

IAC–20–D5.3.4

## Software for Testing and Mitigating Radiation-induced Effects in Commercially Available Integrated Circuits

**Richard Arthurs**

Mechatronic Systems Engineering, Simon Fraser University, Canada, rarthurs@sfu.ca

**Andrada Zoltan**

Computer Engineering, University of British Columbia, Canada, andrada.zoltan@alumni.ubc.ca

### Abstract

Testing electronic components for the radiation environment they will encounter in space has long been an important part of spacecraft engineering. As small satellites increasingly rely on non-redundant, tightly integrated electronic components with smaller semiconductor feature sizes, understanding the radiation-induced effects on individual components is becoming increasingly important. Even on short-duration nanosatellite missions where little component degradation may occur, the single-event effects on memories, real-time clocks, and processors can have large impacts on mission operations if they are not taken into account.

The ORCASat Command and Data Handling team has performed radiation testing on commercial off-the-shelf real-time clocks, non-volatile memory, and microcontroller components. The procedure undergone for testing these components is presented, alongside a description of how software can be developed to gain useful results during radiation testing. The results of the testing are also presented and the mission impacts of observed single-event effects in selected components are discussed. Emphasis is placed on how the functional implications of single-event effects are tested for, and how flight software can be designed to tolerate them, once they are understood.

**Keywords:** CubeSat, proton radiation testing, TMS570, single event upset, fault-tolerant software, ECC

### 1. Introduction

The use of commercial-off-the-shelf (COTS) components in nanosatellites is popular due to budgetary reasons and ease of acquisition [1]. As with all commercially available parts, susceptibility to radiation-induced effects is a risk. As such, it is important to radiation test the critical components used in a system to properly understand and qualify their resilience to such an environment. These effects can be mission critical, and without testing, it is not possible to determine what the behaviours look like [2].

ORCASat is a Canadian Space Agency funded project, involving the development of a 2U CubeSat used for optical telescope calibration. The Command and Data Handling (C&DH) system consists of a custom on-board computer (OBC) design made entirely from COTS components and an in-house firmware architecture built upon FreeRTOS. The team has performed proton radiation testing at the TRIUMF Proton Irradiation Facility on the three most critical components in the system: NOR flash, which stores on-board telemetry, real-time clock (RTC), which is used as the central time source in the spacecraft, and the TMS570-series microcontroller unit (MCU) that executes all C&DH functionality.

We present the methodology and lessons learned from

testing, with emphasis on the software that allowed for significant automation. These techniques enabled us to execute our testing within the short time frame of several hours. We also present the test results, which affirm the importance of considering radiation-induced effects, and discuss their implications on our mission. The contributions of this paper further lie in offering mitigation strategies implemented in firmware that can improve the robustness of a CubeSat system, and discussing techniques that can be used if radiation testing is not an option.

### 2. Background

Proton testing of commercially-available microelectronics for spacecraft applications has been recommended for its ability to simultaneously screen for total ionizing dose (TID) and single-event effects (SEE) [2]. During these tests, parts are powered on and functionally exercised, and anomalies such as communication failure, memory corruption, and increased current draw are recorded [3]. The cross section of a particular device is calculated from the number of detected upsets and the fluence delivered during the test, and can be used to estimate the upset rate in a particular environment [2].

Devices from the TMS570 MCU series have been radiation tested by at least two groups [4][5]. During both

sets of tests, the MCU was running test firmware, and a UART connection to a host PC was used to monitor for radiation-induced effects on the device. The results of [4] show cross sections for several different benchmarks, each designed to exercise specific elements of the processor. The software used for automating the test result collection is also discussed. The results of [5] are collected with firmware tests focused on verifying communication interfaces such as CAN and Ethernet. The number of detected memory errors and program crashes, as well as the fluence delivered, are reported.

