

Systematic Evaluation of Critical Configuration Memory Bits In FPGAs Via SEM-Based Fault Injection

Aiden Christy
Montana State University
Bozeman, MT, USA
aidenrchristy03@gmail.com

Benjamin Macht
Montana State University
Bozeman, MT, USA
benjamin.macht@student.montana.edu

Hezekiah Austin
Resilient Computing, LLC
Bozeman, MT, USA
hezekiah.austin@resilient-computing.com

Chris Major
Resilient Computing, LLC
Bozeman, MT, USA
chris.major@resilient-computing.com

Garrett Perkins
Montana State University
Bozeman, MT, USA
garrett.perkins1@student.montana.edu

Kris Allick
Montana State University
Bozeman, MT, USA
kristoffer.allick@student.montana.edu

Brock LaMeres
Montana State University
Bozeman, MT, USA
lameres@montana.edu

Abstract—In aerospace computing, field programmable gate arrays (FPGAs) are widely used as processing nodes. These applications require ground-based reliability testing to simulate radiation induced faults. Configuration memory error injection can be used to simulate single event upsets (SEUs), also known as soft errors. The Xilinx Soft Error Mitigation (SEM) controller provides error correction and classification for SEUs; however, not all faults are correctable by the SEM. This research systematically injects errors into each of the 2,274,483 essential bit addresses reported by Vivado on the Resilient Computing RadPC-SBC-101 Single Board Computer (SBC). RadPC is a fault tolerant RISC-V computer architecture implemented on an AMD Artix-7 FPGA used as an example design to determine which addresses can be classified as critical. Of the essential bits tested, only 0.065% produced critical responses, suggesting that Vivado’s essential bit report significantly overestimates the set of truly critical bits. Critical bits were found to cluster in five distinct regions of the address space rather than distributing uniformly. Additionally, a subset of critical bits was identified that persists through FPGA reconfiguration and requires a full power cycle to resolve, representing the most severe fault class observed. These findings establish a foundation for improving SEM-based fault recovery and optimizing pre-flight radiation effects simulation in aerospace computing systems.

Contents

I	Introduction	1
II	Research	3
III	Results	4
IV	Conclusion	6
V	Further Research	6
VI	Acknowledgments	6
	References	6

Biographies

Index Terms—FPGA, SEM, SEE, SEU, RadPC
I. INTRODUCTION

The configuration memory of a field programmable gate array (FPGA) sets the connections between logic elements in the FPGA fabric. The critical bits in an FPGA’s configuration memory are the necessary bits for the logic operations implemented in the design. Xilinx’s Vivado FPGA design suite generates a report of essential bits identified in a design corresponding to physical bits in the FPGA’s configuration memory. Depending on the design, these bits may or may not be essential. In this report, we define **essential bits** to be bits on the list generated by Vivado, distinct from **critical bits** defined as bits that are vital to the operation of the system. Identifying the validity of the information from the Vivado report in a soft error mitigation (SEM) error correction system could drastically improve radiation fault recovery performance and radiation resistance [1]. This research aims to begin verifying the criticality of essential and non-essential bits reported by Vivado. Verified results can be utilized in SEM conditioning and data driven error recovery. Finding the critical bit’s locations can improve the SEM’s ability to fix faults in the FPGA’s configuration memory by allowing it to focus only on the regions that would cause uncorrectable operational errors if struck by radiation.

A. Radiation Induced Faults

Small satellites operating in orbit around Earth do not benefit from the protective qualities of the atmosphere or magnetosphere of Earth with respect to radiation [2]. Satellite space computers are subject to adverse radiation that can

induce negative effects on operation. Radiation effects are grouped into two failure types, Total Ionizing Dose (TID) and Single Event Effects (SEE) [3]. These radiation effects are caused by ionizing radiation that encounters and disrupts the integrated circuit (IC) substrate and injects unwanted energy into the circuitry. TID is primarily caused by low energy protons and electrons ($< 30 \text{ MeV/amu}$) striking the substrate and creating extraneous electron/hole pairs that get trapped in circuit materials [4]. Should a transistor's gate oxide trap the excess charge, the threshold voltage can change, resulting in a constant on or off state in the device. If the charge is trapped in the isolation region between devices, it can result in power consumption through current leakage, burning out the device. Due to fabrication advances in the semiconductor industry that seek to shrink the feature sizes of transistors, commercial ICs are becoming more naturally resistant to TID [5]–[8]. COTS nodes ($< 65\text{nm}$) today have increasingly high levels of resistance to TID ($> 30\text{krad}$) as a result of small insulation regions mitigating the risk of charge trapping [7], [9]. Figure 1 illustrates both scenarios of TID damage. SEEs are caused by heavy ions and high energy particles colliding with the diffusion regions of a device, as illustrated in Figure 1. Erroneous logic level transitions are caused by the resulting electron/hole pairs caused by SEEs. Changes in the systems logic levels cause failure when digital logic lines transition, causing single event transient (SET). When such changes are stored, single event upsets (SEU) occur. When an SEU cannot be corrected, the system faults or has a single event functional interrupt (SEFI) occurred. If the system cannot correct the fault, the entire design must be power cycled and undergo a full system re-initialization, or reconfiguration.

