiroBlaze soft error sensitivity through fault injection [26]. This paper implements a more robust CoreMark benchmark, newer SpyDrNet TMR Python API, and utilizes Xilinx's SelectMap configuration interface for high speed configuration scrubbing.

III. Design Under Test

The soft processors developed in this work target the Alpha Data KU060 development kit (ADA-SDEV-KIT2) with the Xilinx UltraScale FPGA architecture. This FPGA family is developed with 20 nm FinFET technology [27]. This commercial-off-the-shelf (COTS) FPGA is pin compatible with Xilinx's radiation-tolerant space-grade XQRKU060-

CA1509 FPGA. In order to collect more data during radiation testing, multiple instances of the same processors are tiled in a single experimental design. These designs ranged from 8 to 264 tiled processors sharing the same UART telemetry via a ring buffer (see Figure 2). Each processor used 4 kB of local memory to execute identical CoreMark test software and reported the results via a UART connection.

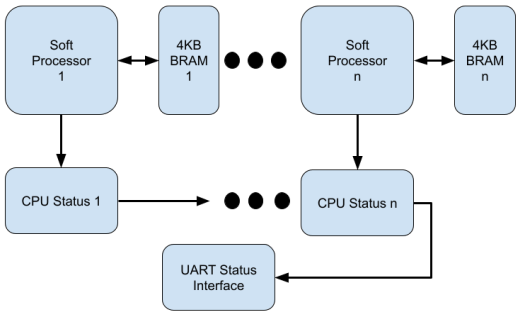

Fig. 2. Tiled Soft Processor Implementation.

A. Targeted Processors

The five different processors that were selected for this experiment are detailed below. The selected configurations are in-order processors with no floating point capabilities.

1) MicroBlaze: The MicroBlaze is a 64-bit capable RISC (not a RISC-V architecture) soft processor using a unique ISA and optimized by Xilinx for its products [28]. Xilinx reports a maximum frequency (F_{max}) of 212 MHz and a Dhrystone million of instructions per second (DMIPs)/MHz of 1.04 for the default configuration implemented on the targeted Artix FPGA. The default configuration is a 32-bit single-issue 5-stage pipeline. This was used for comparison to the following open source solutions used in this experimental design.

2) PicoRV32: The PicoRV32 is a small RISC-V processor that can achieve a high F_{max} when implemented in FPGAs [29]. This processor can achieve a F_{max} of 454 MHz on Xilinx Series 7 devices with a DMIPs/MHz of 0.516. This processor has been formally verified by Symbiotic EDA's RISC-V formal verification framework [30]. The regular configuration of a single-issue non-pipeline processor was used for testing with the addition of an AXI4Lite interface.

3) Taiga: Taiga, a 32-bit RISC-V processor, was chosen for its optimized performance for Intel and Xilinx SRAM-based FPGAs [31]. The processor implements multiple independent execution units, allowing for a single-issue variable-length pipeline. The Taiga has shown better performance when implemented in FPGAs when compared to other open source solutions [11]. The Taiga implementation includes an AXI4 bus for cached transactions and an AXI4Lite bus for peripheral devices. Two additional BRAMs are utilized in this configuration for the instruction and data caches.

4) VexRiscv: The VexRiscv is a 32-bit pipelined processor developed with the high-level language SpinalHDL [32]. This processor has a performance of 1.21 DMIPS/MHz and 2.27 CoreMark/MHz. The processor also takes advantage of a large ecosystem of open source IP within the LiteX project for the quick integration of SoCs within FPGA digital designs [33]. The VexRiscv was configured for maximum performance with a single-issue 5-stage pipeline and AXI4 buses in this experimental design. Two additional BRAMs are utilized in this configuration for the instruction and data caches.

5) NOEL-V: NOEL-V is a dual-issue 64 bit RISC-V processor developed in Cobham Gaisler's RISC-V line of processors [34]. This processor can achieve a performance of 4.69 CoreMark/MHz. The selected TIN32 configuration is a 32-bit single issue 7-stage pipeline. Two additional BRAMs are utilized in this configuration for internal instruction and data memory.