Flight software considerations are also present in prior work, arising from the results of radiation testing. Considerations such as making software resilient to unexpected resets and choosing to store information in radiation-tolerant memory technology have been applied on small satellites such as MarCO [6] [7]. Employing triple modular redundancy techniques in software has also been studied [8], and implemented [6]. Memory scrubbing, typically for FPGA bitstreams, is another technique used to detect and correct faults in space-based systems [9] and memory scrubbing is discussed in this paper for peripheral configuration registers.

### 3. Methodology

Devices were tested at the TRIUMF cyclotron Proton Irradiation Facility using the BL2C beam [10], with the intent of characterizing single event upsets (SEU) and single event latch-ups (SEL) for all devices and TID effects for the TMS570 microcontroller. Table 1 shows the beam characteristics and dosage applied to each device. An energy greater than 95 MeV was used as that is sufficient to pass through the thick substrate used in processors [3].

The physical test setup can be seen in Figure 1. The OBC board held all the tested devices, and was the direct target of the irradiation. As the beam spot size was configured to a 5x5 cm region, not all components on the board could be irradiated at the same time. The movable platform, directed from the control room, allowed us to center the beam on a specific device on the board to ensure maximum exposure. During setup, the position of all the devices under test were recorded so that the platform may be moved between tests without re-entering the irradiation room. The setup also features a custom test platform board, which provided power to the OBC, monitored current, and transmitted periodic log messages back to the host. Power and communication to the entire system was provided using long USB cables that routed to the control room.

All components on the OBC board connected to and communicated with the on-board microcontroller, the TMS570. This device was tested last so that it could also be used in the testing of the other two devices. The MCU

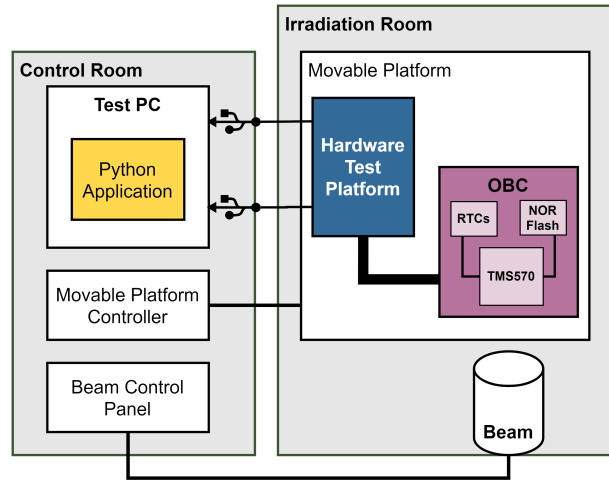


Fig. 1: Radiation Test Setup

executed firmware built on top of FreeRTOS, and could be controlled using ASCII commands over UART. Command functionality ranged anywhere from commencing execution of a task, to requesting transmission of on-board data, to simply requiring acknowledgement. This interface provided great flexibility throughout testing by enabling the configurability of certain features without re-flashing the firmware. The firmware also consisted of pre-written test sequences, contained inside isolated FreeRTOS tasks, that could be activated by command. Another FreeRTOS task was used to blink an LED, which was a useful indicator of whether or not the firmware had been running, and could be easily seen through the video feed from the irradiation room.

A custom Python application, known as Houston [11], was used during testing to facilitate a user friendly interface for communicating with the OBC and automate the execution of tests. All commands to the OBC were transmitted through Houston, with easy-to-access buttons for common commands. The Python application also provided a mechanism for viewing log messages throughout testing and saving them with a timestamp in the background. The timestamps were used to sync hardware activity with the periodically recorded dose, as displayed in the beam control panel. This application is used for all user interfacing and automated testing of the OBC, and was a very convenient tool.