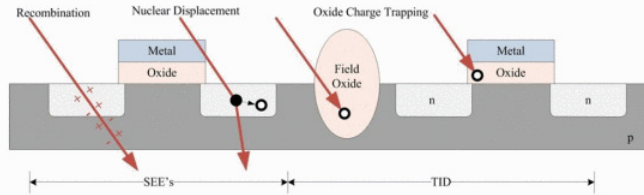


Fig. 1: TID and SEE in CMOS Cross Section [9]

B. Single Event Upsets

SEEs can generally be divided into three major types based on the effects they have on the device. The types are single-event transients (SETs), single-event upsets (SEUs), and single-event latch-up (SEL). The type of error utilized in this investigation is single-event upsets. SEUs are errors that occur in digital circuits that have memory like qualities. When a flipped bit is latched in a flip-flop, the erroneous bit can remain in the system for many clock cycles. This flipped bit can then propagate through the system and cause errors such as incorrect states in finite state machines to be triggered. SEUs can originate from direct strikes on physical register locations or can result from SETs occurring in upstream combinational logic. In large SRAM memories, SEUs can cause multiple erroneous values to be stored. In this research, we focus on

SEUs in the configuration memory of an FPGA as the cause of a system breakdown that halts the operation.

C. RadPC

The design implementation selected for essential bit verification is the RadPC-SBC-101 single board computer from Resilient Computing, LLC. RadPC is a custom, reduced instruction set computer (RISC-V 32I) designed for radiation tolerance [2]. Figure 2 displays the architecture of a 4 core RadPC soft-core. RadPC is a logic design developed in the VHSIC Hardware Description Language (VHDL). RadPC mitigates SEEs on space computers by leveraging redundancy in its architectural approach. A modification of the triple modular redundancy (TMR) strategy, known as quad modular redundancy (QMR), protects the central processing unit (CPU). This computer implements an RISC-V architecture on an FPGA with a QMR microcontroller [10], [11]. When a fault is detected, the faulted CPU is reloaded with the correct internal register values. Should an individual CPU experience complete failure, it will be partially reconfigured followed by register reloading and system synchronization [12], [13]. In coordination with these methods, memory scrubbers add another layer of resistance to SEE faults [10], [12], [14]. Finally, an SEM is used within RadPC to fix SEUs in the FPGA configuration memory.

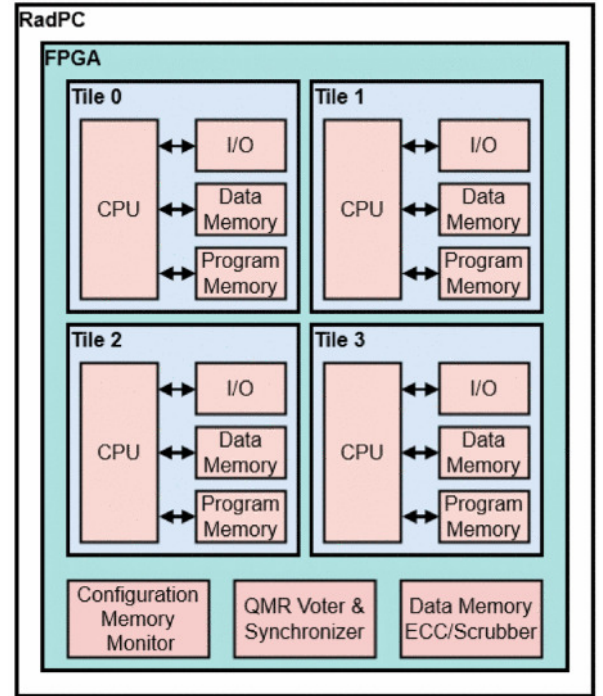


Fig. 2: RadPC 4 core computer architecture. [2]

