Real-Time, Dynamically Reconfigurable Coprocessor for Image Processing in Small Satellites

Hezekiah Austin
Electrical & Computer Engineering
Montana State University
Bozeman, USA
hezekiah.austin@student.montana.edu

Dr. Chris Major
Research & Development
Resilient Computing
Bozeman, USA
chris.major@resilient-computing.com

Kris Allick
Electrical & Computer Engineering
Montana State University
Bozeman, USA
kris.allick@gmail.com

Tristan Running Crane,
Benjamin Macht
Electrical & Computer Engineering
Montana State University
Bozeman, USA
tristanrunningcrane@gmail.com
benjamin.macht@student.montana.edu

Dr. Clemente Izurieta
Gianforte School of Computing
Montana State University
Bozeman, USA
lemente.izurieta@montana.edu

Dr. Brock LaMeres
Electrical & Computer Engineering
Montana State University
Bozeman, USA
lameres@montana.edu

***Abstract*—Field Programmable Gate Arrays, FPGAs, are increasingly used in space as data processors for sensors. For low power spacecraft, payloads and small satellites, the reconfiguration capability of FPGAs increases the versatility via hardware acceleration without greatly increasing the power consumption. Using partial reconfiguration, the implemented logic design is changed on the hardware level by altering the connections between resources. A dynamically reconfigurable coprocessor allows small satellites with power constraints to optimize the onboard processing capability for computationally intensive tasks. The coprocessor functions as a hardware accelerator providing data intensive processing or functionality unsupported by the main processor. This approach was developed and tested by swapping between one-dimensional and two-dimensional first difference edge detection coprocessors. A single coprocessor was implemented as a peripheral for a 32-bit integer aerospace processor, RadPC RISC-V RV32I. Dynamic partial reconfiguration is used to swap the coprocessor between different edge detection algorithms in real-time. The coprocessor was evaluated for use in small satellites based on four main criteria, successful coprocessor swapping, real-time operation, FPGA resource utilization, and power consumption. Successful operation was verified through waveform logging on the FPGA configuration interface. The dynamic partial reconfiguration time and operation runtime were measured to confirm that real-time operation for the overall system was possible. FPGA resources were measured via the Vivado design tool after implementation. Power consumption was estimated by the Vivado Power Analysis Tool and confirmed via lab power supply measurements. The results of this study provide evidence that the use of dynamically reconfigurable coprocessors has the ability to increase the computational capability of small satellite computers while fitting within the constrained resources of such missions.**

Thanks to NASA for funding this research.
Thanks to Montana State University for supporting this research.

***Index Terms*—fault tolerant, FPGA, image processing, partial reconfiguration, small satellite**

I. Introduction

Dynamic partial reconfiguration takes advantage of hardware flexibility inherent in a Field Programmable Gate Array (FPGA) to swap hardware accelerators during real-time operation. As FPGA-based system are increasingly being used as sensor data processors, swappable accelerators provides a possible solution for onboard processing of flight data. This line of research balances power consumption, real-time processing, radiation tolerance, and reconfiguration time as shown in Figure 1.

Low-power spacecraft in particular would benefit as dynamic partial reconfiguration greatly increases versatility without dramatically increasing power consumption or monopolizing the main processor. Mass, size, and power are all constrained resources for small satellites, creating obvious limitations for the onboard computers. Small satellite computers must balance performance, power consumption, and reliability as design constraints [1]. The dynamically reconfigurable coprocessor functions as a hardware accelerator for data intensive operations which are unsupported or slow for the main processor. Dynamic partial reconfiguration enhances an FPGA-based system with in-flight reprogrammable hardware and real-time adaptive functionality [2]. This approach has several advantages for low-power, FPGA-based computer systems. First, hot-swapping a coprocessor does not interrupt the main processor operation. Parallel implementation of a softcore processor and coprocessor allows the coprocessor to be swapped out in the background. Secondly, the coprocessor

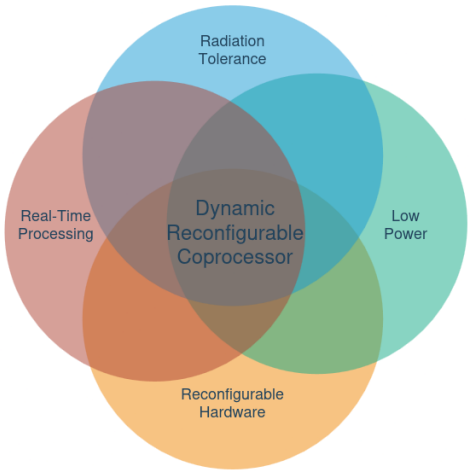


Fig. 1: **Venn diagram of research scope.**

reuses a set number of FPGA resources. This limits the power consumption for the coprocessor and allows the total usage of FPGA resources to be maximized. Third, coprocessors allow for a wide degree of flexibility in functionality. By standardizing the processor/coprocessor interface, a variety of coprocessors are integrable into the computer architecture and accessible to the user on the software level.