TABLE I
SINGLE PROCESSOR UTILIZATION

Design	LUT	FF	BRAM	DSP	F_{max}
MicroBlaze	1070(0.3%)	922(0.1%)	4(0.4%)	0(0.0%)	272.5
TMR	5126(1.6%)	2766(0.4%)	12(1.1%)	0(0.0%)	215
Cost Ratio	4.79×	3×	3×	n/a	0.79×
PicoRV32	1016(0.3%)	793(0.1%)	4(0.4%)	0(0.0%)	275
TMR	4417(1.3%)	2280(0.3%)	12(1.1%)	0(0.0%)	258.25
Cost Ratio	4.35×	2.88×	3×	n/a	0.94×
Taiga	2261(0.7%)	1473(0.2%)	6(0.6%)	4(0.1%)	200
TMR	10002(3.0%)	2201(0.7%)	18(1.7%)	12(0.4%)	182
Cost Ratio	4.42×	2.99×	3×	3×	0.91×
VexRiscv	2665(0.8%)	2073(0.3%)	6(0.6%)	4(0.1%)	166
TMR	11115(3.0%)	6207(0.9%)	18(1.7%)	12(0.4%)	155
Cost Ratio	4.17×	2.99×	3×	3×	0.93×
NOEL-V	9612(2.9%)	6333(1.0%)	6(0.6%)	4(0.1%)	155
TMR	39712(12%)	19047(2.9%)	18(1.7%)	12(0.4%)	125
Cost Ratio	4.13×	3×	3×	3×	0.81×

B. Utilization

Ten different designs were implemented: five for each unmitigated processor and five for the TMR variants. The utilization and performance of each processor listed in this paper are specific to the selected implementation employed in this experiment and do not necessarily reflect the wide variety of implementation styles each processor may use. The LUT, FF, BRAM, and DSP utilization for these processors are reported in Table I. The maximum frequency (F_{max}) is also reported in MHz.

To generate a TMR version of each processor design, all digital logic (LUT, FF, BRAM, and CARRY) was targeted for triplication by the SpyDrNet TMR tool. The tools did not target input/output buffers, SERDES, MMCM, VCC, or GND primitives. The TMR designs required at least three times as much logic and in one case required 4.8× more resources for SEU mitigation. On average, the TMR designs had a lower F_{max} of 12.4% . The F_{max} was calculated by increasing the frequency output of the MMCM and using Vivado place and route tools to verify a passing timing result.

Each processor ran identical test software consisting of 40 iterations of a CoreMark benchmark. Table II shows the

TABLE II
SINGLE PROCESSOR PERFORMANCE

Design	F_{max} (MHz)	CoreMark Cycle Count	Normalized Performance
MicroBlaze	272.5	612210145	1.00
PicoRV32	275	1213692115	0.51
Taiga	200	152448725	2.95
VexRiscv	166	185641945	2.01
NOEL-V	155	1437999515	0.24

maximum frequency and cycle count of the benchmark. The right-most column normalizes the performance (the product of the F_{max} and CoreMark cycle count) to the vendor provided MicroBlaze processor. The performance for the TMR variants are only affected by the decreased F_{max}, and the normalized values are listed in Table V of Section VI.

The processors were tiled on the FPGA to maximize utilization and data collection, and they reported over serial communication. The tiled designs fully utilize the available resources on the FPGA to increase the number of errors observed at the neutron radiation test and thus improve test statistics. Table III in Section IV lists the LUT utilization for the tiled designs. Even though the designs used 82% to 97% of the available LUT resources, there were no problems with the default vendor placement and routing steps. Other than a global clock and reset, these designs lack long global routes throughout the FPGA.

IV. RADIATION TESTING

Radiation testing induces SEEs on the target system at an accelerated rate [35]. In order to collected sufficient data, the accelerated flux of the high energy particles during the radiation testing is much higher than the flux of radiation during normal operation in the target environment. Neutron radiation testing is used to emulate the effects of the terrestrial neutron environment on a device [36]. This data can be extrapolated to understand how the mitigation techniques will handle SEUs in a space environment [37].

The radiation experiment was performed at the ICE house beam line (target 4, 30 degree right at the Weapons Neutron Research (WNR) facility) at LANSCE [38]. This facility provides a high energy neutron beam that provides a wide spectrum of neutron energies (up to 800 MeV) that closely matches the energy spectrum of neutrons found on earth. Using a single development board, each of the ten designs were tested at a normal angle of incidence and at room temperature as shown in Figure 3. The neutron beam was collimated to 2 inches to limit exposure of other board components.

