# Nanosatellites in LEO and beyond:
## Advanced Radiation Protection Techniques for COTS-based Spacecraft

**PREPRINT VERSION**

Published article available at: http://dx.doi.org/10.1016/j.actaastro.2016.11.032

David Selčan*, Gregor Kirbiš, Iztok Kramberger
Faculty of Electrical Engineering and Computer Science, University of Maribor, Smetanova ulica 17, 2000 Maribor, Slovenia

*Corresponding author
E-mail addresses: david.selcan@um.si (D. Selčan)

**Abstract – This paper presents an approach for implementing radiation protection FDIR (Fault Detection, Isolation and Recovery) techniques designed especially for nanosatellites, capable of ensuring reliable operation in harsh orbits using COTS (Commercial off the Shelf) components. The radiation environment, as encountered by nanosatellites utilizing Flash-based FPGAs in orbits higher than Low Earth Orbit, is analyzed, primarily focusing on SEE (Single Event Effects). In order to assure reliable operation, the FDIR policy is split into two levels: the Low Level FDIR which ensures that no permanent damage occurs to the satellite's electronics, which then allows the use of a High Level FDIR, which is tasked with maintaining high availability. A hierarchical approach, consisting of three types of current limiters in combination with watchdog timers and fault tolerant logic implemented inside a flash-based FPGA is proposed for the Low Level FDIR. The impacts of various radiation-induced faults are analyzed with respect to how the FDIR techniques mitigate them. The proposed current limiters and watchdog timers have been implemented and evaluated for suitability of use with the hierarchical FDIR policy. In order to decrease the impacts on the size and weight footprints, the current limiters were implemented as stacked 3D modules.**

**Keywords: Nanosatellites, FDIR, COTS components, SEE, fault tolerant logic, over-current protection**

## 1. Introduction

In recent years a new trend has emerged in the design and verification of spacecraft. Rapid advances within the field of integrated electronics have enabled the use of inexpensive and highly performant electronic components, which have also found their way into the space industry. These so-called Commercial off-the-Shelf (COTS) components allow the constructions of spacecraft, specifically satellites, at significantly lower costs and development times. This has allowed small, interwoven teams (as typically found within university environments) to design nanosatellites from the initial concept stages to the final stages within drastically reduced time frames. These nanosatellites typically have a mass of less than 10 kg and occupy a volume of less than 8.4 $dm^3$.

Indeed, the popularity of this approach cannot be denied, with more than 90 different nanosatellites being launched in 2013 alone, with the number expected to increase even further in the following years [1]. Though their roots lie in university-based education and technology demonstration

missions, their usages since their inceptions have evolved into including science, remote sensing, telecommunication, and even commercial interests [2]. Indeed, perhaps the most important aspect of the nanosatellite approach is the possibility of launching a multitude of nanosatellites as a single satellite constellation, whereby these tasks previously thought of as too expensive could be accomplished (e.g. on-demand remote sensing, global monitoring, other real-time satellite applications).

Unfortunately, due to the history of nanosatellite development (university based mission with limited funding) and their format (severe mass and size constraints for most parts), little effort has been invested increasing their reliability. Specifically, most nanosatellites today feature little redundancy. Additionally, due to heavy uses of COTS components, they are typically unprepared for operating within an environment that includes higher amounts of radiation. The operations of nanosatellites within a relevant environment are also not usually verified thoroughly.

Though these limitations have already resulted in the failure of a couple of nanosatellite missions [3], their use in primarily Low Earth Orbit (LEO) meant that most missions proceeded without major problems even with the potential lack in reliability. However, talks are already underway to bring the nanosatellite platforms along even further by using them for interplanetary missions. In order for nanosatellites to still be practical beyond LEO, where the radiation environment and operational constraints are much harsher, one method would be to modify their designs to be more in line with how larger satellites are designed and built [4].

Such a change in the development approach for nanosatellites to a more formal, rigid process would detrimentally affect the more important aspects of nanosatellites: their low costs and short development cycles. To this extent, we propose an alternative approach for ensuring the reliabilities of nanosatellites. This approach is consistent with the nanosatellite design approach, meaning it is based on COTS components and imposes as little impact as possible on the mass and size of the satellite. This approach is based on extensive use of overcurrent-protection circuitry (called current limiters) and watchdog timers, and consists of two levels of Fault Detection, Isolation and Recovery (FDIR). The primary focus of this method is the prevention of faults caused by radiation (specifically Single Event Effects) but it is also useful for preventing other types of faults, including those caused by design failures. By following this approach, it is possible to increase the reliability of a nanosatellite operating in harsher conditions than those found in LEO.

The proposed FDIR technique is presented in the following format: Section 2 contains a survey of typical orbits of interest with regard to the radiation environment found in them, which is contrasted with a LEO orbit. Additionally, how the requirements for the satellite system can be estimated is presented by examining this radiation environment. Section 3 contains a detailed description of the proposed FDIR scheme, including the designs of the current limiters, watchdog timers, special consideration for logic design, and the implementation of more complex FDIR schemes on top of the one presented. Section 4 then analyzes how radiation affects the specific components of the implementation and how the proposed FDIR policy copes with these effects. Finally, Section 5 presents some of the more important performance characteristics, obtained by measuring an implementation of the proposed FDIR policy.

## 2. Space beyond Low Earth Orbit

Space is a harsh environment for any electronic system, but not all parts of space are equally hostile to electronics. For this reason, it is extremely important to be aware of the radiation environment present in the orbit into which a satellite is launched. For this purpose, the SPENVIS online tool (http://www.spenvis.oma.be/) was used for estimating the amount of radiation that would be present in each of the three distinct orbits, which could present the next step for nanosatellites. The estimates were compared to a reference 600 km, 97.8° inclination Sun Synchronous Low Earth Orbit (SSO), which is one of the more popular orbits where nanosatellites are presently being launched into at the time of writing of this paper. The three orbits evaluated were: a 0° inclination Geostationary Orbit (GEO), a Geostationary Transfer Orbit (GTO), which was simplified as an elliptical 0°inclination orbit with a perigee of 300 km and an apogee of 35,786 km, and a Lunar Transfer Orbit (LTO), which was also approximated as an elliptical 0° inclination orbit with a perigee of 300 km and an apogee of 384,400 km. A Lunar Orbit (LO) was not directly evaluated due to lack of simulation tools and the fact that the radiation present around the Moon is only slightly higher than a LEO with a high radius [5]. Though the transfer orbits' approximations are not directly applicable for a nanosatellite, due to limited propulsion options a part of a nanosatellite transfer orbit to either Lunar Orbit (LO) or GEO would have to travel through a similar radiation environment [6], at least for part of the way. Other SPENVIS parameters were left at their default settings, with the Total Ionizing Dose (TID) analysis performed on a model with finite aluminum slab shields.

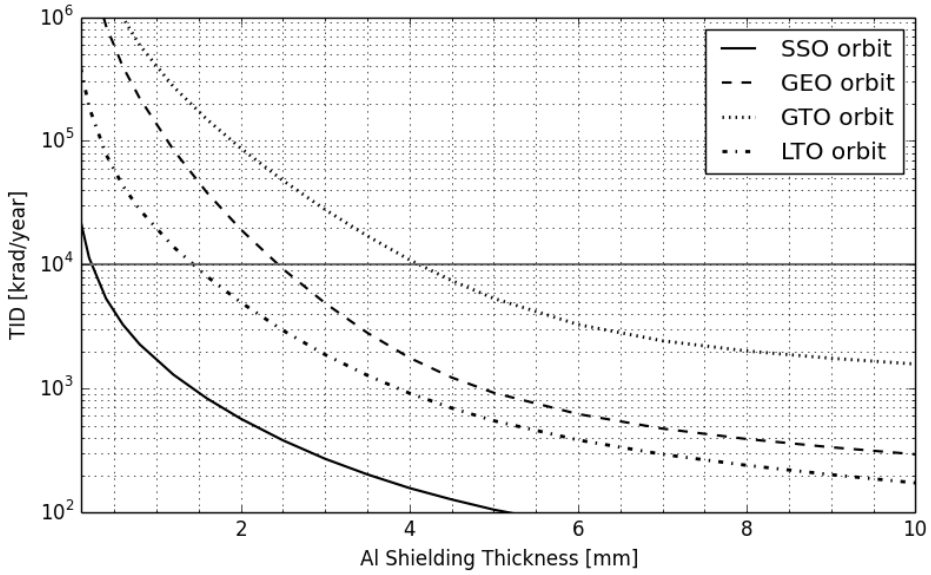


**Figure 1: Total Ionizing Dose for four reference orbits in relation to thickness of aluminum shielding.**

It can be seen immediately that, when compared to the reference SSO orbit, all three orbits had drastically increased radiation profiles. Since the proposed approach aimed at retaining one of the primary advantages of nanosatellites – their reliance on COTS components, electronic components on board such a nanosatellite must be shielded by aluminum in order to operate over extended periods of time. How much shielding is required is heavily dependent on the required life-time of the satellite as well as its design. Carefully selected COTS components can survive up to 30 krad [7], meaning that for a 3 year mission, a cut-off of 10 krad per year is selected. Following the results

presented in Figure 1: at least 1.75 mm of aluminum for a LTO, 2.25 mm of aluminum for GEO, and 4 mm of aluminum for GTO would be required.