D. Soft Error Mitigation (SEM) Controller

The radiation fault recovery controller utilized by the RadPC is the Xilinx Soft Error Mitigation core controller. The SEM uses a combined strategy of emergency and scheduled memory maintenance and running repairs to maintain the integrity of

the device's configuration memory [15]. The FPGA's configuration memory uses Error Correction Codes (ECCs) to monitor for faults. The SEM has the ability to monitor these ECCs in order to track the types of errors (single versus double bit errors) in addition to initiating repairs. Memory maintenance can also consist of a full reconfiguration of the FPGA configuration memory by resetting the device to its intended state, regardless of errors. These full reconfigurations can be initiated on a schedule or upon error detection from a device scan [15]. The SEM allows the device to repair errors while still in operation and does not require a device reconfiguration. The SEM uses a combination of these techniques to recover from ionizing radiation faults. One of the most beneficial features of the SEM is that the controller core allows users to communicate with the soft core via UART. This communication line allows many control commands, including error injections. By providing the soft core with a linear frame address (LFA) in the device's configuration memory, the controller will simulate an error at that address by toggling the corresponding bit. If a user wishes to inject an error in multiple bits, it can be done through a series of SEM commands [15].

E. Vivado's Essential Bit Report

Given that a design has an SEM core incorporated and essential bit reporting is enabled, Vivado provides two .eb* files containing the locations and values of the configuration memory bits used to hold the design [15]. While these bits are necessary for configuration, not all are needed for continued system operation. This can be because regions of the FPGA are not used by the design, but also because a bit within the FPGA may not have any impact on the logic operation of the circuit.

II. RESEARCH

The purpose of this research is to identify and categorize bits in configuration memory as critical to operation. This determination will allow for improved aerospace computing pre-flight testing and verification. Decreasing the volume of bits labeled as critical will streamline response testing and decrease the area of vulnerable bit locations on the processor.

A. Hardware Setup

The DUT in this research is a RadPC-SBC-101 single-board-computer. This SBC uses a QMR RISC-V soft processor on a Xilinx Artix-7 FPGA. It uses an SEM IP core with error injection and classification enabled. The SEM UART communication pins are routed to GPIO pins on the board. Two FTDI cables are connected to the board, one for the SEM UART ports, the other for configuration. The configuration FTDI cable is connected to the RadPC programming port. Figures 3 and 4 display the testing setup that connects the RadPC to a host machine that controls the injection. The host machine for the injection script is a windows machine running WSL2 but could also be done using a native Linux environment.

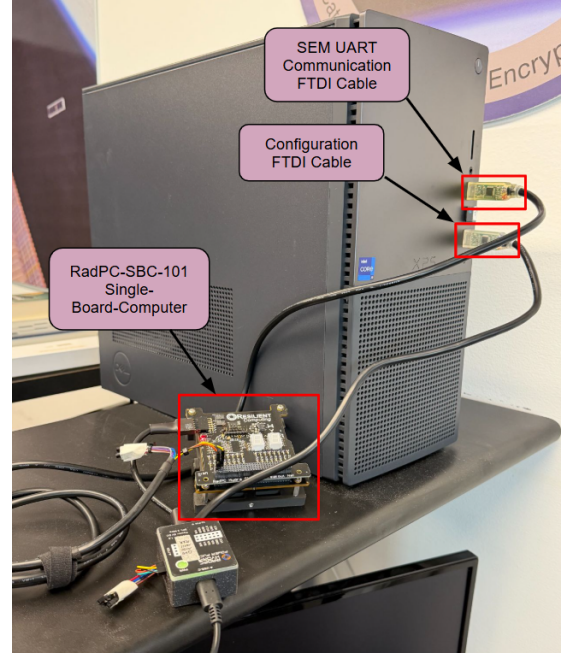


Fig. 3: Connections from host machine to RadPC-SBC-101.

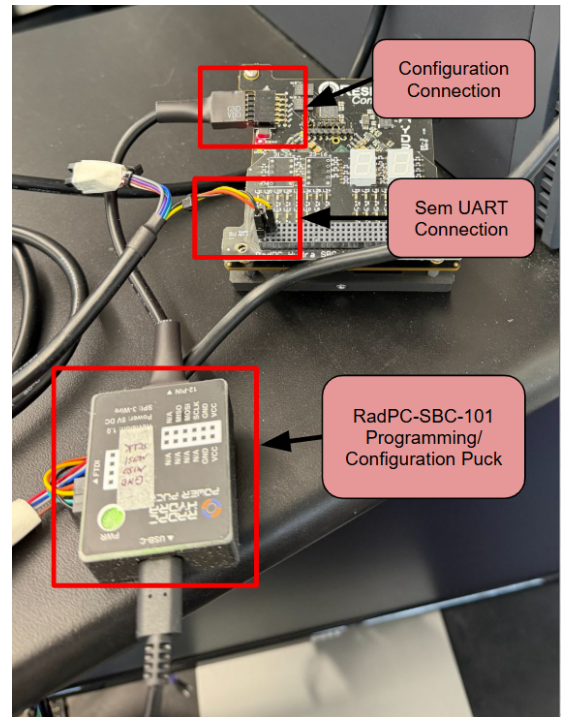


Fig. 4: Direct FTDI connections on RadPC-SBC-101.