A. High-Speed Configuration Scrubber

The target development board utilizes Xilinx's high-speed SelectMap configuration interface, providing a 32× wider bus at higher speeds compared to JTAG methods. This experiment uses the JCM with a 500 Mbps data rate to perform continual CRAM scrubbing, record any upsets within the FPGA fabric, and reconfigure the FPGA when the system needs to recover (Figure 3). All CRAM upsets identified and repaired by the JCM were logged with the scrub cycle time stamp.

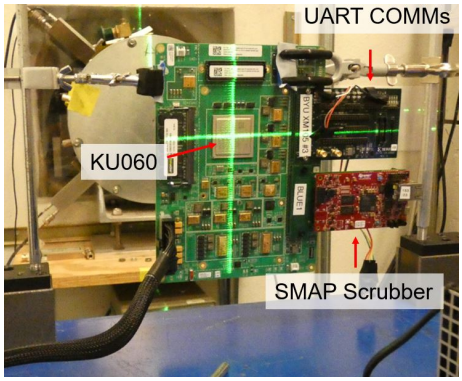

Fig. 3. LANSCE Radiation Test

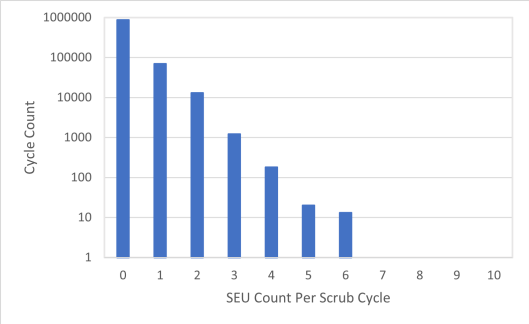

Fig. 4. Distribution of CRAM Upsets in Scrubbing Cycles.

The continuous scrubbing 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 of the individual CRAM frame with minimal overhead. The average time for each scrub cycle is 500 milliseconds with a 40 MHz clock. The actual number of upsets varies from scrub cycle to scrub cycle as demonstrated in Figure 4.

Out of the observed CRAM upsets, 83% of the non-zero scrub cycles contained one upset, 15% of these cycles had two upsets, and less than 3% contained three or more upsets. The high bandwidth of the SelectMap scrubber reduces the accumulation of sequential SEUs. In previous work, the JTAG scrubber observed 3 or more upsets CRAM upsets within 77% of the scrub cycles [7]. The SelectMap scrubber is a repair mechanism that is able to keep up with the accelerated rate induced during testing. This reduces accumulation-based multi-bit upsets and provides a more realistic repair window to devices under a normal radiation load.

B. Radiation Testing

During the radiation test, the processors continuously executed the CoreMark benchmark. For each test iteration of this benchmark, a checksum was computed to identify processor failures, and a test iteration count was reported to the host

TABLE III
NEUTRON RADIATION TEST DATA

Design	Processors (LUT %)	Fluence (n/cm^2)	Observed CRAM Upsets	Failures	Cross Section (cm^2)	+95% Confidence -95% Confidence	Reduction of σ_{pp}	Improvement of MWBF
MicroBlaze Unmitigated	264 (88.09%)	3.01×10^{9}	1492	265	8.80×10^{-8}	9.88×10^{-8} 7.72×10^{-8}	1×	1×
MicroBlaze TMR	60 (93.71%)	4.97×10^{10}	20010	26	5.24×10^{-10}	7.65×10^{-10} 3.42×10^{-10}	38.73×	30.60×
PicoRV32 Unmitigated	265 (82.46%)	5.09×10^{9}	2485	409	8.10×10^{-8}	8.90×10^{-8} 7.30×10^{-8}	1×	1×
PicoRV32 TMR	68 (97.19%)	4.37×10^{10}	17983	31	7.10×10^{-10}	1.01×10^{-9} 4.81×10^{-10}	29.85×	28.06×
Taiga Unmitigated	135 (93.29%)	3.65×10^{9}	1747	239	6.55×10^{-8}	7.40×10^{-8} 5.70×10^{-8}	1×	1×
Taiga TMR	31 (93.97%)	3.08×10^{10}	12382	22	7.13×10^{-10}	1.08×10^{-9} 4.47×10^{-10}	20.31×	18.48×
VexRiscv Unmitigated	117 (96.38%)	7.08×10^{9}	3473	422	5.96×10^{-8}	6.54×10^{-8} 5.38×10^{-8}	1×	1×
VexRiscv TMR	28 (93.51%)	5.38×10^{10}	21783	30	5.58×10^{-10}	7.96×10^{-10} 3.75×10^{-10}	25.05×	23.29×
NOEL-V Unmitigated	33 (95.86%)	5.35×10^{9}	2266	138	2.58×10^{-8}	3.02×10^{-8} 2.14×10^{-8}	1×	1×
NOEL-V TMR	8 (96.51%)	4.14×10^{10}	17196	4	9.67×10^{-11}	2.47×10^{-10} 2.42×10^{-11}	75.99×	61.55×