One method of achieving this aboard a nanosatellite is, during the designing of nanosatellite electronics, to generate 3D models of the electronics, which are then used to mill out a block of aluminum to the required width, which is then fastened to the Printed Circuit Board (PCB) which also houses the electronic components. Though this could adversely impact the mass budget, it should be noted that a 4mm shield of size 10 cm x 10 cm weighs only 108 g. Further, due to the stackable nature of nanosatellite electronics, only the two boards at the extreme ends of the satellite would need to be shielded by the full amount. The details of how the internal nanosatellite electronic boards are affected are presented in [8–10], where it can be seen that the internal boards receive the full amount of radiation only at the edges, and even there it is reduced when compared to the boards in the extreme positions, meaning that less shielding is needed for them.

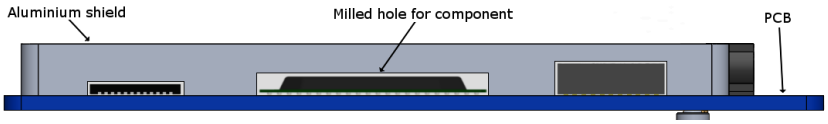


**Figure 2: An approach to shielding electronics on nanosatellites.**

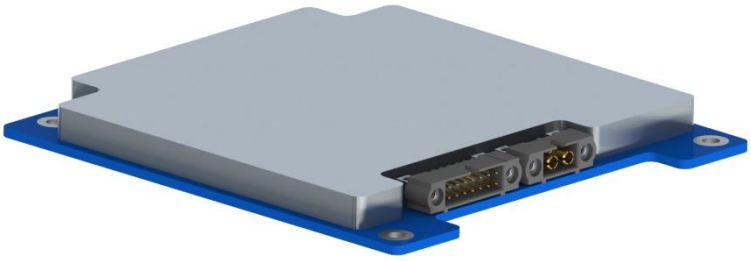

**Figure 3: 3D rendering of a shielded nanosatellite PCB example.**

While lowering the TID exposure of electronics is fairly straightforward, there are other radiation induced effects that are not as easily mitigated, especially with COTS components. Single Event Effects (SEE) occur when a high-energy particle strikes the active area of an integrated circuit, causing changes to the behavior of the electronic system or even permanent damage if not properly treated. For orbits where higher amounts of shielding are required these effects pose an even bigger threat, as the production of secondary particles [11] when high-energy particles collide with the atoms in the shield can increase the rate of SEE effects when compared to not shielding the electronics at all.

The only method for preventing failures caused by SEE is to implement proper design techniques to guard against them. The FDIR techniques presented in this paper are specifically aimed at presenting an approach to designing electronic systems, which are tolerant to most SEE effects caused by radiation. The following SEE effects were specifically considered [12]:

- Single Event Upsets (SEU), which are changes in the state of memory elements.
- Single Event Transients (SET), which are temporary changes in the electrical level of any signal path.

- Single Event Latch-ups (SEL), which cause low impedance current paths through an integrated circuit, causing permanent damage to a component if not mitigated and persisting until power cycled.
- Single Event Burnouts (SEB), Single Event Gate Ruptures (SEGR), which are destructive events occurring to a power MOSFET.
- Single Event Functional Interrupt (SEFI), which are events that result in a non-destructive interrupt to the operation of an electronic component, which persis t until power cycled.

# 3. Hierarchical Fault-tolerance

Over the last couple of years, there have been a couple of proposals for using nanosatellites in higher than LEO orbits. What is usually missing or is not yet fully defined due to the proposals dealing with other issues such as communication, propulsion, etc., is how such a nanosatellite would cope with the increased levels of radiation. For example, [13] presents an interplanetary nanosatellite concept for space weather monitoring but the radiation tolerance and FDIR techniques presented are limited to shielding and watchdog timers. Similarly, [14] shows how nanosatellites could be used for various interplanetary missions and even presents some techniques as to how reliable operation could be achieved, most notably the use of SEL immune parts, periodic resets, and robust software design techniques. Though the use of SEL immune parts would mitigate most of the radiation induced issues that might occur on such a satellite, they can be difficult to procure for most nanosatellite teams, due to costs constraints and various international trade restrictions.

A hierarchical approach to the FDIR policy was chosen in order to make the FDIR process more transparent and to decouple the design of the FDIR policy from other design requirements. The policy was split into two levels, similar to the method presented in [15], where one FDIR level, the vital layer, is responsible for enforcing the minimum required safety procedures, preventing permanent damage to the satellite and preventing it from becoming unresponsive. This level, which we named the Low-level FDIR, is primarily tasked with protecting the system from SEL events, and to restart those parts that fail in an unrecoverable way due to other SEE events. The other level, the nominal layer as presented in [15], is tasked with maximizing the performance and uptime of the system in the presence of other errors. This level, which we call the High-level FDIR policy, is tasked with mitigating these errors that occur due to SEE events and can be recovered from without affecting the operation of the satellite.

The exact scopes of both FDIR levels are determined by relying on the definition of functionality levels, as presented in [16], where three levels of functionality are defined: always-on functionality, mission-critical functionality, and non-critical functionality. Low-level FDIR policy is in charge of protecting the functionalities of all three categories, while the high-level FDIR policy is only responsible for the protection of certain parts of mission-critical functionalities and all non-critical functionalities. To illustrate how functionality can be grouped into the three defined categories:

- Always-on functionality – These are the parts of a satellite that the requirements specify must never be turned off. The electrical power generation, storage and distribution systems definitely fall into this category. The primary communication system (at least the reception part and the part that manages it) is also usually in this category. Needless to say, the Low-level FDIR policy is also in this category.

- Mission-critical functionality – This is the part of the satellite that is tasked with ensuring proper operation but will not result in mission failure if it is turned off temporarily. The less critical parts of the communication system, as well as the on-board data handling and attitude control might fall into this category.
- Non-critical functionality – This category is usually reserved for payloads and instruments, the inactivity of which can cause the loss of certain mission goals but does not result in a total loss of mission.

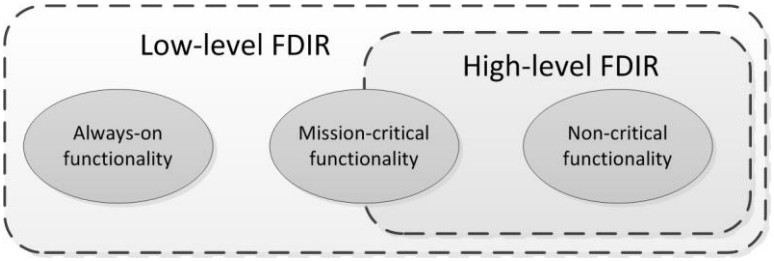


**Figure 4: Two levels of FDIR and their covered functionality.**

## 3.1. Low-level FDIR

The Low-level FDIR policy is primarily responsible for preventing permanent damage occuring to the spacecraft. A major part of this is ensuring that the COTS components used on a nanosatellite do not become permanently damaged through SEL or SEB/SEGR effects, or are permanently stuck in an unstable configuration due to other SEE effects. Depending on mission requirements, it can also be used to prevent certain actions being performed that could result in serious loss of functionality, though the extent of which is beyond the scope of this paper. In order to do these things, it relies heavily on the uses of current limiters, which are used to turn off or power cycle components and systems. Additionally, heavy usage of watchdog timers prevents a system from becoming completely unresponsive.

It must not be forgotten that the Low-level FDIR policy must also protect its own implementation against the same effects. This presents a sort of "chicken and egg" problem, which is dealt with in a very specific way – only those components which are inherently immune to SEE or can be easily made to be so are used for the implementation of Low-level FDIR. Specifically, this means relying on completely discrete, SEL tolerant analog components (SEB/SEGR tolerant P-type MOSFETS, Bipolar technology, latch-up hardened ICs) [7,17] for the most basic functionality and then linking them together in a flash based FPGA for the more complex parts. The use of an FPGA, in contrast with microprocessor circuits, allows fine-tuning the fault tolerance of the logic implementation, which simply cannot be done using COTS microcontrollers.