The developed edge detection coprocessor leverages these advantages to demonstrate its feasibility for image processing onboard a small satellite. Given the bandwidth limitations inherent to communications, even limited onboard image processing is greatly beneficial. Furthermore, this coprocessor design allows for swapping during flight. This allows more efficient hardware to handle the data intensive task of image processing at greater speed.

A. Radiation Environment

Small satellites generally operate in Low Earth Orbit (LEO) beyond the environmental protection provided by Earth's atmosphere and magnetosphere. In these orbits, bombardment by high-energy particles causes a significant number of failures in the computer [3]. While operating inside this harsh radiation environment, computers suffer from two board categories of radiation effects, Total Ionizing Dose (TID) and Single Event Effects (SEE) [4]. Both of these radiation effects are caused by ionizing radiation striking the substrate of an integrated circuit (IC) and depositing energy.

TID is a culmination of gradual degradation of the device. When lower energy protons and electrons ($\leq$ 30 MeV/amu) deposit energy in the substrate, it creates electron/hole pairs in the IC's layers and TID occurs [6]. The two main causes of damage are trapped charge on the gate oxide or inside isolation region. Trapped charge on the gate oxide of a transistor will alter the threshold voltage and switch the semiconductor to either always on or always off. When the trapped charge is inside the isolation region between transistors, it causes leakage current and power consumption until the device is burnt out. Modern IC feature sizes ($< 22nm$) makes most devices naturally resistant to TID since lower energy particles are less likely to be trapped in isolated region or gate oxide of devices [7]–[10]. Modern COTS process nodes ($< 22nm$) have high levels of TID immunity ($> 50krad$) due to their thin insulation regions reducing the likelihood of charge trapping [9], [11]. This allows for longer missions with COTS parts before failure due to TID becomes a factor.

A SEE is a rapid spike in energy on the logic lines inside a device due to a radiation strike. When high energy particles and heavy ions strike the diffusion regions of a device, an SEE can occur. These effects appear as transient pulses in circuitry and bitflips in memory cells or registers [12]. An SEE does not necessarily cause permanent damage. Destructive SEEs are generally second order effects were the transient pulse causes a current draw which damages a device [13]. For non-destructive SEEs, the electron/hole pairs cause unwanted logic-level transitions inside ICs. These logic-level transitions escalate to system faults when the transitions occur on a digital logic line, single event transient (SET), or when the transitions are stored, single event upset (SEU). Single event functional interrupts (SEFI), occur when the system cannot be recovered by resetting the affected circuit and entire system requires a power cycling, full system re-initialization, or reconfiguration.

B. Softcore Processor, RadPC

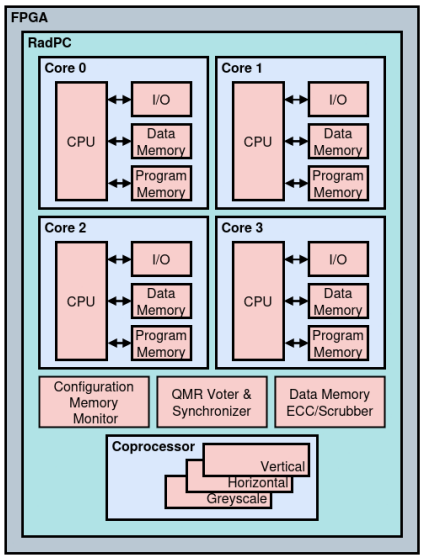


Fig. 2: **RadPC+Coprocessor block diagram.**

This research focused on the development of coprocessors for RadPC, a softcore processor. RadPC is an FPGA-based, flight-proven aerospace processor developed by Montana State University (MSU). This processor was designed for radiation tolerance and uses a unique 32 bit, reduced instruction set (RISC-V RV32I) softcore with modular redundancy. RadPC was development in the VHSIC Hardware Description Language (VHDL) for implementation on an FPGA. This proces-