B. Essential Bit List Extraction

Xilinx's Vivado Design Suite generates a file of essential bit data (.ebd) that was used to create a list of addresses. This was done by converting the physical frame address as defined in the ebd into linear frame addresses using a custom C code parser. This tool parses the .ebd file sequentially, extracting non zero bit positions and converting them into corresponding frames, words, and bits. These were then combined into a 40 bit address to be transmitted to the SEM formatted in the

way shown in Figure 5. Bits 29 and 30, denoted SS, refer to the hardware SLR number, in this case 00 due to RadPC being non-SSI [15]. The resulting list, stored in a plain-text file, was used as an input into the injection interface. For the RadPC design on an Artix-7 FPGA, the parsing tool generated 2,274,483 different addresses.

39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	0	0	S	S	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L

Fig. 5: Linear Frame Address Format. [15]

C. Injection Framework

A python based automated injection script was developed and used to systematically inject errors into each address. This script simultaneously injected the errors and recorded the corresponding response from the SEM core. It communicates with the FPGA using a UART interface using the pyserial library. The SEM injection protocol was the following: Enter injection mode, issue injection command with LFA, return to observation, and record injection and status report. An example status report of the SEM reaction after correcting a fault is shown in Figure 6. The framework operated autonomously across the full address space with three primary behaviors:

- **Normal injection:** The address was injected, the SEM response and status were recorded, and the framework advanced to the next address.
- **Recovery handling:** If the SEM did not respond to the injection command, the framework polled the SEM status for up to 60 seconds. If the core returned to Observation state within this window, the same address was retried. This condition was observed to occur transiently following FPGA reconfiguration and did not indicate a critical bit.
- **Critical bit detection and reboot:** If a critical error response was detected or if the SEM failed to recover within the 60-second polling window, the address was recorded as a critical bit, the FPGA was reconfigured and injection resumed 100 addresses prior to the critical address to ensure that no surrounding addresses were missed during the reconfiguration window. All previously identified critical addresses were skipped on the resumed sweep.

The results were recorded in a csv, marking the timestamp, LFA, and trigger type. Figure 7 shows the operation flow of the injection framework, featuring two selection menus and a status bar that is updated as injections are performed. It is important to note that this tool was constructed by the authors for more uses than critical bit identification, and hence for multiple menu options.

— Status Report (S) —	
Max Frame (linear count):	0x000049D7
Hardware SLR Number:	00
State:	Observation (SC 02)
Flags:	Correctable, Non-Essential (FC 00)
Feature Set:	Error Detection + Injection (FS 09)

Fig. 6: Error injection status report.

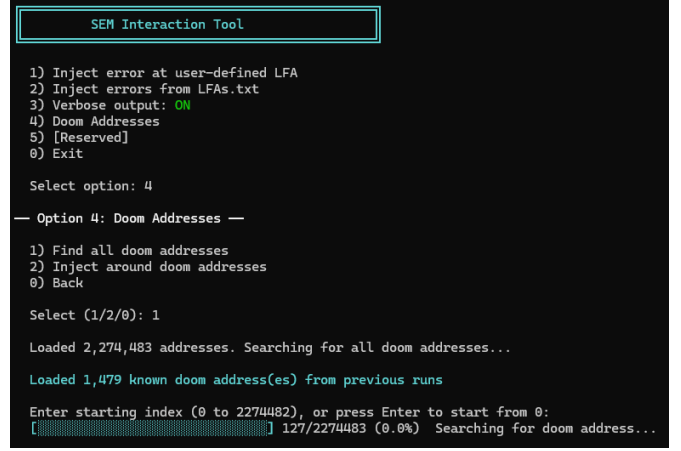


Fig. 7: Error injection script menu and operational displays.