computer over a serial link. The processor was considered to have failed if it reported the wrong checksum and test iteration combination. The computer recording the telemetry compared the results with a golden value and the results from the other processors on the design.

The effectiveness of TMR is measured as the reduction in the neutron cross section, which is the ratio between the failures and the total fluence. The radiation test results in Table III show processor count per design, the total fluence each design was exposed to, and the number of observed CRAM upsets. The estimated cross section for each design (consisting of multiple tiled processors) is reported with 95% confidence intervals [35]. The TMR designs were exposed to about 10× greater fluence to provide adequate statistics. After a 10% threshold of processors failed, the JCM reconfigured the FPGA to continue testing. The threshold was used to collect more data and minimize down time for reconfiguration. The number of correctly operating processors was considered when calculating the neutron cross section per processor (σ_{pp}). These cross sections were evaluated with the recorded fluence and active processors between each failure event (n),

$$\sigma_{pp} = \frac{Failures}{\sum_{i=1}^{n}(Fluence_i \cdot Active\ Processors_i)}. \quad (1)$$

The σ_{pp} results are reported with other metrics for comparing the individual processors in Table V of Section VI.

The right two columns of Table III report the reduction of σ_{pp} and improvement of mean work between failures (MWBF) between the unmitigated and TMR implementations for each processor. To provide results independent of a specific environment, the MWBF is computed as normalized (see subsection VI-B). Each of the TMR soft processors achieved at least a 20× reduction per processor in the neutron cross section of system failures. The improvement in MWBF accounts for the decreased F_{max} and corresponding performance reduction of the TMR variants. The TMR soft processors achieved at least 18× improvement of MWBF. The NOEL-V achieved the greatest improvement of MWBF at 61.55×.

V. POST-IRRADIATION TESTING

Following the radiation test, CRAM fault injection was performed to compare results and provide confidence in our test methodology. Fault injection is used to emulate CRAM upsets within an SRAM-based FPGA [39]. By interfacing with the Xilinx FPGA configuration manager, CRAM frames can be read, modified, and written back to the device during operation. After the emulated upset, the processors can be tested to verify operation. Fault injection is only capable of emulating the CRAM SEU subset of the possible SEEs and does not take into account any non-CRAM SEUs, single event transients (SETs), or single event functional interrupts (SEFIs).

The JCM performed JTAG fault injection of random CRAM bits on the same experimental designs. In order to avoid accumulating faults, this fault campaign repaired faults and validated processors between injections. Instead of waiting for a 10% threshold of processors to fail, the FPGA was reconfigured as soon as one processor failure was detected.

The fault injection results in Table IV show processor count per design, the total injections each design was exposed to, and the number of observed processor failures. The CRAM

TABLE IV
RANDOM FAULT INJECTION TEST DATA

Design	Processor Count	Injections	Failures	Sensitive CRAM bits	Coefficient of Variance	Failure per Observed CRAM Neutron SEU	Percent Difference
MicroBlaze Unmitigated	264	24738	3719	15.03%	0.015	17.76%	- 15.36%
MicroBlaze TMR	60	101715	23	0.02%	0.208	0.13%	- 82.60%
PicoRV32 Unmitigated	265	25073	3300	13.16%	0.016	16.46%	- 20.03%
PicoRV32 TMR	68	32511	44	0.14%	0.151	0.17%	- 21.49%
Taiga Unmitigated	135	30883	6787	21.98%	0.011	13.68%	+ 60.64%
Taiga TMR	31	76738	12	0.016%	0.289	0.18%	- 92.93%
VexRiscv Unmitigated	117	10151	559	5.51%	0.041	12.15%	- 54.68%
VexRiscv TMR	28	31841	25	0.079%	0.200	0.14%	- 42.99%
NOEL-V Unmitigated	33	19931	524	2.63%	0.043	6.09%	- 56.83%
NOEL-V TMR	8	108111	4	0.004%	0.500	0.02%	- 84.09%