sor is an architectural approach to mitigate SEEs during space operations. The RadPC computer architecture expands triple modular redundancy (TMR) to quad modular redundancy (QMR) at the central processing unit (CPU) level as shown in Figure 2. RadPC functions as a single core microcontroller using a QMR RISC-V architecture implemented on an FPGA [14], [15]. This processor uses a number of recovery systems including, voters, memory scrubbers, register-level reloading, and core synchronization to maintain operation during faults. When faults are detected, the faulted core undergoes recovery procedures. Suspect results from the faulted core are outvoted by the three trusted cores and ignored during recovery. In the event of a complete failure of a core from SEEs, the individual core is partially reconfigured, memory and registers are reloaded to synchronize with the other cores and operation is continued [16], [17]. These recovery methods, combined with memory scrubbers, provide resilience from SEE faults [14], [16], [18]. This architectural approach to recovery from radiation induced faults allows the system to be easily expanded to include dynamic reconfigured coprocessors. Since RadPC uses partial reconfiguration for recovery from radiation strikes, the hardware and softcore architecture is easily adaptable to using partial reconfiguration to dynamically swap coprocessors. Furthermore, RadPC is an aerospace processor developed for small satellites. The developed edge detection coprocessor focuses on increasing the capability of edge computing for limited resources, harsh radiation environment applications.

C. Prior Coprocessor Development

Three key developments were completed which made the dynamically reconfigurable image coprocessor possible. First, a compact version of a first difference edge detector was developed. Pictures of the Moon's lunar craters were selected as test images. These images were chopped square and downsampled to 16x16 pixels. An RGB Standard to Greyscale convertor was developed using the rightwise bitshift and addition method for efficient FPGA resource utilization [19]. See Table for conversion factors I for details.

TABLE I: Conversion Factors

Colors	Greyscale Conversion Weights	
	Standard Weight	***Used Weight***
Red	0.299	0.298828125
Green	0.587	0.5859375
Blue	0.114	0.11328125

This approach illustrated the advantage of using a coprocessor instead of the main processor. As the main processor, RadPC, is an integer operation only processor and is not capable of rapidly calculating the greyscale conversion of multiple pixels. This method generates slight incorrect conversion factors for Red, Green, and Blue (RGB) pixels. However, it does save FPGA resources by only using D-Flip-Flop and addition logic elements on the FPGA. This allowed decimal weights to be calculated using a coprocessor to implement Equation 1 for RGB pixels conversion into greyscale pixels.

$$Grey = C_{red} * R_{RGB} + C_{green} * G_{RGB} + C_{blue} * B_{RGB} \quad (1)$$

Second, a proof-of-concept coprocessor was integrated into the RadPC QMR RISC-V computer architecture. After a number of test variations were tested, a final integration setup was selected. In this setup, the coprocessor is implemented outside the QMR architecture with the coprocessor inputs and outputs written to as a peripheral.

The advantages of this setup is the coprocessor is not duplicated into the QMR and the required FPGA resources are reduced by a factor of 4. Limiting the number of partial reconfiguration blocks for the coprocessor directly decreases the required partial reconfiguration times for a coprocessor swap. However, the coprocessor is exposed to possible radiation strikes without any redundancy. Currently, the solution is to limit the amount of time a coprocessor is operating and periodically swap coprocessors to flush out the PR region. This approach has been demonstrated to effectively mitigate SEEs in LEO.

Finally, the proof-of-concept coprocessor was implemented for real-time function swapping using manually triggered partial reconfiguration of different coprocessors. The results showed that partial reconfiguration during real-time operation was possible [20]. A major limiting factor was the requirement of manually triggering a partial reconfiguration as operation was paused until the user could order a partial reconfiguration. Further research illustrated that a dynamic partially reconfigurable coprocessor requires an external microcontroller to rapidly completed for real-time operation.

D. The Hydra Single Board Computer

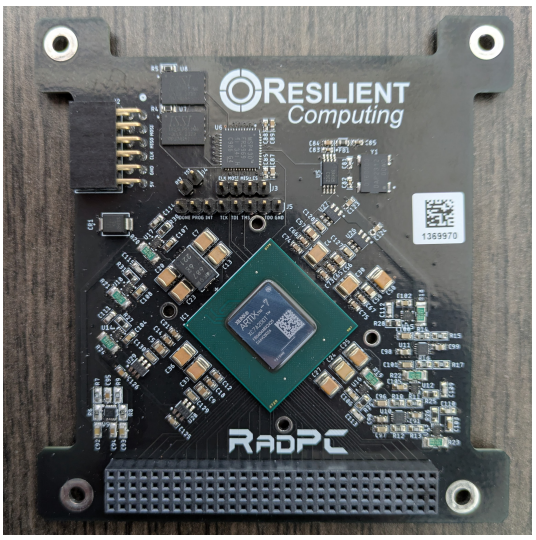


Fig. 3: **Hydra: Single Board Computer**

The Hydra Single Board Computer, shown in Figure 3, was developed to provide the dynamic partial reconfiguration

capability needed for this research. Hydra uses a Xilinx Artix 7 200T FPGA with RadPC implemented on it as the central processing unit (CPU) and a microcontroller implemented as the background system for FPGA configuration. Hydra consolidates all capabilities used in previous research onto a single board, including FPGA configuration, power monitoring, UART support, SPI support, memory storage for bitstreams.