D. Critical Error Categorization

Each injection response was compared with a list of SEM output triggers. If the response was categorized by any of the statuses outlined in Figure 8, it was recorded as a critical bit along with its response details. All response types, with the exception of INJECT-SKIPPED-NO-RECOVERY, are defined by the Xilinx SEM IP core specification [15]. The INJECT-SKIPPED-NO-RECOVERY condition was defined by the authors to describe instances where the SEM core became non-responsive to any commands and did not return to Observation state within a 60-second polling window, indicating an unrecoverable fault state requiring FPGA reconfiguration.

SEM OUTPUT TOKEN	CLASSIFICATION LABEL	TRIGGER ID
DED	Double-bit ECC — uncorrectable	DED
FC 60	Uncorrectable, essential frame	FC_60_Uncorrectable_Essential
FC 20	Uncorrectable, non-essential frame	FC_20_Uncorrectable_NonEssential
SC 1F	SEM core fatal error	SC_1F_FATAL
HLT	SEM core halted — unrecoverable	HLT
I> timeout	SEM non-responsive — no recovery within 60 s	INJECT_SKIPPED_NO_RECOVERY

Fig. 8: SEM Response and Definitions.

III. RESULTS

A. Critical Bit Identification

The data gathered in this investigation consist of a list of critical bits identified on the RadPC. These bits are categorized by type and location. Throughout the testing process, the authors also observed that there were many cases of the SEM not recognizing that an error had occurred in the configuration memory.

The list of critical bits found in the FPGA essential bit list highlights the amount of bits that, when faulted, result in a system crash that the SEM can recognize but not correct. This list is comprised of only a small subset of the essential bits reported by Vivado. This alludes to the fact that this list of critical bits is incomplete. This conclusion is drawn as a result of two findings; that the list of critical bits is far smaller than the list of essential bits, and that the SEM did not flag

an event on the error injection into some of the essential bit addresses.

The critical bits identified in the system made up only 0.065% of the total essential bits, with 1,479 critical addresses identified out of 2,274,483 essential bit addresses tested. This outcome is much smaller than the expected outcome for this research. Because of this, it is believed that the list of critical bits is not comprehensive. One likely explanation for this is that the SEM reported only critical errors in its own configuration. When an error was detected elsewhere, it would repair it, with the associated information being reported. However, this report does not contain information about the continued operation of the system. Due to this, it could be that the only critical bits that this research could identify were the critical bits that make up the fabric of the SEM core. Figure 9 displays the implementation of the RadPC design in the FPGA. The SEM core(blue) occupies a small fraction of the entire design(purple) in the fabric. Given that the SEM core appears to only classify bits as essential in its own fabric, the ratio of the SEM fabric to the system fabric provides a visual explanation for such a small set of critical bits. The relationship between the physical size of the SEM and the number of identified critical bits is consistent with the conclusion that the current methodology captures only bits within the SEM's own fabric as critical.

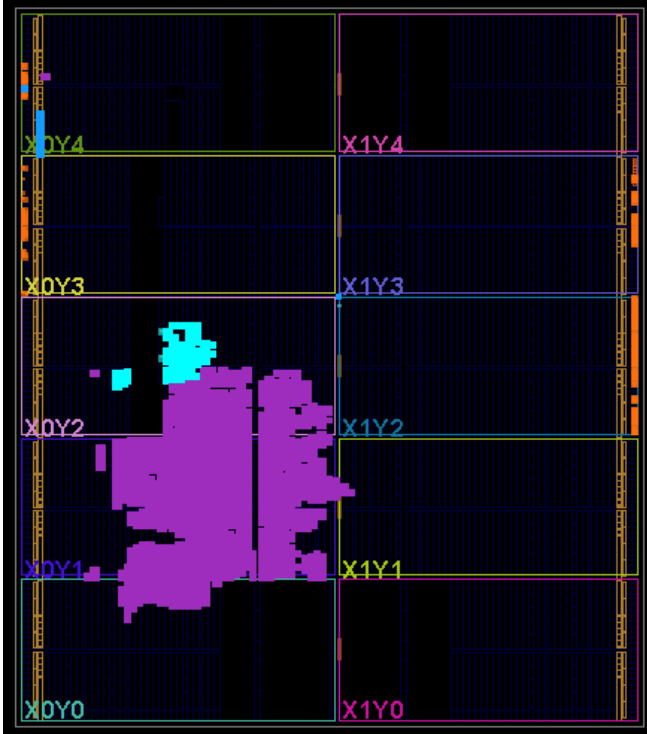


Fig. 9: Vivado implementation view of the RadPC Hydra Rev. A design. SEM core shown in blue, remaining design fabric in purple.