#### 3.1 NOR Flash

The device under test was Micron's 128Mb MT25QL128ABA NOR flash. The purpose of this test was to determine the frequency of SEU in the memory, resulting in data corruption, when exposed to radiation. We were interested in both the data retention of

Table 1: Proton Exposures for Tested Devices

Device	Energy (MeV)	LET (MeV cm <sup>2</sup> /g)	Fluence (protons/cm <sup>2</sup> )	Dose (rad (Si))
MT25QL NOR Flash	116	5.291	$3.31 \times 10^{10}$	3000
PCA2129T RTC	116	5.291	$3.22 \times 10^{10}$	2916
TMS570 MCU	116	5.291	$6.60 \times 10^{10}$	5974

sectors that were not written to, as well as data retention with active programming of the flash. The flash was erased prior to the test, and the procedure while irradiated was as follows:

1. Write a 64-byte sequence of alternating 0's and 1's to flash.
2. Read the 64-byte sequence back from flash twice and compare to what was written.
3. Repeat for the next 64-byte region in flash.

Due to time limitations, this sequence was not completed for the entirety of the flash device. By the end of the test, only 0.2% of the flash had been written to, the remainder was left unprogrammed. After irradiation, a full sweep of the flash was completed to see if any errors were detected in the unaccessed regions. In retrospect, it would have been valuable to also have had some pre-programmed sections of flash prior to irradiation, as most single-bit errors have been shown to be from logic level low to high [12].

### 3.2 Real-Time Clock

The device under test was the PCA2129T real-time clock from NXP Semiconductors [13]. The purpose of this test was to determine the propensity for upsets in the registers of the device.

The proton beam was collimated with an aperture slightly larger than the IC package and was centered on one of the two RTCs present on the OBC, RTC B. Both RTCs were connected to the TMS570 using the SPI interface. The TMS570 executed a test routine that was started before the irradiation began. This routine compared the registers between the irradiated and non-irradiated device, and reported any mismatches. The testing procedure was as follows:

1. Initialize all user-settable registers in each RTC to known values.
2. Read and check all registers in the device. Configuration registers are expected to match their initial values, and are expected to match between both RTCs. The time registers are expected to match

between both RTCs, although the expected value changes as the RTC increments.

3. Report any mismatches.
4. Wait for a period of 2 seconds.
5. Repeat steps 2 through 4 until the test is stopped.

### 3.3 TMS570 Microcontroller

The device under test was the TMS570LS0714PGE from Texas Instruments. The purpose of this test was to assess the propensity for errors in the CPU data SRAM of the device.

The TMS570 CPU data RAM is capable of correcting single-bit errors and detecting double-bit errors with its ECC functionality [14]. For the test, the single-bit error correction functionality was enabled and initialized using default settings from the TI HALCoGEN tool [15].

The TMS570 provides registers that indicate the number of single-bit corrections performed. The counts contained in these registers were polled once per second and reported using the UART. The correction count values were timestamped once received. Positive counts in the correction registers indicate that soft upsets generated as a result of the irradiation were corrected. While logging correction counts, commands were sent to the OBC to verify that its communication peripherals were still active and that firmware was still running.

## 4. Results

### 4.1 NOR Flash

Throughout the duration of testing, there were no detected SEUs in the data written to the flash or issues with communicating to it. In the post-testing sweep of the flash, we noticed that all the data remained as programmed and the unprogrammed regions were untouched.

### 4.2 Real-Time Clock

During the test, two upsets were detected in RTC B, the device centered in the beam. All upsets were single-bit, and were in the time or timestamp registers. No upsets of the configuration registers were detected.

Table 2: Upset incidents during RTC test

RTC	Upset No.	Upset Type	Dose (rad (Si))
B	1	Single-bit	497
B	2	Single-bit	735

On RTC B, both unexpected values were in the years registers. Once the unexpected value appeared, it remained until manually reconfigured over the SPI interface. The test was not long enough to determine if the RTC’s circuitry could increment the year count naturally after the upset. The cross section for RTC B at the tested LET is  $6.21 \times 10^{-11} \text{ cm}^2$ . RTC B received a dose of 2916 rad (Si) over the course of the test, at a rate of approximately 2.62 rad/s. It was functional after the test concluded.