sensitivity r was computed as the ratio of failures to the total injections ($r = k/n$), where k equals the number of failures and n equals the number of faults injected. The standard deviation of the CRAM sensitivity is calculated as

$$\sigma = \sqrt{\frac{k}{n^2}\left(1 - \frac{k}{n}\right)} \tag{2}$$

[39]. The coefficient of variance of the fault injection results demonstrates the variance in the available data in relation to the population mean, *Coefficient of variance* $= \frac{\sigma}{r}$. The lower this metric is, the more confidence in a results of the data set [40]. As the CRAM sensitivity decreases, more injections are required to meet the same level of confidence.

In order to compare the random fault injection results against the neutron radiation data, the processor failure per observed CRAM upset of the radiation test is reported as a comparison against the CRAM sensitivity of the fault injection. This does not provide a perfect comparison as the fault injection can only emulate the CRAM SEU subset of all the possible SEEs induced during the neutron radiation test. The fault injection results are similar to the radiation results, with the percent differences below 100% and with low coefficients of variance. Only one of the designs has a positive difference, suggesting that there were more CRAM single point failures in the design that were not fully observed by the limited accelerated radiation testing.

VI. SOFT PROCESSOR COMPARISON

The right two columns in Table III of Section IV summarize the effectiveness of the TMR mitigation for each processor. In this section, all of the processors are compared against a single reference, the unmitigated MicroBlaze benchmark design, for the metrics of neutron cross section, performance, and power. This section also introduces two composite metrics that combine neutron cross section, performance, and power. These primary metrics and the composite metrics are shown in Table V for all processors (unmitigated and mitigated).

A. Primary Metrics

The first primary metric is the observed neutron cross section per processor (σ_{pp}) of the processor failures. The σ_{pp} has been calculated for the individual processors using Equation 1 for both the mitigated and unmitigated variations and the reduction is normalized against the σ_{pp} of the unmitigated MicroBlaze processor. As expected, the cross sections of the unmitigated processors show an direct relationship to their LUT utilization. The cross section of the TMR processors does not show this same relationship between cross section and LUT utilization. The smallest processor, the PicoRV32 achieved the greatest normalized σ_{pp} reduction among the unmitigated processors at 1.10×. The MicroBlaze achieved the greatest normalized σ_{pp} reduction among the TMR processors at 38.73× the baseline processor.

The second primary metric is raw performance and is computed as the product of the benchmark cycle count and F_{max}. The Taiga processor showed the greatest performance among the unmitigated and TMR processors, 2.95× and 2.68× respectively. The decreased F_{max} of the TMR mitigation results in the lower performance of the TMR processors.

The third primary metric is dynamic power consumption of the processor. This value is collected from the vendor power reports of the implemented designs. The power consumption

TABLE V
NEUTRON RADIATION DATA PER PROCESSOR

Design	σ_{pp} (cm^2)	+95% Confidence -95% Confidence	Normalized σ_{pp} Reduction	Normalized Performance	Normalized Power (W)	Normalized MWBF	Norm. MWBF Per Watt
MicroBlaze TMR	8.40×10^{-12}	1.27×10^{-11} 5.49×10^{-12}	38.73×	0.79×	5.62×	30.56×	5.44×
PicoRV32 TMR	9.90×10^{-12}	1.40×10^{-11} 6.71×10^{-12}	32.85×	0.48×	3.65×	15.70×	4.30×
Taiga TMR	2.36×10^{-11}	3.55×10^{-11} 1.48×10^{-11}	13.81×	2.68×	11.78×	37.03×	3.14×
VexRiscv TMR	2.05×10^{-11}	2.93×10^{-11} 1.38×10^{-11}	15.85×	1.88×	10.86×	39.74×	2.74×
NOEL-V TMR	1.27×10^{-11}	3.22×10^{-11} 3.16×10^{-12}	25.70×	0.20×	20.51×	5.02×	0.24×
MicroBlaze Unmitigated	3.25×10^{-10}	3.65×10^{-10} 2.85×10^{-10}	1.00×	1.00×	1.00×	1.00×	1.00×
PicoRV32 Unmitigated	2.95×10^{-10}	3.25×10^{-10} 2.66×10^{-10}	1.10×	0.51×	1.09×	0.56×	0.51×
Taiga Unmitigated	4.78×10^{-10}	5.40×10^{-10} 4.16×10^{-10}	0.68×	2.95×	2.11×	2.00×	0.95×
VexRiscv Unmitigated	5.14×10^{-10}	5.64×10^{-10} 4.64×10^{-10}	0.63×	2.01×	2.36×	1.27×	0.54×
NOEL-V Unmitigated	9.61×10^{-10}	1.13×10^{-9} 7.98×10^{-10}	0.34×	0.24×	6.14×	0.08×	0.01×