This idea is supported by the fact that the SEM was not recognizing error injections into essential bits. Although it could be possible that these bits were not actually necessary

for the operation of the design, it is unlikely. The more likely reason is that the SEM does not have mechanisms to recognize every single fault in the system. Given that this assumption is true, that would mean that the SEM missed a multitude of possible critical bits that should have been categorized.

Although the confirmed bits are critical to the system and their locations are important to note, they may not be the only critical bits in the system. This means that further investigation and research is necessary. This research would have the primary objective of validating the claim that the SEM missed critical bits.

B. Critical Bit Clustering

One primary observation of this investigation is the clustering of critical bits around certain LFAs. Figure 10 shows the distribution of critical bits across regions of linear frames. There were 5 main groupings of critical bits identified across the address space. These identified clusters had two main behaviors.

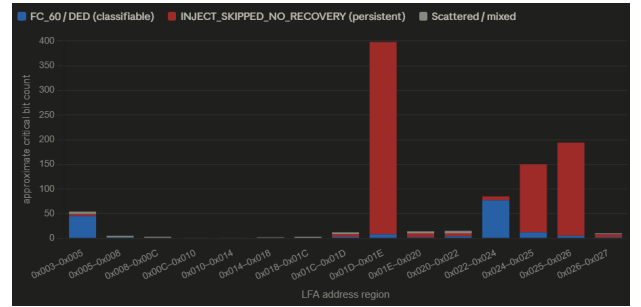


Fig. 10: Distribution of critical bits across the LFA address space of the RadPC Hydra Rev. A design.

The first is a dense collection of critical bits that caused uncorrectable SEM errors but were resolved by FPGA reconfiguration. These clusters did not cause problems with data collection and the injection resumed normally following reconfiguration.

The second is a dense cluster in which a single error of any classification was followed by exclusively INJECT-SKIPPED errors. Although there are several small cases of the design moving past this region of errors, there were also three large clusters of errors that would continue until the system was power cycled. These errors are likely to persist through reconfiguration.

Due to approximately two weeks of continuous testing, there were cases in which the authors were not observing the experiment at all times. This could have potentially led to inflated collections of critical bits in the large clusters, as a power cycle could not be issued promptly. As a result, it can be assumed, but not yet verified, that an earlier intervention would have resulted in lower cluster sizes.

C. Configuration-Persistent Errors

There were several critical bit addresses that behaved differently than others. When an error was injected into these, a cascading list of Injection-Skipped-No-Recovery errors would

result. However, if the first of these addresses were not faulted, the rest of the list would not be erroneous. It was assumed that a reconfiguration of the FPGA would remove any transient effects of the injected fault. After this discovery, that appears not to be the case.

It was found that if the FPGA were power cycled after the initial fault rather than reconfigured, cascading errors would not occur. This means that there are some critical bit locations that are much more important than others. This is because they cannot be fixed simply by reconfiguring the FPGA, the fault persists through the reconfiguration. Classifying persistent critical errors could not be done automatically by the injection python script. These errors appeared as an initial INJECT-SKIPPED error followed by immediate INJECT-SKIPPED errors at every injected address immediately after until the system underwent a power cycle.

IV. CONCLUSION

This research has demonstrated that the vast majority of the essential bits as reported by Vivado do not produce critical responses from the SEM. Critical bits are present and accessible in this list; however, in the scope of this experiment, the vast majority of essential bits are non-critical to the design's operation.

A. Critical Bit Categorization

Important bits can be categorized into one of three tiers of necessity to the system; non-critical essential, normal critical, and persistent critical. Non-critical essential bits are flagged by Vivado as essential but when faulted are either corrected by the SEM or ignored. These represent the majority of bits tested in this research.³ Normal critical bits produced an uncorrectable error when faulted that the SEM could classify. These damaged the SEM enough that it could not repair itself and required a reconfiguration to resume operation. Persistent critical bits completely crashed the design and resulted in continued errors after reconfiguration, thus requiring the FPGA to be power cycled. The persistent errors were the most rare but also most detrimental to the design's operation.

B. SEM Limitations

Due to the nature of this testing, the SEM seemed to only classify errors in its own fabric. The SEM core was the only metric utilized in identifying and classifying critical bits. A lack of a second defining metric like a regression test could have caused important bits to be overlooked. The SEM reports specifically on errors it can observe, however this does not mean the SEM can be used as an overarching error identification system. Therefore, this testing methodology may be incomplete. As a result, there may be critical bits outside of the identified critical bit list that are necessary for the operation of the RadPC FPGA but were not found.