RTC A, while not directly centered on the proton beam, experienced fluences estimated to be 95% of those for RTC B, based on published beam profiles. [16].

The register mismatches were cleared by reprogramming the registers using the SPI interface. No periods of excess current draw were seen, and all SPI transactions with both devices operated properly.

#### 4.3 TMS570 Microcontroller

The TMS570 experienced 21 single-bit upsets in the RAM throughout the course of testing, all of which were successfully detected and corrected by the hardware. There were no double-bit, and thus uncorrectable, upsets detected. Figure 2 shows an approximate linear trend between dose acquired and number of errors in the RAM, but it is not possible to extrapolate behaviour beyond this point as degradation due to TID occurs. The cross section at the tested LET is  $3.18 \times 10^{-10} \text{ cm}^2$ .

From a functional standpoint, the device remained responsive throughout the duration of testing, acknowledging when requested and periodically transmitting the status of the RAM. The current monitor showed no spikes at any point, and the power draw of the CPU core and peripherals remained relatively constant throughout.

### 5. Discussion of Results

The results from testing show that radiation is of concern to all COTS devices in a system, and not just the central processor. When selecting components, research should be done to understand their behaviour under radiation, and components should be tested where possible. It is also important to analyze the failure scenarios that a component may experience, and develop mitigation techniques to deal with them.

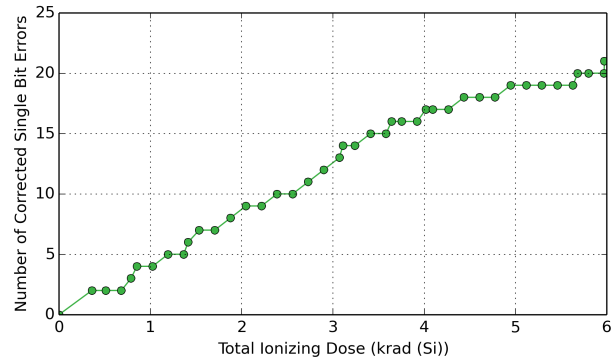


Fig. 2: Corrected single-bit errors in TMS570 RAM vs. dose

#### 5.1 Reliable Processing Unit

Upsets due to radiation in the RAM and flash of modern processors are anticipated to occur, as was seen in the RAM of the TMS570. These upsets can cause data and program corruption, which can lead to execution of unintended instructions, incorrect jumps, program exceptions and other undefined behaviour. Some techniques have been proposed to deal with these problems in software through the form of additional redundancy, but these methods complicate program logic and are difficult to test exhaustively [8] [17]. As shown by the results, the use of error-corrected memory can help alleviate this problem and provide confidence that the code will execute as intended. Many industrial-grade processors are now developed with this technology in place, to ensure reliability in harsher environments. Processors like the TMS570 are well-suited to operate in low-Earth orbit for this reason.

#### 5.2 Reliable Timekeeping

The results of the RTC tests demonstrate that our RTCs can experience non-destructive upsets in the time registers. During flight, this would manifest as unexpected jumps forward or backward in time. Without knowing the details of the construction of the chip, it would also be prudent to assume that the configuration registers (registers whose values are not automatically changed by the chip’s hardware) are susceptible to similar upsets, as the configuration registers are also volatile.

Unexpected jumps in time could certainly have severe mission impacts. On ORCASat, telecommands are scheduled and executed based on the time reference provided by the RTC. If, for example, an SEU were to shift the RTC’s time back a day by corrupting a bit in the day register, it could result in telecommands executing a day late. For important scheduled telecommands such as those used to power-on the communications system, executing at an incorrect time would have an exceptional

negative mission impact.