II. Research

Preliminary research found the edge detection coprocessor was integrable into the RadPC computer architecture and the hardware allowed for dynamic hot-swapping between coprocessors via partial reconfiguration by a microcontroller. Based on this previous work, three (3x) edge detection coprocessors was developed for the operations of RGB Standard to Greyscale Conversion, Horizontal First Difference Edge Detection, and Vertical First Difference Edge Detection. This research expanded on previous work by automating the swapping procedure for the coprocessors and focusing on image processing applications. For dynamic partial reconfiguration, an external microcontroller was used to control FPGA configuration process and an external memory was used to store multiple partial reconfiguration bitstreams. While automated swapping does not decrease actual partial reconfiguration time, it does decrease effective partial reconfiguration time by removing manual commands required from the user.

A. Testing Methodology

Evaluation criteria for the coprocessor focused on four key points: swapping via partial reconfiguration, partial reconfiguration time, coprocessor operation, and coprocessor operation time. Successful completion for these key points demonstrates the potential of dynamic partial reconfiguration as a method to increase capability in small, low power satellites. Coprocessors were dynamically swapped into the system via partial reconfiguration. To verify a successful reconfiguration, test inputs were loaded into the coprocessor and the results were compared to Vivado simulations of the coprocessor's operation. The interface for partial reconfiguration was monitored and recorded for reconfiguration time. Operation was tested via comparison between calculated, simulated, and actual output images. Inputs were limited to 16x16 pixel images (256 pixels total) due to memory constraints on RadPC for this implementation and for easy verification via manual calculation.

B. Testing Hardware Setup

Real-time swapping of the coprocessor via dynamic partial reconfiguration was tested on the Hydra Single Board Computer. The test setup consisted of a power supply, a Hydra Computer, an Analog Discovery 3 (AD3), and a FTDI USB-to-UART cable as shown in Figure 4. Partial reconfiguration on Hydra was monitored by the Analog Discovery via header pins on the FPGA's JTAG interface. Capturing these signals allowed the configuration process to be recorded during coprocessor swapping. During operation, the results were recorded on a

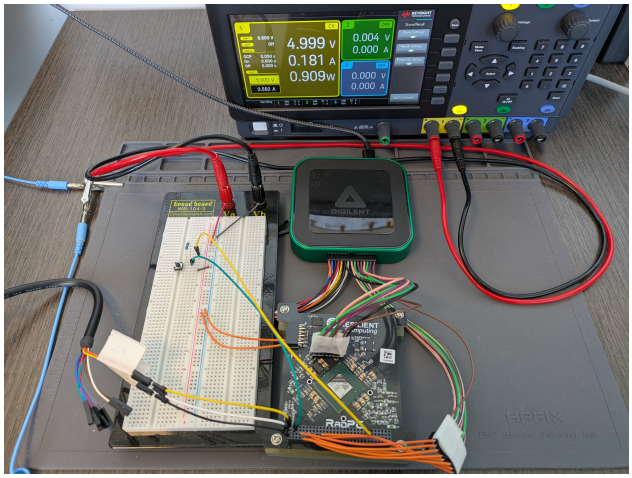


Fig. 4: **Test setup: power supply, AD3, and Hydra**

desktop computer via the UART cable and a terminal logger. The AD3 monitored 9 General Purpose Input/Output (GPIO) pins from the Hydra's PC104 connector. These pins consisted of the most significant byte of the coprocessors output and the coprocessor operation flag. These allow verification for functionality during operation and the measurement of total operation time.

The Hydra operated with two clock domains: 16 MHz for the microcontroller handling partial reconfiguration over JTAG, and 32 MHz for RadPC. For tapping the JTAG interface, the Analog Discovery was clocked at 33.34 MHz to insure stable recording of the swapping process. Timing for the partial reconfiguration was measured in Digilent's waveforms on a recorded transmission. Operation was measured via a GPIO pin which was set to raise immediately after the coprocessor was loaded and fall immediately after the coprocessor outputs were read. This allowed the coprocessor's operation time to be measured with a minimum amount of time added on either end from data transfer.

C. Coprocessor Swapping

The coprocessor was implemented as partial reconfiguration block in parallel to RadPC. As shown in Figure 5, the different coprocessors are swapped out during operation and each coprocessor generates a different processed image.