### 3.1.1. Over-current Protection

When a SEE effect occurs in a COTS component, its functionality is usually severely degraded or even completely disabled until the component is either reconfigured or power-cycled. Even more problematic is when a SEL event occurs on a low-impedance path, causing the component to draw high amounts of current and overheat, which can cause permanent damage. For this reason, over-current protection is needed for COTS components operating within an environment with significant levels of radiation, thus enabling them to be reset to a known initialized state and preventing components from drawing too much current and thereby overheating.

The traditional approach for nanosatellites has usually been to implement current limiters at the system or subsystem levels. For missions that rely on system level over-current protection, this method is usually implemented with current monitors which measure power to those individual subsystems that are being managed by a central unit, which then power cycles the whole satellite when an over-current event occurs [18,19]. Though this does provide rudimentary protection against SEE effects, individual components can still be damaged, since their individual current consumptions are not measured. Further, the system needs to be completely reinitialized whenever a failure occurs anywhere on the satellite.

Subsystem level over-current implementations usually function in a similar manner, though the power distributions to each system can be individually managed. This type of over-current protection can be implemented either as part of the power distribution subsystem [19–21], which means it possesses similar shortcomings to the system level over-current implementation, or it can be implemented as parts of individual subsystems [22,23], where the current into individual components can be more accurately monitored but the distributed nature can present a problem for the system-wide FDIR policy.

The specific implementations of the over-current protection circuits vary in published literature. A simple approach is the use of a poly-fuse [24] or PTC (Positive Thermal Coefficient) resistors [25]. Though this may be sufficient, the slow turn-off times of such passive elements can present a problem, as a component can be damaged before the protection element has time to limit the current. Another approach is the use of a power transistor, in combination with a comparator and microcontroller [26], which alleviates the turn-off time issue, but the SEE vulnerability of the microcontroller and comparator must be considered. The use of integrated high-side current switches is also an option, as presented in [27]. Here, the SEE vulnerability of the COTS integrated switch must also be considered carefully.

The approach presented in this paper is a combination of the best properties of the previously listed methods, building upon the approach presented in [25], where different local and global current limiters are used for different purposes. Three different types of current limiters have been identified, each with a different purpose. They can be used together to protect most of the functionalities of a spacecraft from harm by SEE effects. Discrete, analog components are used, which are inherently tolerant to SEE effects. The number of components is minimized as much as possible to reduce impacts on mass and volume – for this purpose logic gates and comparators are replaced with similarly functioning FET driver ICs, which are inherently latch-up tolerant and consume less space.

Though the presented approach mitigates most destructive SEL effects, there exist components that can fail even when protected with the presented current limiting approach by experiencing a destructive SEL. If this level of risk is unacceptable for a mission, redundant components can be used or a proton irradiation screening test on the components can be applied [28].

#### *3.1.1.1. Controlled Current Limiter*

The first type of current limiters is used to protect single components from radiation induced (and other) failures. Since each subsystem of a nanosatellite can contain a large number of such components, the primary design constraint of this type of current limiter is that its footprint (both current consumption and actual size) must be as small as possible. For this purpose, a design which

relies on a current monitor, a MOSFET driver (used also as a comparator), a P-type MOSFET, some additional passive components and a control circuit inside an FPGA is used.

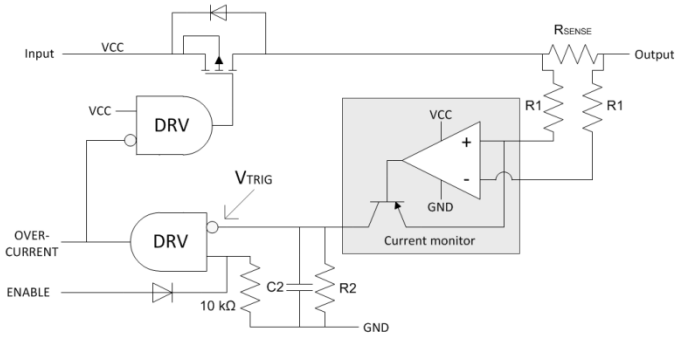


**Figure 5: Controlled Current Limiter Implementation.**

The principle of operation relies on the fact that once the output of the current monitor increases beyond the threshold value of the MOSFET drive, its output is inverted, which temporarily disables the MOSFET transistor. The FPGA logic circuit can then use this input to turn-off the ENABLE line, driving the MOSFET permanently into a disabled state. Care must be taken at turn-on, so that the initial high-current transient event has time to settle before the FPGA starts monitoring the input line. The response time (the time it takes the current limiter to switch the power off when an overcurrent condition occurs) can be set by varying the C2 capacitor. The current limit of this circuit is determined by the following equation:

$$I_{LIMIT} = \frac{V_{TRIG} \cdot R_1}{R_{SENSE} \cdot R_2} \quad (1)$$

### *3.1.1.2. Autonomous Current Limiter*

Since the Controlled Current Limiters rely on the proper operation of an FPGA circuit, it stands to reason that the FPGA circuit (and associated power circuitry) must also be protected by a current limiter in some capacity. This brings forward an already discussed "chicken and egg" problem. As such, a current limiter is needed which does not need an outside component for its control. An implementation can be used which adds a couple of passive component and two additional MOSFET drivers to the Controlled Current Limiter.

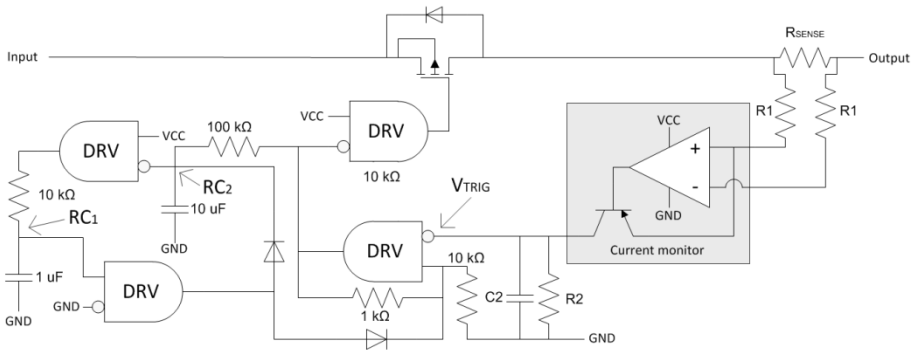


**Figure 6: Autonomous Current Limiter Implementation.**

The principle of operation here is that a resistor is used to connect the output of the MOSFET driver (which is used to compare the output from the current monitor) to its own non-inverting input (see the 1 kΩ resistor in Figure 6). This causes the circuit to remain in a disabled state whenever an over-current condition occurs. The two additional MOSFET drivers are used to re-enable the circuit after a set period of time and force it in an enabled state for a short period of time for the transients to settle. If multiple current limits must be set for different parts of the FPGA circuit, multiple current monitors can be used, with their output wired to the R2 resistor, to protect each part individually. This sums the respective currents, with weights determined by the R1 resistors. The current limit equation is the same as for the Controlled Current Limiter, while the two RC (RC1 controls forced-on time, RC2 controls re-enable time) constants are used to control the maximum "re-enable" and "force-enable" times:

$$t = RC \cdot \ln\left(\frac{VCC}{VCC - V_{TRIG}}\right) \quad (2)$$

### *3.1.1.3. Power Distribution Current Limiter*

The final current limiter type is coupled with the primary FPGA, which controls the Low-level FDIR policy and is as such used to control the power distribution to various subsystems. As such, its primary purpose is to prevent large amounts of current from being drawn by a single subsystem, thereby causing the voltage of the power source to drop below the operating range of the FDIR circuitry, thereby causing the whole satellite to enter an unstable state. Additionally, since the FDIR policy must be aware of the actual power consumption of all subsystems, it includes a Delta-Sigma modulation circuit for enabling the measurements of the current consumptions of each subsystem (an additional Delta-Sigma circuit can be used to also monitor the output voltage, for even more fault detection capability). As such, instead of using MOSFET drivers to drive the P-type MOSFET, it uses an Operational Amplifier (OP) to drive the transistor in a linear mode, which functions as a true current limiting circuit.