Actions need to be taken to mitigate this risk. On ORCASat, there are three backup methods used to ensure that a reliable time source is always maintained. The first is in the form of dual-redundant RTCs, which can be compared against one another for error detection. In the case that a mismatch is detected, we know to not rely on the reported time for either RTC. As a secondary method, the TMS570 maintains a backup counter in software that is relatively protected from radiation due to ECC RAM, but does get reset with a microcontroller reset or power loss. Thirdly, ORCASat flies a Novatel OEM719 GNSS receiver. This device is enabled as needed to acquire a known good time, or periodically to recalibrate drifted RTCs. If ORCASat ends up without a time reference, flight software is programmed to not transmit using its radio unless it receives data from ground, to ensure that transmissions can only occur over the regions that the satellite is licensed for.

### 5.3 Assessment of Configuration Registers

The mission impact of unexpected changes to configuration registers depends on the device in question and the particular configuration register bit(s) that are affected. As an example, reliance on the PCA2129's watchdog feature to reset a microcontroller that was unresponsive could be problematic if the watchdog timeout configuration register became corrupted. A corrupted configuration register could massively increase the time required to reset an unresponsive microcontroller, causing the satellite to be uncontrolled for an extended period of time. This could lead to depletion of batteries, thermal problems, missed contacts, or many other effects. Especially when upset probabilities are not known, we recommend performing a systematic assessment of each feature in peripheral ICs, considering how unexpected configuration changes could affect the satellite's operation. For critical components such as watchdogs, we also recommend using devices that are configured passively, such as those that use an RC circuit to select the watchdog timeout period.

### 5.4 Register Scrubbing

Memory scrubbing refers to the practice of restoring the correct configuration of data in memory after the data has been altered by SEUs. Memory scrubbing techniques have been analyzed for FPGA configurations [9]. We propose that similar techniques can be implemented in software to maintain integrity of peripheral IC configuration registers in the space environment.

The OBC software for ORCASat implements scrubbing of the RTC registers. For the configuration registers, which are not expected to change except under con-

trol of software, blind scrubbing is implemented. Under this scheme, the configuration register set is periodically scanned and if an unexpected value is detected, it is updated to the correct value. The golden copy of the register's expected configuration comes from the TMS570's radiation tolerant ECC flash.

## 6. When Radiation Testing is Not an Option

Radiation testing is expensive and difficult to come by, and is often not feasible for many projects. Measures can still be taken to design a reliable system based on COTS components; [2] presents many good practices for hardware design.

### 6.1 Hardware Selection

Selecting robust hardware can greatly impact the reliability of a system. Devices that have successful flight heritage are generally a good choice, as they have already been tested to be operational in space. This is especially important for processor selection, as there are many ways radiation can impact program execution and it is mission critical to maintain defined behaviour of the spacecraft. Choosing devices with ECC memory can be very effective in this respect, by preventing data corruption or at least detecting when data is corrupted beyond repair.

#### 6.1.1 Storage Mechanisms

Software and systems should be designed to store data in media that is known to be reliable in the flight environment. FRAM and MRAM are technologies known to be reliable in space [18] [7]. ORCASat uses MRAM to save state information, and to allow for reconfiguration of many software components without the need for a full firmware update. Once values are loaded from stable media, care should still be taken with the integrity of the data. Especially on systems without ECC RAM, values loaded from non-volatile storage into RAM may become corrupted. In these cases, software solutions should be considered, such as applying triple module redundancy (TMR) to variables [6], or always reading from a reliable source.

### 6.2 Risk Assessment

Once devices are selected, it is important to attempt to understand the possible failures that can occur as a result of radiation. For each device, a literature review can be performed to start to identify the possible effects and failures, as experienced by similar devices. These failures can then be taken and converted to a list of implications on the mission if they were to occur. This exercise is an important step in developing software mitigation strategies, as shortcomings must be identified before being remedied. It may also be the case that performing

this risk assessment brings about changes in the hardware selection, such as addition of redundant parts or exchange for more robust components. On ORCASat, this exercise was performed for all devices on the OBC and motivated several design decisions even prior to radiation testing the components.