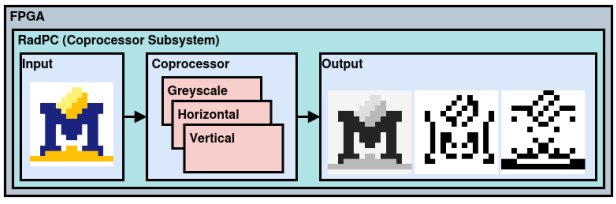


Fig. 5: **Coprocessor block diagram with MSU logo images**

For these tests, images were preloaded into RadPC's memory and results from each step of the edge detection were passed over UART to a terminal for logging. Logged results

were converted from hex codes into full pixels and paired with their respective bitmapped header.

D. Results

Partial reconfiguration was the limiting factor for previous versions of the coprocessor. The previous process required the user to manually initiate a partial reconfiguration in response to RadPC's request for a specific coprocessor. The new process developed in this research has a microcontroller to receive RadPC's request for a specific coprocessor and automatically initiate a partial reconfiguration to implement the requested hardware onto the FPGA. Figure 6 shows a rapid series of partial reconfiguration with each coprocessor processing a 16x16 pixel image between coprocessor swapping. Three operations were performed, greyscale conversion, horizontal first difference edge detection, and vertical first difference edge detection. Full configuration was performed first, followed by Greyscale Conversion. The greyscaled image was piped into the horizontal and vertical edge detection coprocessors and the overall process captured by the AD3. From this signal capture, configuration and operation times were measured for Table II.

TABLE II: Partial Reconfiguration and Operation Times

	Configuration and Operation Times			
Configurations	***Configuration (sec)***		***Operation (sec)***	
RadPC	✓	28.5488	✓	n/a
Greyscale	✓	2.497	✓	60.11
Horizontal	✓	2.497	✓	31.71
Vertical	✓	2.497	✓	30.20
2Dimensional	✓	2.497	✗	30.15
RGB	✓	2.497	✓*	60.15

Reconfiguration times for the coprocessor were mainly driven by microcontroller clock frequency (16MHz), bit-banged JTAG signals, and partial reconfiguration block size. The design is slow overall, with a 2.5 second time for swapping. The purpose of this design to increase hardware flexibility for low power satellites, not speed of configuration. For this purpose, the low speed reconfiguration is acceptable within the design constraints. Operation time is dominated by data transfer in RadPC for reading and writing images to and from memory. Total operation for the program shown in Figure 6 was 163.70 seconds for a full configuration, RadPC programming, three coprocessor swaps and the generation of three processed images of the MSU logo shown in Figure 7.

Coprocessor partial reconfiguration was completely successful for all five cases. After partial reconfiguration, coprocessors were individually swapped into the FPGA and RadPC was able to load image data into the coprocessor. The coprocessors were able to generate recognizable images from images with distinct shapes. Functionality testing on the coprocessor raised some issues with data processing for two out of the five test cases, RGB Conversion and 2D Edge Detection. As shown in Figure 7, Greyscale Conversion, Horizontal Edge Detection, and Vertical Edge Detection worked correctly. These three coprocessors each generated a viable 16x16 image with single byte pixels for its result. 2D Edge Detection also generated a 16x16 image with single byte pixels. However, the output image was not identifiable as a processed version of the input image. The issue was isolated to the 2D Edge Detection needing a Gaussian Blur Function in the pipeline before running the first difference algorithm. This allows the horizontal and vertical components of 2D Edge Detection to correctly combine for all threshold values. In contrast, the RGB Conversion was successful. However, the current RadPC UART used to pass the data out was not able to handle the full array for a 16x16 image with 4 byte pixels. Rapidly sending 1024 bytes causes a single bit offset during transmission. The RGB Conversion appears to be working correctly and has not been verified due to the communication issue.

E. Power

FPGA power was estimated using Vivado's power monitor tools. Board power was measured during operation by the power supply during operation as shown in Table III.

TABLE III: Estimated FPGA and Measured Board Power

	Measured & Estimated Power	
Estimated during Synthesis	**Device**	***Power (W)***
Total	FPGA	0.414
Implemented Design	FPGA	0.273
Clock	FPGA	0.024
RadPC	FPGA	0.149
Coprocessor	FPGA	0.001
I/O	FPGA	0.001
Measured during Operation	**Device**	***Power (W)***
Preconfiguration	Hydra	0.995
Full Configuration	Hydra	1.922
RadPC Reset	Hydra	1.473
RadPC Bootloading	Hydra	1.993
Partial Reconfiguration	Hydra	2.133
Greyscale Operation	Hydra	2.136
Horizontal Edge Detection	Hydra	2.141
Vertical Edge Detection	Hydra	2.137

Power during operation was roughly 2(W). Measured power for the FPGA was higher than estimated at roughly 1(W) due to I/O usage by RadPC. The estimated power assumed significantly less I/O usage than actual operation required.