C. Base Design Limitations

The DUT's base design also may have limited the scope of this testing. While critical bits were identified, this testing

was done on an unloaded RadPC, that is, a RadPC with no compiled operational code running on it. Since this was simply an architecture test, there will likely be an increased risk to systems with operational code on them. These errors will likely show up not as a critical bit error, but as an operational failure seen as incorrect operation or execution.

D. Data Use

The data gathered in this research will be used by the RadPC Research laboratory to improve board design and pre-flight testing. A refined critical bit list allows core redundancy designs to improve voter speed, increasing radiation tolerance. It will also scale down the scope of partial reconfiguration needed to resume system operation. The list of critical bits will also enable increased pre-flight testing capabilities by targeting weak spots on radiation tolerant designs.

V. FURTHER RESEARCH

The results of this experiment, while insightful, leave unknowns about critical bits in FPGAs. The classification of errors by the SEM core provides concrete locations of some critical bits in the design. However, it is believed that the identified list is not an exhaustive list; therefore, more tests are necessary. To fully determine whether an error affected the operation of the FPGA design, the full functionality of the system must also be tested. Using the RadPC Hydra board and a custom regression hat and test, a full functionality verification will be performed, verifying the operation of the system through each error injection. This research will continue to bridge the gap between the SEM response testing and a complete system critical bit identification and verification.

Additionally, more research will be done into the persistent critical bits to determine which error specifically caused the cascading errors, this will be done using similar methods as the regression testing. However, testing will be more targeted at specific locations in the FPGA fabric. This research will be used to locate critical bits that require a full power cycle after injection and to determine the response of the system after it is struck.

VI. ACKNOWLEDGMENTS

The authors thank the Montana State University RadPC Research Laboratory for use of facilities, equipment, funding, and expertise. The authors thank Resilient Computing for providing advisement and personnel for this project through their partnership with MSU.

REFERENCES

- [1] R. Le, "Soft error mitigation using prioritized essential bits," *Xilinx XAPP538 (v1.0)*, p. 72, 2012.
- [2] H. A. Austin, C. Major, Z. Becker, T. R. Crane, K. Allick, and B. J. LaMeres, "Dynamically reconfigurable coprocessor for floating-point arithmetic capability in small satellites," in *2025 IEEE Aerospace Conference*, 2025, pp. 1–8.
- [3] A. Holmes-Siedle and L. Adams, *Handbook of radiation effects*. Oxford university press, 2002.
- [4] H. Hughes and J. Benedetto, "Radiation effects and hardening of mos technology: Devices and circuits," *IEEE Transactions on Nuclear Science*, vol. 50, no. 3, pp. 500–521, 2003.

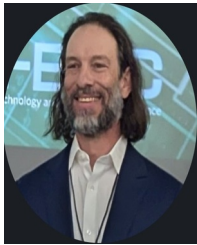
- [5] H. J. Barnaby, "Total-ionizing-dose effects in modern cmos technologies," *IEEE transactions on nuclear science*, vol. 53, no. 6, pp. 3103–3121, 2006.
- [6] R. Liu, A. Evans, L. Chen, Y. Li, M. Glorieux, R. Wong, S.-J. Wen, J. Cunha, L. Summerer, and V. Ferlet-Cavrois, "Single event transient and tid study in 28 nm utbb fdsoi technology," *IEEE Transactions on Nuclear Science*, vol. 64, no. 1, pp. 113–118, 2016.
- [7] C.-M. Zhang, F. Jazaeri, A. Pezzotta, C. Bruschini, G. Borghello, S. Mattiazzo, A. Baschiroto, and C. Enz, "Total ionizing dose effects on analog performance of 28 nm bulk mosfets," in *2017 47th European Solid-State Device Research Conference (ESSDERC)*. IEEE, 2017, pp. 30–33.
- [8] C.-M. Zhang, F. Jazaeri, A. Pezzotta, C. Bruschini, G. Borghello, F. Faccio, S. Mattiazzo, A. Baschiroto, and C. Enz, "Gigard total ionizing dose and post-irradiation effects on 28 nm bulk mosfets," in *2016 IEEE Nuclear Science Symposium, Medical Imaging Conference and Room-Temperature Semiconductor Detector Workshop (NSS/MIC/RTSD)*. IEEE, 2016, pp. 1–4.
- [9] R. Arora, Z. E. Fleetwood, E. X. Zhang, N. E. Lourenco, J. D. Cressler, D. M. Fleetwood, R. D. Schrimpf, A. K. Sutton, G. Freeman, and B. Greene, "Impact of technology scaling in sub-100 nm nmosfets on total-dose radiation response and hot-carrier reliability," *IEEE Transactions on Nuclear Science*, vol. 61, no. 3, pp. 1426–1432, 2014.
- [10] C. M. Major, A. Bachman, C. Barney, S. Tamke, and B. J. LaMeres, "Radpc: A novel single-event upset mitigation strategy for field programmable gate array-based space computing," *Journal of Aerospace Information Systems*, vol. 18, no. 5, pp. 280–288, 2021.
- [11] J. Williams, C. Barney, Z. Becker, J. Davis, C. Major, and B. LaMeres, "Radpc@ scale: A novel approach to the radpc single event upset mitigation strategy," in *2022 IEEE Aerospace Conference (AERO)*. IEEE, 2022, pp. 1–7.
- [12] B. J. LaMeres and C. Gauer, "Dynamic reconfigurable computing architecture for aerospace applications," in *2009 IEEE Aerospace conference*. IEEE, 2009, pp. 1–6.
- [13] C. Gauer, B. J. LaMeres, and D. Racek, "Spatial avoidance of hardware faults using fpga partial reconfiguration of tile-based soft processors," in *2010 IEEE Aerospace Conference*. IEEE, 2010, pp. 1–11.
- [14] J. S. Hane, B. J. LaMeres, T. Kaiser, R. Weber, and T. Buerkle, "Increasing radiation tolerance of field-programmable-gate-array-based computers through redundancy and environmental awareness," *Journal of Aerospace Information Systems*, vol. 11, no. 2, pp. 68–81, 2014.
- [15] AMD Xilinx, *Soft Error Mitigation Controller v4.1 LogiCORE IP Product Guide*, AMD Xilinx, 2023. [Online]. Available: https://docs.amd.com/r/en-US/pg036_s'em