### 6.3 Flight Software

Even with radiation tests that indicate that devices should remain functional, flight software should still be designed to operate with completely failed external devices, and to attempt to recover failed devices when possible. On ORCASat, this involves several aspects:

- Ensuring that failure of devices can be detected. Timeouts should be implemented on all communication interfaces to ensure software does not wait endlessly for a failed device. Where possible, flight software can estimate the data to be returned from a device, and compare to the actual returned data. Certain data patterns, such as all zeros or all ones, may indicate a device that has completely failed. On data storage components, checksums or CRCs can be used to detect corrupted memory and failed devices. This is especially important on devices that communicate to the main processor on interfaces where bus lockups or inactive slaves may not be easily detected, such as SPI.
- Attempting to recover failed devices. On ORCASat, the main processor is able to power cycle most devices that it communicates with. This hardware feature allows software to attempt to recover failing devices and log appropriate information about the failure. If recovery is successful, it may be necessary to reprogram corrupt register configuration values. Some device datasheets also list recovery sequences that may be executed in case of specific failures of the device. If recovery is not successful, reduced operation modes are entered.
- Ensuring that software can operate correctly, even with failed devices. Failed devices should place the software into a reduced operating mode, which should ideally still allow contact with the spacecraft. On ORCASat, software is designed to transparently handle the failure of hardware components. For example, failure of a telemetry storage chip will not cause the filesystem to hang. Instead, requests to save telemetry are accepted as normal but are just not executed at the lower level, allowing functionality to degrade gracefully.

### 6.4 Reliable Reset Source

Many embedded systems, not just limited to satellites, employ the use of a hardware watchdog to monitor and recover from malfunctions in a processor or program, leading to software lockups. The use of a watchdog can therefore help against SEUs by resetting the system if there is a detrimental effect impacting program execution.

In a concurrent system, the implementation of a watchdog must be carefully constructed to ensure all software lockups can be detected. If the software resetting the watchdog to its original count is implemented as a periodic task, it is possible for the watchdog task to preempt another task that has hung, resulting in the watchdog being pet even though a portion of the software has stalled. This can prevent the system from resetting even when large portions of software have locked up.

The use of a *task-aware watchdog* can solve this problem, and enable detection of all software lockups [19]. The design of this concept consists of a watchdog task that maintains the status of all other tasks in the code base. If a task has not reported its status in some time, it is possible that it has hung and the watchdog task will refrain from petting the watchdog, thus causing a system reset. This method results in a software monitor source that is responsive to all software lockups.

## 7. Lessons Learned

Many insights were uncovered from performing radiation testing on the critical components in the OBC system. There were also some lessons learned that were realized after completing the test.

### 7.1 Automation

One factor that we identified as being a good investment of time prior to testing was the automation of test sequences and procedures in software. As we had limited time in our testing, automation was the key to maximizing beam time and completing all of our experiments. We were able to transition between tests in the manner of a few minutes, and we did not have to reenter the irradiation room at any point.

### 7.2 Test Isolation

Due to limited time constraints, all three devices were tested one after the other using the same hardware. Devices on the board are in close proximity to one another, so the residual irradiation from one test could have impacted later tests. A better practice would have been to swap boards between each test with a clean board, however this was not feasible given our time frame or resources. Alternatively, components that were not the target of the test could have been physically shielded for protection.

Additionally, the board used for testing was not designed for the purpose of making testing easy. The current monitors on the board measured the current draw of all components being powered. This made it not possible to see changes in current for small devices, like the flash and RTC, because the change, if any, would be on the order of a few milliamps. There is also only one temperature sensor on the board, but it cannot measure the temperature of specific devices. It was not used during testing as it wasn't anticipated to provide useful information. Perhaps external thermocouples could have been used to measure the temperature of each device individually.