III. Conclusions

The principal results of this work were a functional design demonstrating the potential of implementing a dynamically reconfigurable coprocessor within the RadPC architecture. Automation of the coprocessor swapping process via partial reconfiguration was achieved. RGB to Greyscale conversion and two variations of a first difference edge detection algorithm were successfully implemented.

A. Applications & Advantages

This proof-of-concept design confirmed dynamically reconfigurable coprocessors for low-power, real-time image processing is possible for small satellites. These custom coprocessors increase the versatility and capability for a low-power computing system. Hardware for this design was developed using the PC104-Plus standard, which fits the size, mass, and

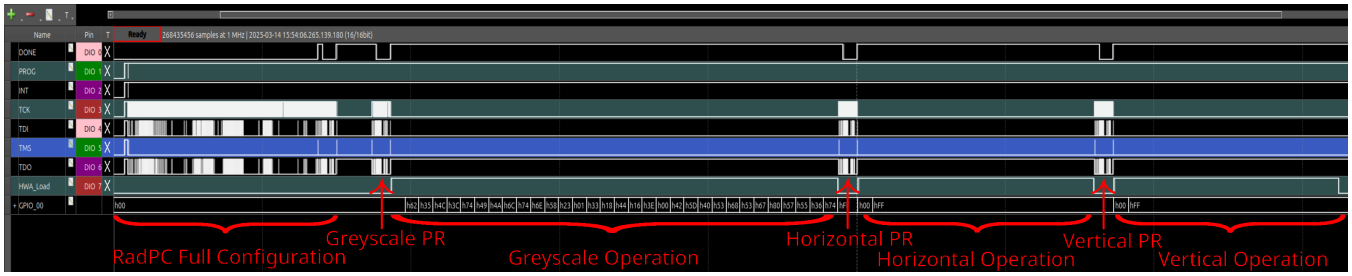
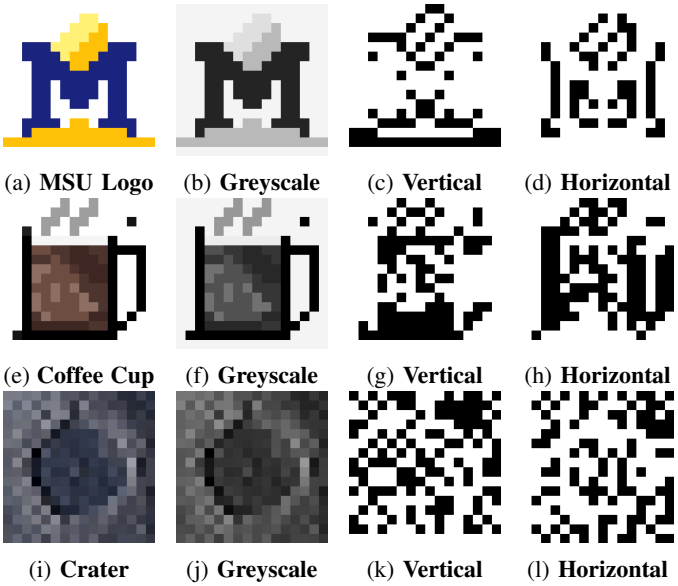


Fig. 6: **RadPC configuration, coprocessor reconfiguration, and image processing**



(a) **MSU Logo** (b) **Greyscale** (c) **Vertical** (d) **Horizontal**

(e) **Coffee Cup** (f) **Greyscale** (g) **Vertical** (h) **Horizontal**

(i) **Crater** (j) **Greyscale** (k) **Vertical** (l) **Horizontal**

Fig. 7: Test images for edge detection coprocessor

power constraints for a small satellites. Key future applications include image preprocessing onboard the small satellite where dedicated processing hardware is needed for a short period. [24], [25] Swapping allows an image preprocessing coprocessor hardware to support other operations after preprocessing is completed. The advantage of this design is the capability to change the hardware during real-time operation. Furthermore, the reconfiguration process clears any SEEs out of the hardware while loading the new coprocessor. This adds a limited, but beneficial recovery method from radiation strikes for the coprocessor subsystem.