of the processors in general shows a direct relationship to their LUT utilizations. The TMR PicoRV32 required less LUT overhead for voter insertion and resulted in the most power efficient TMR processor.

B. Composite Metrics

Using the product of the normalized σ_{pp} reduction and performance, the normalized MWBF can be computed for each processor. This metric accounts for both the benefit and cost of the TMR mitigation. The unmitigated Taiga Processor achieved the best MWBF among the unmitigated processors. The TMR VexRiscv Processor achieved the best MWBF among the TMR processors. Even though the TMR MicroBlaze achieved the smallest σ_{pp}, due to the lack of performance, the Taiga and VexRiscv were able to achieve a greater MWBF.

The power efficiency of a processor is critical within a power limited system. The normalized MWBF per watt represents design and power efficiency in the context of MWBF. The unmitigated and TMR MicroBlaze achieved the greatest efficiency among both unmitigated and TMR processors, showing the maturity and optimization of the implementation.

C. Analysis

There are potentially different causes for the variation in the results among the unmitigated and TMR processors. The CoreMark benchmark may not fully exercise and validate all the functionality included with the processor. This may have masked failures in the more complex processors such as the NOEL-V. The Taiga, VexRiscv, and NOEL-V processors used additional FPGA BRAM and DPS resources. Though these experimental designs triplicated all DSPs and BRAM primitives, there was not any monitoring or additional mitigation (i.e., scrubbing) for the SEUs occurring in these primitives.

These processors implement different pipeline architectures in executing the software. With the lack of inspection for the sensitive CRAM bits and the related component of the digital design, it is difficult to identify any relationship between single point failures and the implemented architectures. The vendor implementation tools may also introduce single point failures in the placement and routing of the TMR processors. The default implementation settings were used for these experiments, but additional placement and routing constraints could be used to provide better results specific to the FPGA architecture. Additional testing of varied implementations with additional inspection tools for SEU monitoring and analysis is required to identify the contributing factors for single point failures.

VII. CONCLUSION

This paper explored the effects of TMR across different soft processors in a neutron radiation environment. The TMR implementations within the experimental designs achieved up to a 75× improvement in reliability at the cost of potentially 4.8× resource utilization and decreased F_{max} by 12.4% average. Future work will identify the possible causes for single point failures and exploration of additional mitigation. The experimental designs could explore variations in processor configuration, operation frequency, and different strategies for placement and routing. Additional techniques and tools could collect detailed data for single point failures such as radiation testing with BRAM monitoring, and targeted fault injection with detailed bitstream analysis of sensitive CRAM bits.

REFERENCES

[1] P. Graham, M. Caffrey, J. Zimmerman, D. Eric Johnson, P. Sundararajan, and C. Patterson, "Consequences and categories of SRAM FPGA configuration SEUs," *Proc. 5th Annu. Int. Conf. Military Aerosp. Program. Logic Devices*, 01 2003.

[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 *Proceedings of the 2008 International Conference on Engineering of Reconfigurable Systems & Algorithms, ERSA 2008, Las Vegas, Nevada, USA, July 14-17, 2008*, T. P. Plaks, Ed. CSREA Press, 2008, pp. 139–145.

[3] Y. Ichinomiya, 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 *2010 18th IEEE Annual International Symposium on Field-Programmable Custom Computing Machines*, May 2010, pp. 47–54.