### 7.3 Firmware Functionality

There were a few things missing from the firmware that could have been tested. In the case of the NOR flash, it would have been useful to perform a register sweep every once in a while to discover any errors present. Although the focus of testing was to ensure that the memory retained any data it was given, registers of CMOS ICs are prone to upsets as well. This would have been useful information to collect.

With the RTC test, attempts were not made to assess any clock drift on the RTC, which could have occurred as a result of TID damage to the RTC's crystal [20]. The PCA2129 provides a clock output pin that may be used for this purpose on future tests.

In the testing of the TMS570, the testing was limited to simple functionality and only a small portion of the code base was executed. A variety of different tests could have been designed to target use of different hardware in the MCU, as is done in [4] and [5].

### 7.4 Log Messages

Firmware for the RTC tests logged detected upsets in a way that made data processing more difficult than it could have been. In the future, it would be useful to log raw register values instead of values that needed several levels of conversion. Additionally, registers could be scanned more frequently than once every two seconds. If multiple upsets changed and then restored a register bit within two seconds, the test firmware may have missed detecting both upsets.

## 8. Conclusion

Radiation testing is an important step in qualifying COTS components for space. The use of modern software tools can increase the efficiency of radiation testing, enabling the collection of more results and faster analysis after the fact. Once an understanding is formed regarding the functional impacts radiation has on these devices, software can be designed to create a robust system catered to handling these effects. In cases where radiation testing

is not possible or available, some of these same steps can still be taken to improve a system's reliability. In any case, these effects should be considered by designers of the system and the use of software mitigation can be an inexpensive way to deal with them.

## Acknowledgments

This work is supported by the generous funding of the Canadian Space Agency (CSA), as part of the Canadian CubeSat Project. Authors are grateful for the technical guidance, resources and time the CSA has donated to this project.

The authors would also like to thank the staff of the Proton Irradiation Facility at TRIUMF for the opportunity to use the facilities, and for their guidance during our experiments.

## References

- [1] S. Cole, "Small satellites increasingly tapping COTS components," *Military Embedded Systems*, 2015. [Online]. Available: <https://militaryembedded.com/comms/satellites/sm-all-tapping-cots-components>
- [2] D. Sinclair and J. Dyer, "Radiation Effects and COTS Parts in SmallSats," in *Proceedings of the 27th AIAA/USU Conference on Small Satellites*, 2013. [Online]. Available: <https://digitalcommons.usu.edu/cgi/viewcontent.cgi?article=2934context=smallsat>
- [3] F. Irom, "Guideline for Ground Radiation Testing of Microprocessors in the Space Radiation Environment," Jet Propulsion Laboratory, National Aeronautics and Space Administration, Pasadena, CA, 2008. [Online]. Available: <http://hdl.handle.net/2014/40790>
- [4] H. Quinn, A. Watkins, Y. Chen, T. Fairbanks, T. Shepard, and E. Raby, "Recent Results for Commercial Microprocessor Testing," in *2018 IEEE Radiation Effects Data Workshop (REDW)*, 2018, pp. 1–7.
- [5] C.-H. Lin, S.-C. Yang, K.-C. Han, Y.-C. Tsai, C.-Y. Pan, T.-C. Chao, and C.-C. Lee, "Verify the Radiation Performance of TI MCU TMS570LS3171 with Pencil Proton Beam Scanning in the Proton and Radiation Therapy Center of CGMH, Taiwan," in *2018 IEEE Radiation Effects Data Workshop (REDW)*. IEEE, 2018, pp. 1–3.
- [6] J. Schoolcraft, A. Klesh, and T. Werne, "MarCO: interplanetary mission development