B. Limitations & Future Work

Dynamic partial reconfiguration is a tradeoff between used FPGA resources and operation time. This imposes limitations, shown in Table IV, on the capability of the system. Total PR time for the coprocessors was 2.497 seconds which is a long delay for a 16MHz system. Optimization of the PR code combined with coprocessor preloading in the background can reduce this time. The parallel port for between RadPC and the microcontroller limits acknowledgement and debug capability. RadPC needs a method of internally identifying coprocessors and a test procedure for verify the coprocessor's partial reconfiguration was successful.

TABLE IV: Current Limitations and Proposed Solutions

Subsystem	**Limitations and Proposed Solutions**	
	Current	***Future Upgrade***
PR Speed	2.497 seconds	PR code optimization
PR Request Port	Parallel	Serial
Coprocessor Internal ID	None	Internal Register with ID
PR Verification	None	Acknowledgement

C. Summary

Dynamic partial reconfiguration was successful in swapping coprocessors during real-time operation. Basic image processing was performed on 16x16 images and three coprocessors generated recognizable results. This researched demonstrated several advantages for low-power, FPGA-based computer systems. Parallel implementation of a softcore processor and coprocessor allows the coprocessor to be swapped out in the background. The coprocessor reused a set number of FPGA resources and power. The FPGA resources were reused during operation for several functions and allowed a large degree of flexibility in capability. The developed edge detection coprocessor leveraged these advantages to demonstrate its feasibility for image processing onboard a small satellite. Even limited onboard image processing is greatly beneficial for satellite operations were power or bandwidth is a constraint. The current limitations with this approach are solvable with further development.

In conclusion, this line of research achieved image processing in real-time, on a low-power system, using partial reconfiguration to swap the implemented FPGA logic design. Low-power spacecraft, including small satellites, would benefit from dynamic partial reconfiguration for data intensive, unsupported, or specialist operations.

REFERENCES

[1] W. Sajjad, A. Shafique and R. Mahmood, "Designing of Reliable, Low-Power, and Performance-Efficient Onboard Computer Architecture for CubeSats," in IEEE Journal on Miniaturization for Air and Space Systems, vol. 5, no. 2, pp. 59-72, June 2024, doi: 10.1109/JMASS.2023.3342208.

[2] B. Osterloh, H. Michalik, S. A. Habinc and B. Fiethe, "Dynamic Partial Reconfiguration in Space Applications," 2009 NASA/ESA Conference on Adaptive Hardware and Systems, San Francisco, CA, USA, 2009, pp. 336-343, doi: 10.1109/AHS.2009.13.

[3] Y. Lu, Q. Shao, H. Yue and F. Yang, "A Review of the Space Environment Effects on Spacecraft in Different Orbits," in IEEE Access, vol. 7, pp. 93473-93488, 2019, doi: 10.1109/ACCESS.2019.2927811.

[4] Holmes-Siedle, A., Adams, L., "Handbook of radiation Effects", NY, Oxford Univ. Press, 2002.

[5] B. J. LaMeres, "Fpga-based radiation tolerant computing," Journal of Aerospace Information Systems, 2012.

[6] H. L. Hughes and J. M. Benedetto, "Radiation effects and hardening of mos technology: devices and circuits," IEEE Transactions on Nuclear Science, vol. 50, no. 3, pp. 500–521, 2003.

[7] H. J. Barnaby, "Total-ionizing-dose effects in modern cmos technologies," IEEE Transactions on Nuclear Science, vol. 53, pp. 3103–3121, 12 2006.

[8] 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, 2017.

[9] C. Zhang and et al., "Total ionizing dose effects on analog performance of 28 nm bulk mosfets," 2017 47th European Solid-State Device Research Conference (ESSDERC), pp. 30–33, 2017.

[10] C. Zhang and et al., "Gigarad total ionizing dose and post-irradiation effects on 28 nm bulk mosfets," 2016 IEEE Nuclear Science Symposium, Medical Imaging Conference and Room-Temperature Semiconductor Detector Workshop (NSS/MIC/RTSD), pp. 1–4, 2016.

[11] Arora, R., et al., "Impact of Technology Scaling in sub-100 nm nMOSFETs on TID Radiation Response and Hot-Carrier Reliability," IEEE Trans. on Nuc. Science, vol. 61, no. 3, June 2014.

[12] G. Kottaras, T. Sarris, A. Psomoulis, I. Ioakeimidis, A. Papathanasiou, D. Pitchford, & I. Sandberg. A low-power, radiation-hardened Single Event Effect rate detection System on a Chip for Real Time Monitoring of Single Event Effects on Low Earth Orbit satellites. *2022 IFIP/IEEE 30th International Conference On Very Large Scale Integration (VLSI-SoC)*. pp. 1-6 (2022)

[13] D. Lo, & T. Tran. Single Event Effects Susceptibilities of Select Commercial-Off-The-Shelf Components for Space. *2022 IEEE Radiation Effects Data Workshop (REDW) (in Conjunction With 2022 NSREC)*. pp. 1-7 (2022)

[14] 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.