[4] M. Berg, C. Poivey, D. Petrick, D. Espinosa, A. Lesea, K. LaBel, M. Friendlich, H. Kim, and A. Phan, "Effectiveness of internal vs. external SEU scrubbing mitigation strategies in a xilinx FPGA: Design, test, and analysis," in *2007 9th European Conference on Radiation and Its Effects on Components and Systems*, 2007, pp. 459–466.

[5] A. E. Wilson, C. Thurlow, and M. Wirthlin, "Fault injection testing of fault tolerant RISC-V soft processors on Xilinx SRAM-based FPGAs," *Journal of Radiation Effects Research and Engineering (JRERE)*, vol. 39, pp. 317–322, April 2021.

[6] A. E. Wilson and M. Wirthlin, "Neutron radiation testing of fault tolerant RISC-V soft processor on Xilinx SRAM-based FPGAs," in *2019 IEEE Space Computing Conference (SCC)*, July 2019, pp. 25–32.

[7] A. E. Wilson, S. Larsen, C. Wilson, C. Thurlow, and M. Wirthlin, "Neutron radiation testing of a TMR Vexriscv soft processor on SRAM-based FPGAs," *IEEE Transactions on Nuclear Science*, vol. 68, no. 5, pp. 1054–1060, 2021.

[8] A. E. Wilson and M. Wirthlin, "Fault injection of TMR open source RISC-V processors using dynamic partial reconfiguration on SRAM-based FPGAs," in *IEEE Space Computing Conference (SCC)*, 2021, pp. 1–8.

[9] D. Skouson, A. Keller, and M. Wirthlin, "Netlist Analysis and Transformations Using SpyDrNet," in *Proceedings of the 19th Python in Science Conference*, Meghann Agarwal, Chris Calloway, Dillon Niederhut, and David Shupe, Eds., 2020, pp. 40 – 47.

[10] R.-V. Foundation. (2022) RISC-V. Accessed: 6-July-2022. [Online]. Available: https://riscv.org/

[11] C. Heinz, Y. Lavan, J. Hofmann, and A. Koch, "A catalog and in-hardware evaluation of open-source drop-in compatible RISC-V softcore processors," in *2019 International Conference on ReConFigurable Computing and FPGAs (ReConFig)*, 2019, pp. 1–8.

[12] Antmicro. (2022) embench-tester. Accessed: 14-Nov-2021. [Online]. Available: https://antmicro.github.io/embench-tester/

[13] J. M. Johnson and M. J. Wirthlin, "Voter insertion algorithms for FPGA designs using triple modular redundancy," in *Proceedings of the 18th Annual ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, ser. FPGA '10. New York, NY, USA: ACM, 2010, pp. 249–258.

[14] J. Heiner, B. Sellers, M. Wirthlin, and J. Kalb, "FPGA partial reconfiguration via configuration scrubbing," in *2009 International Conference on Field Programmable Logic and Applications*, Aug 2009, pp. 99–104.

[15] M. J. Wirthlin, A. M. Keller, C. McCloskey, P. Ridd, D. Lee, and J. Draper, "SEU mitigation and validation of the LEON3 soft processor using triple modular redundancy for space processing," in *Proceedings of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, ser. FPGA '16. New York, NY, USA: ACM, 2016, pp. 205–214.

[16] A. Lindoso, L. Entrena, M. García-Valderas, and L. Parra, "A hybrid fault-tolerant LEON3 soft core processor implemented in low-end SRAM FPGA," *IEEE Transactions on Nuclear Science*, vol. 64, no. 1, pp. 374–381, Jan 2017.

[17] M. Psarakis, A. Vavousis, C. Bolchini, and A. Miele, "Design and implementation of a self-healing processor on SRAM-based FPGAs," in *2014 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*, Oct 2014, pp. 165–170.

[18] A. M. Keller and M. J. Wirthlin, "Benefits of complementary SEU mitigation for the LEON3 soft processor on SRAM-based FPGAs," *IEEE Transactions on Nuclear Science*, vol. 64, no. 1, pp. 519–528, Jan 2017.

[19] N. H. Rollins, "Hardware and software fault-tolerance of softcore processors implemented in SRAM-based FPGAs," Ph.D. dissertation, Brigham Young University, Provo, UT, USA, 2012, aAI3506158.