- on a CubeSat scale,” in *Space Operations: Contributions from the Global Community*. Springer, 2017, pp. 221–231. [Online]. Available: <https://trs.jpl.nasa.gov/bitstream/handle/2014/46089/CL%2316-1589.pdf>
- [7] K. F. Strauss and T. Daud, “Overview of Radiation Tolerant Unlimited Write Cycle Non-Volatile Memory,” in *2000 IEEE Aerospace Conference*. Jet Propulsion Laboratory, National Aeronautics and Space Administration, 2000. [Online]. Available: <http://hdl.handle.net/2014/18640>
- [8] H. Quinn, Z. Baker, T. Fairbanks, J. L. Tripp, and G. Duran, “Software Resilience and the Effectiveness of Software Mitigation in Microcontrollers,” *IEEE Transactions on Nuclear Science*, vol. 62, no. 6, pp. 2532–2538, 2015.
- [9] Q. Martin and A. D. George, “Scrubbing optimization via availability prediction (SOAP) for reconfigurable space computing,” in *2012 IEEE Conference on High Performance Extreme Computing*. IEEE, 2012, pp. 1–6.
- [10] “PIF Beam Specifications,” TRIUMF. [Online]. Available: <https://www.triumf.ca/pif-beam-specifications>
- [11] A. Zoltan and R. Arthurs, “Development of Automated Testing Infrastructure for a CubeSat On-Board Computer,” in *Progress in Canadian Mechanical Engineering*, vol. 3, 2020. [Online]. Available: <https://library.upei.ca/islandora/object/csme2020%253A14>
- [12] D. L. Hansen, R. Hillman, F. Meraz, J. Montoya, and G. Williamson, “Radiation Performance of a Flash NOR Device,” in *2018 IEEE Radiation Effects Data Workshop (REDW)*, 2018, pp. 1–5. [Online]. Available: <https://ieeexplore.ieee.org/document/8584289>
- [13] “PCA2129 Automotive accurate RTC with integrated quartz crystal,” NXP Semiconductors N.V., 2014. [Online]. Available: <https://www.nxp.com/docs/en/datasheet/PCA2129.pdf>
- [14] “TMS570LS09x/07x 16/32-Bit RISC Flash Microcontroller Technical Reference Manual,” Texas Instruments Incorporated, 2018. [Online]. Available: <https://www.ti.com/lit/ug/spnu607a/spnu607a.pdf>
- [15] “Hardware Abstraction Layer Code Generator for Hercules MCUs.” [Online]. Available: <https://www.ti.com/tool/HALCOGEN>
- [16] E. W. Blackmore, “Operation of the TRIUMF (20-500 MeV) proton irradiation facility,” in *2000 IEEE Radiation Effects Data Workshop. Workshop Record. Held in conjunction with IEEE Nuclear and Space Radiation Effects Conference (Cat. No. 00TH8527)*. IEEE, 2000, p. 3. [Online]. Available: [https://www.triumf.ca/sites/default/files/nsrec\\_abs.pdf](https://www.triumf.ca/sites/default/files/nsrec_abs.pdf)
- [17] S. A. Asghari, H. Taheri, H. Pedram, and O. Kaynak, “Software-Based Control Flow Checking Against Transient Faults in Industrial Environments,” *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 481–490, 2014.
- [18] Y. Kovo, “Command and Data Handling,” National Aeronautics and Space Administration, 2020. [Online]. Available: <https://www.nasa.gov/smallsat-institute/sst-soa/command-and-data-handling>
- [19] N. Murphy, “Watchdog Timers,” in *Embedded Systems Programming*, 2000, pp. 112–124. [Online]. Available: <https://m.eet.com/media/1175014/f-murphy.pdf>
- [20] C. Renaudie, M. Markgraf, O. Montenbruck, and M. Garcia, “Radiation testing of commercial-off-the-shelf GPS technology for use on low earth orbit satellites,” in *2007 9th European Conference on Radiation and Its Effects on Components and Systems*. IEEE, 2007, pp. 1–8. [Online]. Available: [https://www.dlr.de/rb/Portaldata/38/Resources/dokumente/GSOC\\_dokumente/RB-RFT/RADECS\\_07.pdf](https://www.dlr.de/rb/Portaldata/38/Resources/dokumente/GSOC_dokumente/RB-RFT/RADECS_07.pdf)