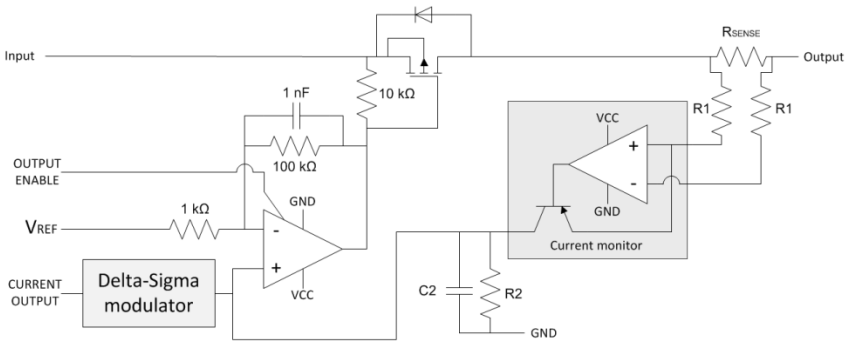


**Figure 7: Power Distribution Current Limiter Implementation.**

In order to prevent any thermal damage to the MOSFET due to this type of operation, the FPGA which controls the Power Distribution Current Limiters must disable the OP when an over-current condition occurs. Since the FPGA performs an Analog to Digital Conversion of the output current, the current limit, where this occurs, can be set arbitrarily. The analog current limit of this circuit is determined by the following equation:

$$I_{LIMIT} = \frac{V_{REF} \cdot R_1}{R_{SENSE} \cdot R_2} \quad (3)$$

### 3.1.2. Fault tolerant FPGA logic implementation

Since a critical part of the operations of the previously mentioned current limiters – as well as most functionalities of the satellite rely on the proper operation of an FPGA, it stands to reason that the FPGA must also be tolerant to SEE, specifically SEU, SET and SEFI (SEL effects are primarily mitigated by Autonomous Current Limiters). The first issue is that the FPGA chip itself must possess enough radiation tolerance – for this purpose, Flash-based or Anti-fuse-based FPGAs are most suitable for nanosatellites, as they are highly immune against SEE effects in the logic configuration – which can be a major issue in SRAM based FPGAs.

Another important aspect is that the logic implementation inside must also be fault tolerant. One of the more heavily used primary methods to mitigate such faults is Triple Modular Redundancy – this is a method of mitigating errors by triplicating all elements of a system and voting on each action then deciding on the option where two or more elements agree.

There are many possible ways that a TMR strategy can be implemented – the implementations mostly distinguish themselves by implementation complexity and resilience to faults. Many TMR methods exist [29], such as Block TMR, where the whole logic chain is triplicated, with voting only at the outputs, Local TMR, where individual logical blocks are triplicated, with voting at each individual block, Global TMR, which is an extension of Local TMR, but with everything (including I/Os, clocks, reset lines, even the voter circuits) triplicated and Distributed TMR which is similar to Global TMR, without replicating the clock and reset lines. Numerous implementation options exist as well, as can be seen in [30].

The difficulty in implementing effective TMR on nanosatellite missions does not lie in the difficulty of implementation itself, but rather in the lack of support for it in FPGA vendor tools for COTS FPGAs. For this purpose, a specific TMR implementation was used, which provides good fault protection against the more common or dangerous SEE effects, while being simple enough to be synthesized with FPGA vendor tools for COTS FPGAs,.

The proposed method is a variation of the Block TMR approach, where the whole logic implementation, including clocks, reset circuitry, and I/Os are triplicated. The I/Os are triplicated following the approach outlined in [30] – three outputs are joined together outside the FPGA, and a voting circuit is used to toggle the enable line of each output, while the inputs are either wired directly to each logic system's instance, or are registered and then voted upon by three voter instances, if synchronization is an issue. This eliminates any single SEU and SEFI, as well as most SET effects that can cause functional failures.

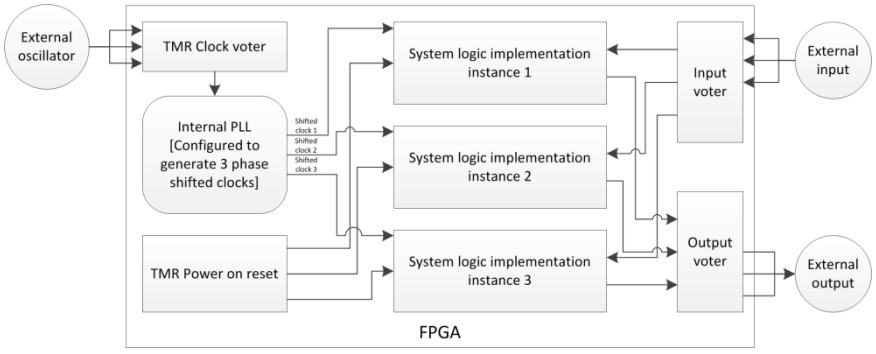


**Figure 8: Simplified TMR approach for COTS-based FPGAs.**

A method known as Temporally-Redundant TMR is used [31] to completely prevent SET effects from affecting the FPGA system. This avoids the need for modifying the place and route procedures to prevent the propagation of SET upsets [32]. Instead, shifted clocks are used to clock each logic instance. The shifted clocks are generated by the PLL circuits present inside the FPGA system, which can also be triplicated if deemed necessary.

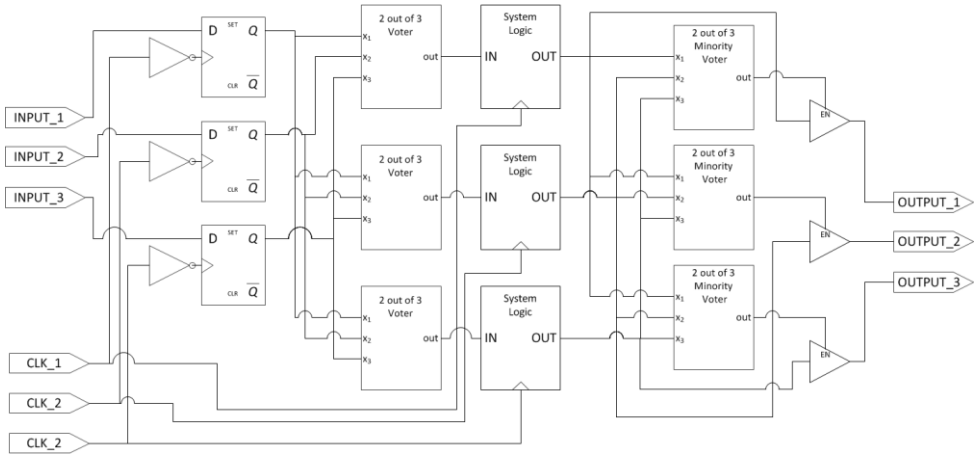


**Figure 9: Connecting System Logic instances to FPGA I/Os.**

An additional approach traditionally used is to implement Error Detection and Correction on all memory cells/devices that are used in a system. There are many such possible implementations (e. g. [33]) and as such, the details are beyond the scope of this paper.

### 3.1.3. Watchdog timers

In order to further secure the FPGA and other parts of the subsystem against SEE effects and other faults, specifically multiple ones in case they propagate in multiple logic instances, watchdog timers are used to perform a power cycle of the FPGA circuit and therefore the whole subsystem (or satellite, if the Primary FPGA is power cycled). Two types of Watchdog timers are used for this FDIR approach: FPGA Watchdogs and Analog Watchdogs. FPGA Watchdogs, which are implemented inside the Primary FPGA, are used to keep track of all other FPGAs (and therefore subsystems) on a satellite. This is done via common communication interfaces, where periodic messages are transmitted to the Primary FPGA. Each time the Primary FPGA receives such a message, the watchdog timer for that subsystem is reset. If a message is not received within a certain amount of time, the Primary FPGA initiates the restart of that subsystem and eventually turns it off.

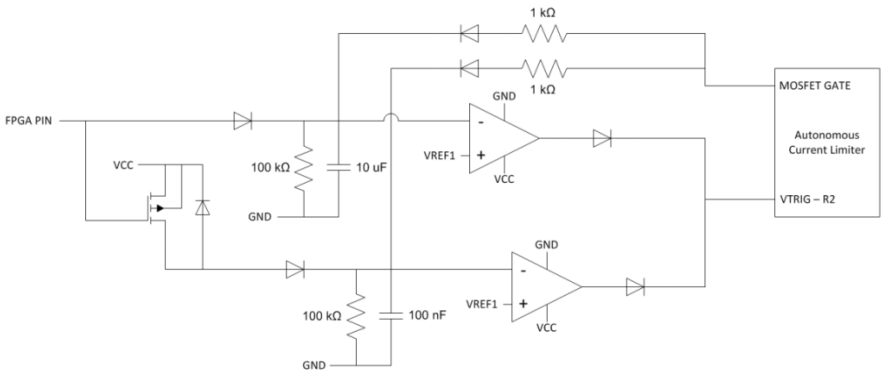


**Figure 10: Analog Watchdog timer implementation.**

Here another aspect of the aforementioned "chicken and egg" problem can be seen, namely the Primary FPGA cannot act as its own Watchdog timer. A custom built Analog Watchdog, which only relies on a transistor, passive elements and two OPs, is used for this purpose. The Primary FPGA must periodically pulse this circuit, otherwise (in case the line is stuck either in asserted or de-asserted positions for extended periods of time) it asserts the reset line, which is wired with the Primary FPGA Autonomous Current Limiter, causing the whole satellite to be power cycled.