[20] A. Ramos, J. A. Maestro, and P. Reviriego, "Characterizing a RISC-V SRAM-based FPGA implementation against Single Event Upsets using fault injection," *Microelectronics Reliability*, vol. 78, pp. 205–211, 2017.

[21] L. A. Aranda, N. J. Wessman, L. Santos, A. Sánchez-Macián, J. Andersson, R. Weigand, and J. A. Maestro, "Analysis of the critical bits of a RISC-V processor implemented in an SRAM-based FPGA for space applications," *Electronics*, vol. 9, no. 1: 175, 2020.

[22] L. A. C. Benites and F. L. Kastensmidt, "Automated design flow for applying triple modular redundancy (TMR) in complex digital circuits," in *2018 IEEE 19th Latin-American Test Symposium (LATS)*, 2018, pp. 230–233.

[23] A. B. de Oliveira, L. A. Tambara, F. Benevenuti, L. A. C. Benites, N. Added, V. A. P. Aguiar, N. H. Medina, M. A. G. Silveira, and F. L. Kastensmidt, "Evaluating soft core RISC-V processor in SRAM-based FPGA under radiation effects," *IEEE Transactions on Nuclear Science*, vol. 67, no. 7, pp. 1503–1510, 2020.

[24] F. Minnella, M. Pavis, and S. Bufalino, "Protection and characterization of an open source soft core against radiation effects," no. April, 2018. [Online]. Available: http://cds.cern.ch/record/2313417

[25] A. Gruwell, P. Zabriskie, and M. Wirthlin, "High-speed FPGA configuration and testing through JTAG," in *2016 IEEE AUTOTESTCON*, Sep. 2016, pp. 218–225.

[26] N. A. Harward, M. R. Gardiner, L. W. Hsiao, and M. J. Wirthlin, "Estimating soft processor soft error sensitivity through fault injection," in *2015 IEEE 23rd Annual International Symposium on Field-Programmable Custom Computing Machines*, 2015, pp. 143–150.

[27] *UG116 Device Reliablity Report*, Xilinx, 6 2022.

[28] Xilinx, "Microblaze processor reference guide," v2019.1. [Online]. Available: https://docs.xilinx.com/v/u/2019.1-English/ug984-vivado-microblaze-ref

[29] C. Wolf. (2020) Picorv32. YosysHQ. Accessed: 27-April-2020. [Online]. Available: https://github.com/cliffordwolf/picorv32

[30] ——. (2021) riscv-formal. SymbioticEDA. Accessed: 26-Nov-2021. [Online]. Available: https://github.com/SymbioticEDA/riscv-formal

[31] 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 *2018 IEEE 26th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, April 2018, pp. 1–8.

[32] C. Papon. (2021) Vexriscv. SpinalHDL. Accessed: 1-Feb-2021. [Online]. Available: https://github.com/SpinalHDL/VexRiscv

[33] F. Kermarrec. (2020) Linux on LiteX VexRiscv. Accessed: 3-Feb-2020. [Online]. Available: https://github.com/litex-hub/linux-on-litex-vexriscv

[34] J. Andersson, "Development of a NOEL-V RISC-V SoC targeting space applications," in *2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, 2020, pp. 66–67.

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

[36] F. Libano, B. Wilson, J. Anderson, M. J. Wirthlin, C. Cazzaniga, C. Frost, and P. Rech, "Selective hardening for neural networks in fpgas," *IEEE Transactions on Nuclear Science*, vol. 66, no. 1, pp. 216–222, 2019.

[37] 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 *2014 IEEE Radiation Effects Data Workshop (REDW)*, 2014, pp. 241–245.

[38] P. W. Lisowski, C. D. Bowman, G. J. Russell, and S. A. Wender, "The los alamos national laboratory spallation neutron sources," *Nuclear Science and Engineering*, vol. 106, no. 2, pp. 208–218, 1990. [Online]. Available: https://doi.org/10.13182/NSE90-A27471

[39] C. Thurlow, H. Rowberry, and M. Wirthlin, "Turtle: A low-cost fault injection platform for SRAM-based FPGAs," in *2019 International Conference on ReConFigurable Computing and FPGAs (ReConFig)*, Dec 2019, pp. 238–245.

[40] H. Quinn and M. Wirthlin, "Validation techniques for fault emulation of sram-based fpgas," *IEEE Transactions on Nuclear Science*, vol. 62, no. 4, pp. 1487–1500, 2015.