[15] J. Williams, C. Barney, Z. Becker, J. Davis, C. Major, B. J. LaMeres, and B. Whitaker, "Radpc@scale: A novel approach to radpc single event upset mitigation strategy," 2022 IEEE Aerospace Conference, 2022.

[16] B. J. LaMeres and C. Gauer, "Dynamic reconfigurable computing architecture for aerospace applications," 2009 IEEE Aerospace Conference, 2009

[17] C. Gauer, B. J. LaMeres, and D. Racek, "Spatial avoidance of hardware faults using fpga partial reconfiguration of tile-based soft processors," 2010 IEEE Aerospace Conference, 2010.

[18] J. Hane, B. J. LaMeres, R. W. T. Kaiser, and T. Buerkle, "Increasing the radiation tolerance of fpga-based computers through redundancy and environmental awareness," Journal of Aerospace Information Systems, vol. 11, no. 2, pp. 61-75, 2014

[19] K. Kumar, R. Mishra, & D. Nandan, "Efficient Hardware of RGB to Gray Conversion Realized on FPGA and ASIC," Procedia Computer Science. **171** pp. 2008-2015 (2020), https://www.sciencedirect.com/science/article/pii/S187705092031200X, Third International Conference on Computing and Network Communications (CoCoNet'19)

[20] H. Austin, C. Major, Z. Becker, T. Running Crane, K. Allick, B. J. Lameres, "Dynamically Reconfigurable Coprocessor for Floating-Point Arithmetic Capability in Small Satellites," "unpublished"

[21] J. W. Abdala Castro and A. Morales-Villanueva, "Exploring Dynamic Partial Reconfiguration in a Tightly-coupled Coprocessor Attached to a RISC-V Soft-processor on a FPGA," 2021 IEEE XXVIII International Conference on Electronics, Electrical Engineering and Computing (INTERCON), Lima, Peru, 2021, pp. 1-4, doi: 10.1109/INTERCON52678.2021.9532810.

[22] N. Charaf, A. Kamaleldin, M. Thümmler, and D. Göhringer, "RV-CAP: Enabling Dynamic Partial Reconfiguration for FPGA-Based RISC-V System-on-Chip," 2021 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), Portland, OR, USA, 2021, pp. 172-179, doi: 10.1109/IPDPSW52791.2021.00033.

[23] J. Tarrillo, F. A. Escobar, F. L. Kastensmidt and C. Valderrama, "Dynamic partial reconfiguration manager," 2014 IEEE 5th Latin American Symposium on Circuits and Systems, Santiago, Chile, 2014, pp. 1-4, doi: 10.1109/LASCAS.2014.6820293.

[24] B. Krill, A. Amira, A. Ahmad and H. Rabah, "A new FPGA-based dynamic partial reconfiguration design flow and environment for image processing applications," 2010 2nd European Workshop on Visual Information Processing (EUVIP), Paris, France, 2010, pp. 226-231, doi: 10.1109/EUVIP.2010.5699127.

[25] C. V. Borkute, A. Y. Deshmukh and C. N. Kharkar, "Dynamic Partial Reconfiguration in FPGAs for DSP Applications," 2011 Fourth International Conference on Emerging Trends in Engineering & Technology, Port Louis, Mauritius, 2011, pp. 253-257, doi: 10.1109/ICETET.2011.70.

[26] X. Shi and A. Zou, "Preemptive FPGA Scheduling Based on Dynamic Partial Reconfiguration," 2024 Conference of Science and Technology for Integrated Circuits (CSTIC), Shanghai, China, 2024, pp. 1-3, doi: 10.1109/CSTIC61820.2024.10531952.

[27] A. Dhar et al., "DML: Dynamic Partial Reconfiguration With Scalable Task Scheduling for Multi-Applications on FPGAs," in IEEE Transactions on Computers, vol. 71, no. 10, pp. 2577-2591, 1 Oct. 2022, doi: 10.1109/TC.2021.3137785.

[28] M. Angioli, M. Barbirotta, A. Mastrandrea, S. Jamili and M. Olivieri, "Automatic Hardware Accelerators Reconfiguration through LinearUCB Algorithms on a RISC-V Processor," 2023 18th Conference on Ph.D Research in Microelectronics and Electronics (PRIME), Valencia, Spain, 2023, pp. 169-172, doi: 10.1109/PRIME58259.2023.10161944.