## 3.2. Current Limiter Implementation

Since a typical nanosatellite subsystem requires a significant number of current limiters (especially Controlled Current Limiters), it makes sense to minimize their sizes and weights as much as possible. In order to do so, the three current limiter types were implemented as 3D stacked circuits (consisting of two PCBs). The two PCBs are interconnected with soldered wires and an epoxy is poured over all components, increasing the mechanical stability of the current limiter implementations. Though this method of implementation does increase the weight and height of the current limiter's implementations, the decrease in the required area on the PCB is significant. This additionally allows the current limiters to be tested individually before being assembled as part of a nanosatellite subsystem.

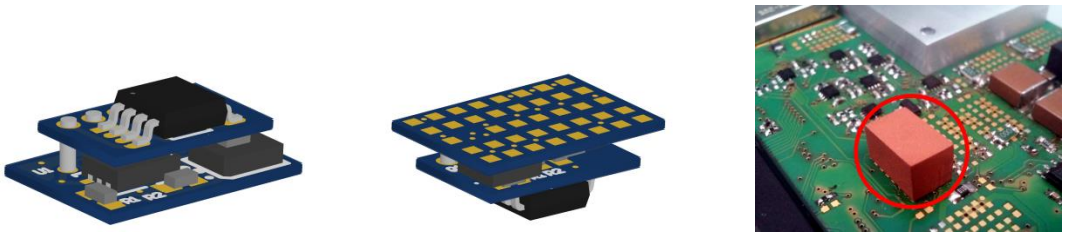

**Figure 11: 3D stacked Controlled Current Limiter implementation (left – 3D model top view, middle – 3D model bottom view, right – finished module placed on subsystem).**

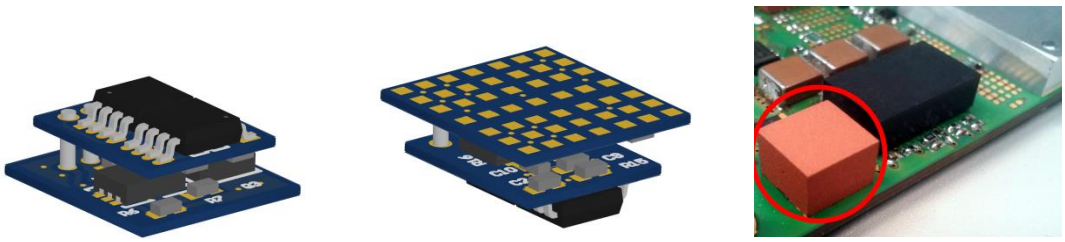

**Figure 12: 3D stacked Autonomous Current Limiter implementation (left – 3D model top view, middle – 3D model bottom view, right – finished module placed on subsystem).**

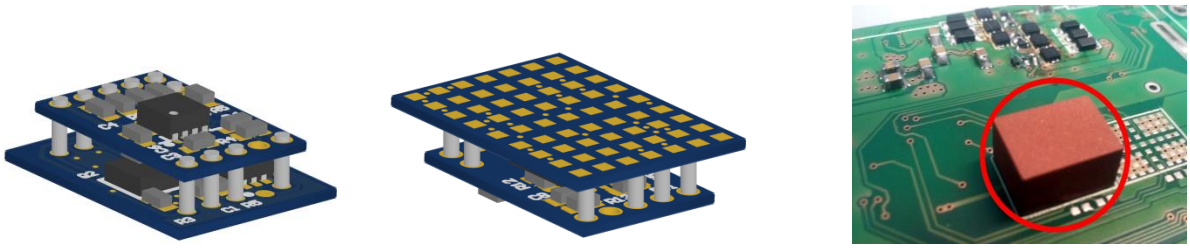

**Figure 13: 3D stacked Power Distribution Current Limiter implementation (left – 3D model top view, middle – 3D model bottom view, right – finished module placed on subsystem).**

**Table 1: Size and mass of 3D stacked current limiter implementations**

| Current limiter type | Weight | Size (Length x Width x Height) |
|---|---|---|
| Controlled Current Limiter | 0.5 g | 8.6 mm x 5.1 mm x 5.3 mm |
| Autonomous Current Limiter | 0.7 g | 8.3 mm x 7.8 mm x 5.2 mm |
| Power Distribution Current Limiter | 0.9 g | 10.2 mm x 7.9 mm x 5.0 mm |

## 3.3. Failure Analysis

When all the methods listed above are combined, the Low-level FDIR policy is split into manageable parts, implemented hierarchically. The autonomous current limiters are used to protect their respective FPGAs, with the Low-level FDIR FPGA being protected by a Hardware Watchdog, while other FPGAs can be protected by a Watchdog inside the primary FPGA. Each subsystem is then powered by a Power-distribution Current limiter, which is how the Low-level FDIR policy can monitor the states of all subsystems. Finally, each SEL (or otherwise) vulnerable COTS component on each subsystem is protected by a Controlled Current limiter.

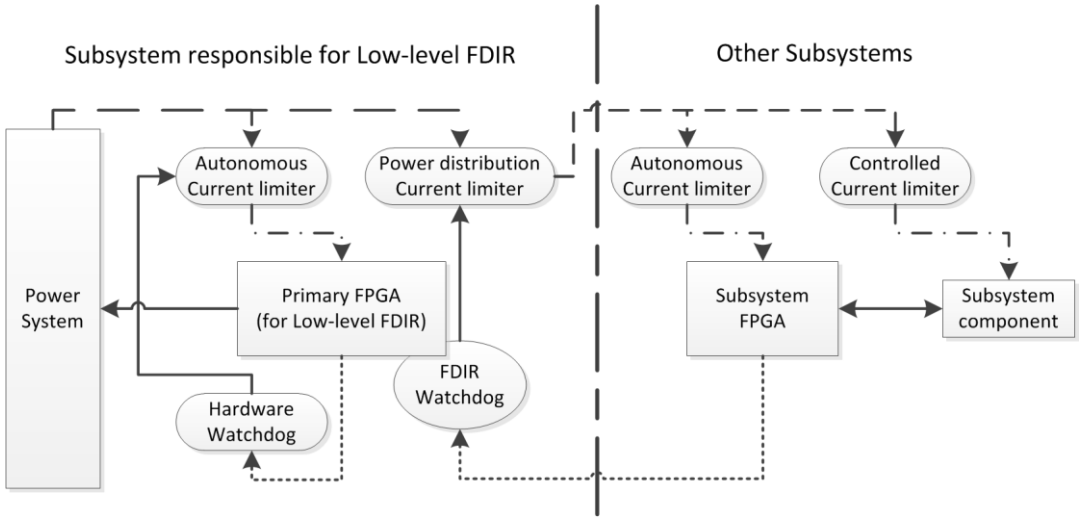


**Figure 14: Two levels of FDIR and their covered functionalities.**

How this approach mitigates all SEE is presented in Table 2, where the mitigation process is presented, with emphasis on the SEU, SET, SEL and SEFI effects. SEB/SEGR events were not evaluated, as they apply only to power MOSFETs. While the implementations presented here do make use of power MOSFETs, only P-type MOSFETs are used, where SEB effects are not applicable, and as the paper deals with nanosatellite electronics, the voltages on these transistors are fairly low (below 24 V), where SEGR effects do not usually occur. It should be noted that SEL effects can be either destructive or non-destructive (which are usually not detected by current limiters) but for the ease of analysis the non-destructive SEL effects were merged into the SEFI category, as they behave very similarly to one another, and the method of prevention (power cycling) is the same. Similarly, the FPGA categories have been split into multiple parts for analysis, namely the FPGA logic (registers and asynchronous gates), FPGA power regulator, FPGA I/O, and the FPGA PLL.

Obviously, the Power System itself must be designed in such a way as to be immune to SEE effects that might cut off power to the Primary FPGA. Since the Power System on a nanosatellite usually consists of a battery pack, which is connected directly to the Primary FPGA and its power regulation

circuitry (which is already protected), this should not present a problem for most nanosatellite designs.