Aiden Christy graduated with honors with a BS in Computer Engineering at Montana State University. He has spent the two years working in the Radiation Tolerant Research lab at MSU and spent three summers interning in industry. He is currently researching FPGA configuration and essential bit mapping utilizing a SEM controller on RadPC's hydra board.



Kris Allick is currently a Masters student in his first year of a M.S. degree in Electrical Engineering at Montana State University. He has spent each year of his degree working in aerospace labs, first with the MSU Balloon Outreach, Research, Exploration and Landscape Image System (Borealis), and currently as a graduate researcher with the Radiation Tolerant Computer Lab at MSU. From these experiences, he has developed a specialty in embedded systems design and testing.



Benjamin Macht received a B.S. with highest honors in Electrical Engineering from Montana State University in 2024 and is currently pursuing an M.S. in Electrical Engineering at MSU. His current research focuses on embedded systems, FPGA based rule engines, and hardware–software co-design for FPGA-based, radiation tolerant computing platforms.



Hezekiah Austin received an A.A. degree in General Studies from Regent University in 2014, a B.S. with Honors in Electrical Engineering from Montana State University in 2020, and an M.S. in Electrical Engineering from Montana State University in 2023. He is currently a research assistant and Ph.D. candidate at Montana State University, researching dynamically reconfigurable hardware accelerators for FPGA-based, low power, radiation tolerant computer systems.



Chris Major received his Ph.D. in electrical engineering from Montana State University (MSU). He is currently the principal engineer at Resilient Computing and the principal investigator on multiple NASA SBIR projects to advance space computing. His expertise is in FPGA design with a specific focus on implementing novel computer architectures that leverage real-time configuration of the programmable FPGA fabric. Major's research is sponsored in part by NASA and the National Science Foundation.



Brock J. LaMeres (M'98-SM'09) received the B.S. degree in electrical engineering from Montana State University, Bozeman, in 1998, the M.S. degree in electrical engineering from the University of Colorado, Colorado Springs, in 2001, and the Ph.D. degree in electrical engineering from the University of Colorado, Boulder, in 2005. He is currently a Professor in the Department of Electrical and Computer Engineering at Montana State University, Bozeman. LaMeres teaches and conducts research in digital systems. His research is sponsored in part by NASA, the National Science Foundation, the Idaho National Laboratory, and the Pacific Northwest National Laboratory. He has published over 100 papers and five textbooks in computer engineering and holds 15 US patents in digital systems.



Garrett Perkins received the B.S. in Computer Engineering and the B.A. in Modern Languages and Literature from Montana State University (MSU), Bozeman, MT, in 2022. He went on to pursue an M.S. in Cybersecurity from MSU, graduating in 2025. He is currently a Ph.D. student in Electrical Engineering at MSU. His research interests include bleeding edge cybersecurity solutions for field-programmable gate arrays (FPGAs) with applications in aerospace systems.