**Table 2: List of vulnerable parts and the method of SEE mitigation**

| | SEU | SET | SEFI | SEL |
|---|---|---|---|---|
| **Hardware watchdog** | N/A | Possible reset of whole satellite to known state. | N/A | Uses only SEL tolerant components. |
| **Autonomous Current Limiter** | N/A | Possible reset of whole satellite to known state. | N/A | Uses only SEL tolerant components. |
| **Primary FPGA power regulator** | Possible loss of power to FPGA – Hardware watchdog performs reset. | Possible loss of power to FPGA – Hardware watchdog performs reset. | Possible loss of power to FPGA – Hardware watchdog performs reset. | Protected by Autonomous Current Limiter. |
| **Primary FPGA logic element** | TMR voting assures no change in logic functionality. | Temporally redundant TMR voting assures no change in logic functionality. | TMR voting assures no change in logic functionality – in case of accumulation, Hardware watchdog performs reset. | Protected by Autonomous Current Limiter. |
| **Primary FPGA I/O** | N/A | All I/Os are tippled, other two I/Os prevent it. | All I/Os are tippled, loss of one does not cause a change in functionality. | Protected by Autonomous Current Limiter. |
| **Primary FPGA PLL/Clock** | N/A | Temporally redundant TMR voting prevents propagation. | Hardware watchdog performs reset. | Protected by Autonomous Current Limiter. |
| **Power Distribution Current Limiter** | N/A | Possible power failure on Subsystem. FPGA Watchdog performs reset of Subsystem. | N/A | Uses only SEL tolerant components. |
| **Watchdog communication interface** | Possible loss of functionality – FDIR watchdog performs reset of subsystem. | Possible loss of functionality – FDIR watchdog performs reset of subsystem. | Possible loss of functionality – FDIR watchdog performs reset of subsystem. | Protected by Controlled Current Limiter, if applicable. |
| **Subsystem FPGA power regulator** | Possible loss of power to FPGA – FDIR watchdog performs reset. | Possible loss of power to FPGA – Hardware watchdog performs reset. | Possible loss of power to FPGA – Hardware watchdog performs reset. | Protected by Autonomous Current Limiter. |
| **Subsystem FPGA logic element** | TMR voting assures no change in logic functionality. | Temporally redundant TMR voting assures no change in logic functionality. | TMR voting assures no change in logic functionality – in case of accumulation, FDIR watchdog performs reset. | Protected by Autonomous Current Limiter. |
| **Subsystem FPGA I/O** | N/A | All I/Os are tippled, other two I/Os prevent transient effect. | All I/Os are tippled, loss of one does not cause a change in functionality. | Protected by Autonomous Current Limiter. |
| **Subsystem FPGA PLL/Clock** | N/A | Temporally redundant TMR voting prevents propagation. | FDIR watchdog performs reset. | Protected by Autonomous Current Limiter. |
| **Controlled Current Limiter** | N/A | Possible power failure on component – Subsystem FPGA identifies error and acts accordingly. | N/A | Uses only SEL tolerant components. |
| **Subsystem component** | Subsystem FPGA identifies loss of functionality and power cycles via the Controlled Current Limiter. | Subsystem FPGA identifies loss of functionality and power cycles via the Controlled Current Limiter. | Subsystem FPGA identifies loss of functionality and power cycles via the Controlled Current Limiter. | Protected by Controlled Current Limiter. |

Since the system functions are dependent on the correct functioning of the protection circuitry, even though the analysis presented in Table 1 provides an assurance that single points-of-failure will not affect the system, there still remains a possibility that a fault occurs on the one of the protection circuits. Though the use of the presented circuitry will improve the reliability of the system over

current state-of-art methods, if even a small possibility of failure is not acceptable for a mission, there remains the possibility of using redundant systems together to prevent possible failures in a protection circuit to cascade into other parts of the system.

### 3.4. High-level FDIR

An important part of the FDIR approach is how to handle reactivating power to subsystems, as well as when certain parts of subsystems (or even whole subsystems) should be powered off. Most approaches either handle this manually via remote operation [34] or feature an automatic restart, implemented in a microcontroller. The Low-level FDIR uses the three types of current limiters to split the de-activation and re-activation actions into two categories, which coincide with the Low and High levels of FDIR. The parts of the satellite which fall within the scope of only the Low-level FDIR policy are reactivated automatically after a set period of time (approximately 100 ms) and are never de-activated automatically except in cases of faults or errors. The Autonomous current limiter and the Controlled current limiter, which is configured to reactive automatically, are used for this process. Parts of the satellite's functionalities protected by them usually have a low restart time and as such no major interrupts are expected to occur to the operation of the satellite.

Other parts of the satellite are usually more complex – the errors that occur are also more predictable and cannot usually be mitigated using a restart. Such systems fall under the scope of the High-level FDIR policy. The primary difference between it and the Low-level FDIR policy is that due to the more complex issues involved, it cannot be simply implemented as a "detect failure then restart" process. Instead, the High-level FDIR policy should be implemented as software inside the (primary) on-board computer (or as part of a reconfigurable FPGA, as presented in [35]), an example of how a High-level FDIR policy can be implement is found in [36]. Controlled Current Limiters, which are not automatically reactivated and Power Distribution Current Limiters are used for this purpose. The specifics of a given High-level FDIR policy are heavily mission-defendant and as such the details of its implementation are beyond the scope of this paper.

## 4. Tests and Measurements

We used a set of FPGA cores for all tests and measurements. Firstly, the control circuit for the Controlled Current Limiter illustrates the proposed TMR method. The circuit contains a D-Register to enable or disable the power to the protected device, a counter, which is used to force the power during the initial transient period, and a couple of logic gates, which detect any over-current condition and switch the power off. The device can then be restarted by an external circuit.

For this circuit, only the SENSE and POWER signals (the SENSE pin is the input, which signals an overcurrent condition and the ENABLE is the output which controls the state of the current limiter) are connected to I/Os and as such, care must be taken as to how to connect them inside the FPGA. The power pin should be connected to an output voter as presented in Figure 9, while the SENSE input can be either connected directly or through the use of the circuit presented in Figure 9. Other pins are internal and are connected to each separate instance. The clocks should be shifted and generated by a PLL with individually settable delays, while the reset lines should be synchronous to this clock. The START input itself is wired internally, so it is separately connected to each instance.

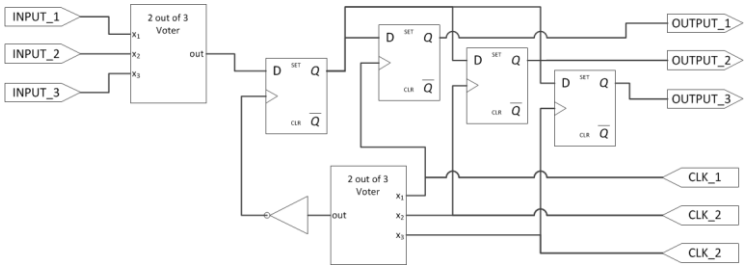


**Figure 15: Controlled Current Limiter logic implementation.**

Additionally, two separate cores for the Power Distribution Current Limiter were used. The first implements a first-order Delta-Sigma modulator, while the second uses a third-order Delta-Sigma modulator. Finally, the watchdog logic circuit makes use of a simple counter, which toggles the outputs periodically to trigger the external hardware watchdog.

For these FPGA core implementations, the logic utilization was analyzed for two different Flash-based FPGAs (the IGLOO2 family and the ProASIC3 family), contrasting the single implementations with the proposed TMR methods. It can be seen from the results presented in Table 3 that in addition to the 300% overhead taken by the triplication of the logic, the additional overhead imposed by the TMR methods is quite small – 7% and 5% of the single implementation respectively for the combinatorial logic elements, and 3% of the single implementation in both cases for the sequential logic elements.

**Table 3: FPGA Logic resource utilization.**

| 1st row: LUTs 2nd row: DFFs | Reset and clocking circuitry | Watchdog Timer circuitry | Controlled Current Limiter | Power Distribution Current Limiter (1st order ADC) | Power Distribution Current Limiter (3rd order ADC) | Total | Total (Percentage of Single) |
|---|---|---|---|---|---|---|---|
| **IGLOO2 FPGA** | 0 | 33 | 72 | 113 | 253 | 471 | 100% |
| **Single logic** | 3 | 16 | 37 | 93 | 325 | 474 | 100% |
| **IGLOO2 FPGA** | 4 | 105 | 222 | 348 | 767 | 1446 | 307% |
| **TMR logic** | 12 | 51 | 114 | 282 | 978 | 1437 | 303% |
| **ProASIC3 FPGA** | 0 | 48 | 171 | 173 | 1293 | 1685 | 100% |
| **Single logic** | 3 | 16 | 37 | 93 | 326 | 475 | 100% |
| **ProASIC3 FPGA** | 13 | 150 | 531 | 524 | 3928 | 5146 | 305% |
| **TMR logic** | 12 | 51 | 114 | 282 | 981 | 1440 | 303% |

In order to evaluate the functionality of the proposed FDIR approach a test was performed under simulated conditions on all the types of current limiters together, including the Analog Watchdog. The parts used to implement the current limiter and watchdog timer can be seen in Table 4. We specifically chose parts that are inherently tolerant to SEE, to perform tests on hardware that could be used in a nanosatellite. Passive components, which are normally not susceptible to radiation, are not listed.

**Table 4: Parts used for current limiter and watchdog timer implementations.**

| Component type | Part used | Rationale |
|---|---|---|
| MOSFET | SIA483DJ | COTS P-type MOSFETS are not susceptible to latch-up. |
| FPGA | IGLOO2 M2GL010 | Flash-based FPGA. |
| Current Monitor | LT6105 | Commercial version of radiation-hardened part. |
| Inverting MOSFET Driver | UCC27423 | MOSFET drivers are designed to be resistant to latch-up. |
| Operational Amplifier | OPA835 | Operational amplifiers from TI, produced on a BiCOM-3 SOI process – immune to latch-up. |

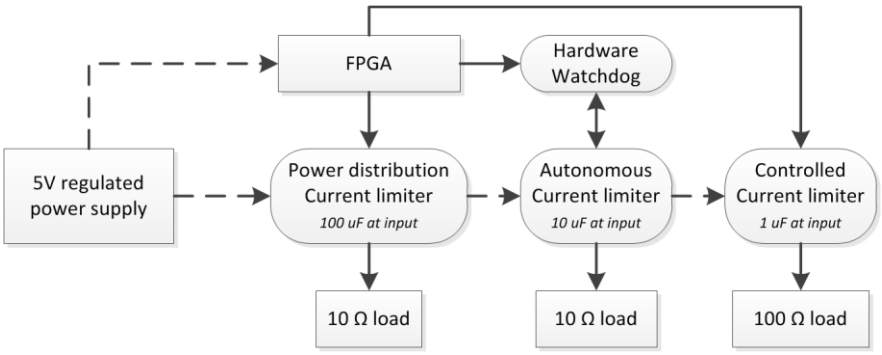


**Figure 16: Functional measurements and testing setup.**

The Power Distribution Current Limiter was used to power a load and the Autonomous Current Limiter, which in turn was used to power a load and the Controlled Current Limiter and its load. Everything was controlled by a single FPGA powered separately (due to the Power Distribution Current Limiter) and the watchdog was also evaluated alongside. The current consumption was simulated to be approximately 1A in total – 50 mA load for the Controlled current limiter and 500 mA loads for the other two current limiters. The output of each current limiter was then short-circuited to ground by a silicon diode, which mimics the effect of a SEL induced short-circuit event. The FPGA functionality was also disabled to evaluate the functionality of the external watchdog.

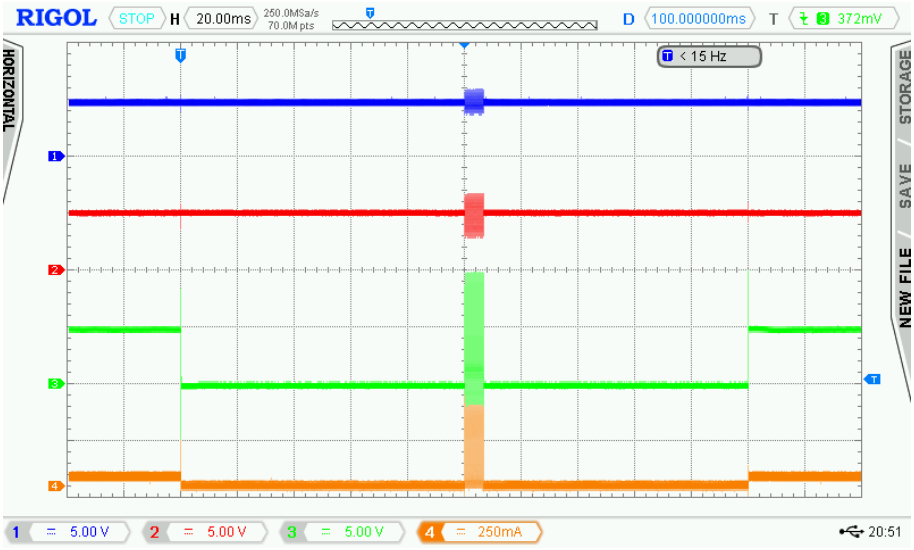


**Figure 17: Short circuit on load of Controlled Current Limiter – duration from short-circuit to recovery**

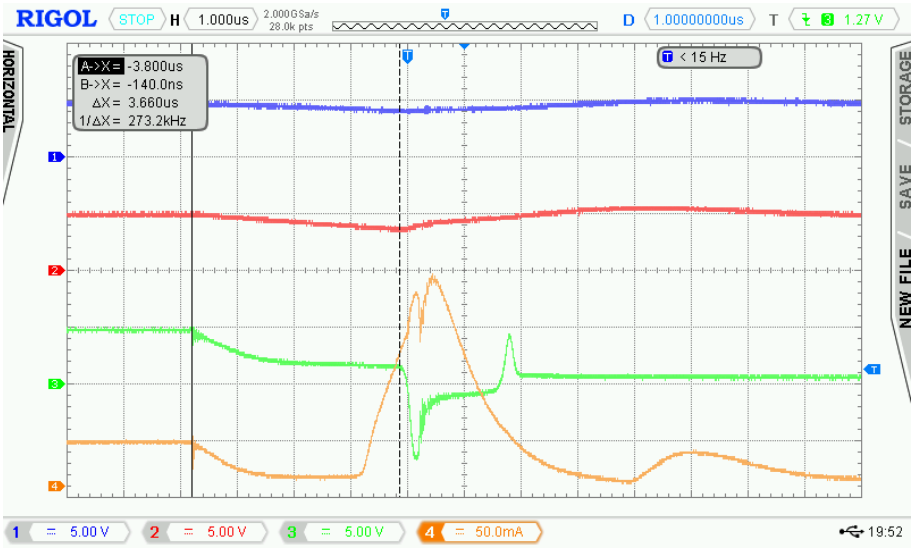


**Figure 18: Short circuit on load of Controlled Current Limiter – moment of power disconnect**

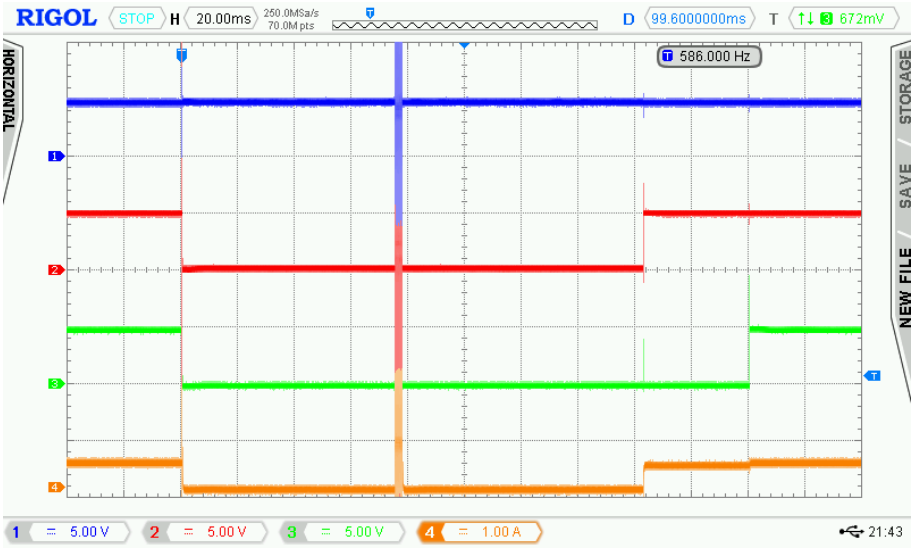


**Figure 19: Short circuit on load of Autonomous Current Limiter – duration from short-circuit to recovery**

**Figure 20: Short circuit on load of Autonomous Current Limiter – right – moment of power disconnect).**

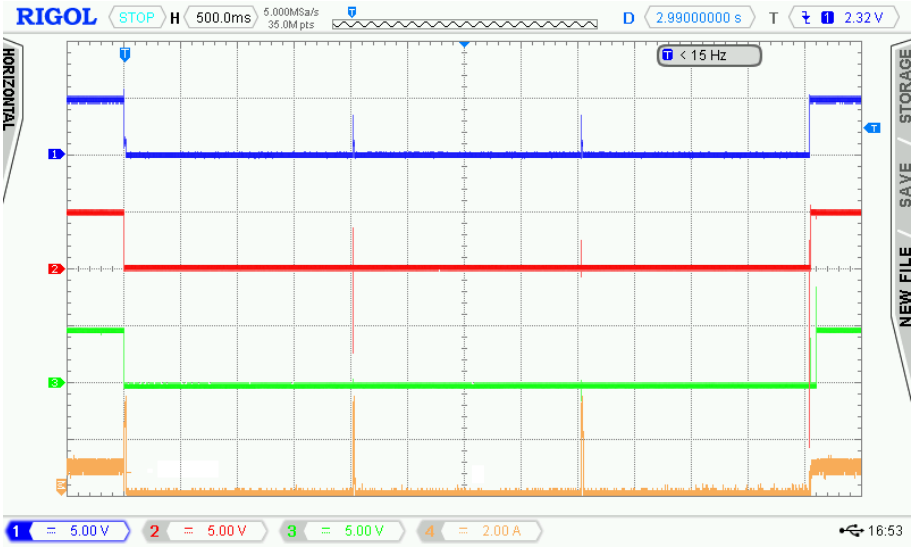


**Figure 21: Short circuit on load of Power Distribution Current Limiter – duration from short-circuit to recovery**

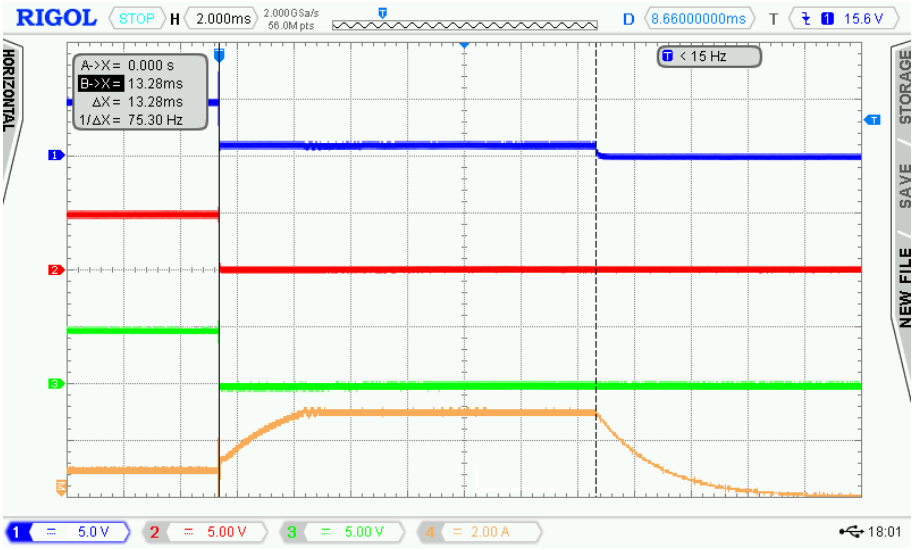


**Figure 22: Short circuit on load of Power Distribution Current Limiter, 1st order Delta-Sigma – moment of power disconnect**


**Figure 23: Short circuit on load of Power Distribution Current Limiter, 3rd order Delta-Sigma – moment of power disconnect**

The graphs presented in Figure 17 to Figure 25 show the output voltages of the individual current limiting circuits (the upper plot (line 1) presents the output from the Power Distribution Current Limiter, the upper middle plot (line 2) presents the output from the Autonomous Current Limiter, the bottom middle plot (line 3) presents the output voltage of the Controlled Current Limiter and the bottom plot (line 4) presents the measured current of the relevant Current Limiter under test.

It can be seen from the results of the functionality test, that the power distribution is handled hierarchically – the faults do not propagate up the chain. Additionally, the reaction time (the time that must pass for the power to be switched off after an overcurrent condition occurs) can be measured: the values are 3.66 us for the Controlled Current Limiter, 12 us for the Autonomous current limiter, 13.28 ms for the Power Distribution Current Limiter with a first-order Delta-Sigma modulator and 1.46 ms for the Power Distribution Current Limiter with a third-order Delta-Sigma modulator. Here it should be noted that the Power Distribution Current Limiter also limits the input current to a set value, which means that the relatively longer fault condition does not present an issue. Further, in certain Figures it can be seen that the fault conditions do cause voltage spikes that propagate up the hierarchy. The reason for this is that the tests were performed with low capacitances on each current limiter and load (a sort of worst case scenario). In practice this would not occur, as larger bypass capacitors would be used as part of best practice design techniques. It can also be seen from Figure 24 that if the FPGA ceases to trigger the Watchdog, its power is cycled, which restores functionality.

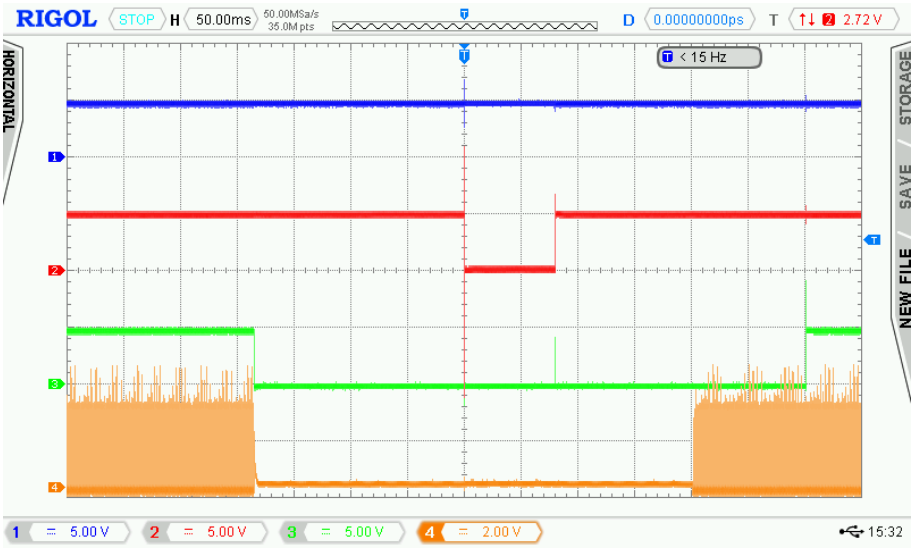


**Figure 24: Loss of watchdog timer trigger – fault on FPGA I/Os**

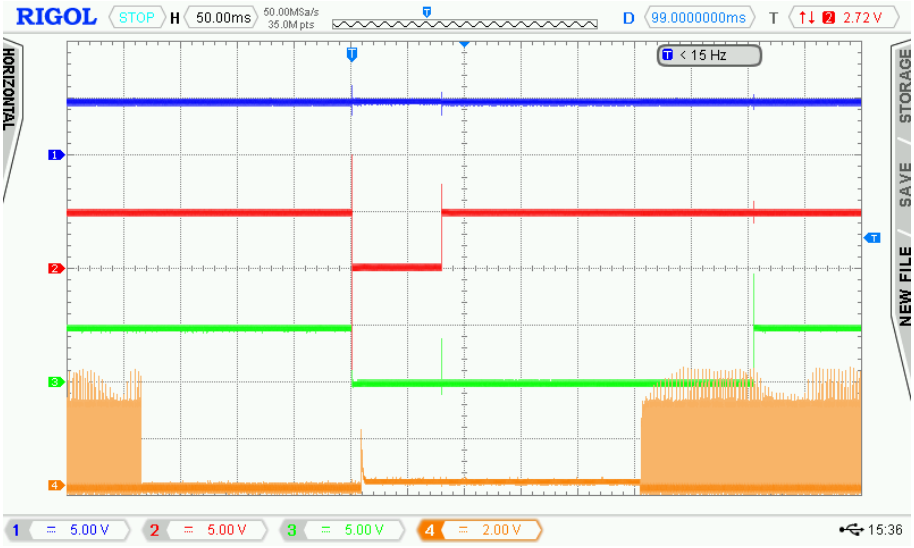


**Figure 25: Loss of watchdog timer trigger – fault on FPGA clock**

The final test that was performed was to determine whether the trigger points of the current limiters are stable across a wide temperature range. This test was performed within a thermal-vacuum chamber on each current limiter individually. All the current limiter trip currents were set to approximately 2.5A. We can see from the results in Figure 26 that the trigger current remained fairly constant, a total variation of 3.0%, 6.5% and 8.5% for the Power Distribution Current Limiter, Autonomous Current Limiter and Controlled Current Limiter, respectively, across a temperature range of -25°C to 85°C. The specific variation can be attributed to the characteristics of the components used – for example, the variations of the resistance with regards to temperature, as well the variations of the triggering point (of the internal comparator) of the FET drivers.

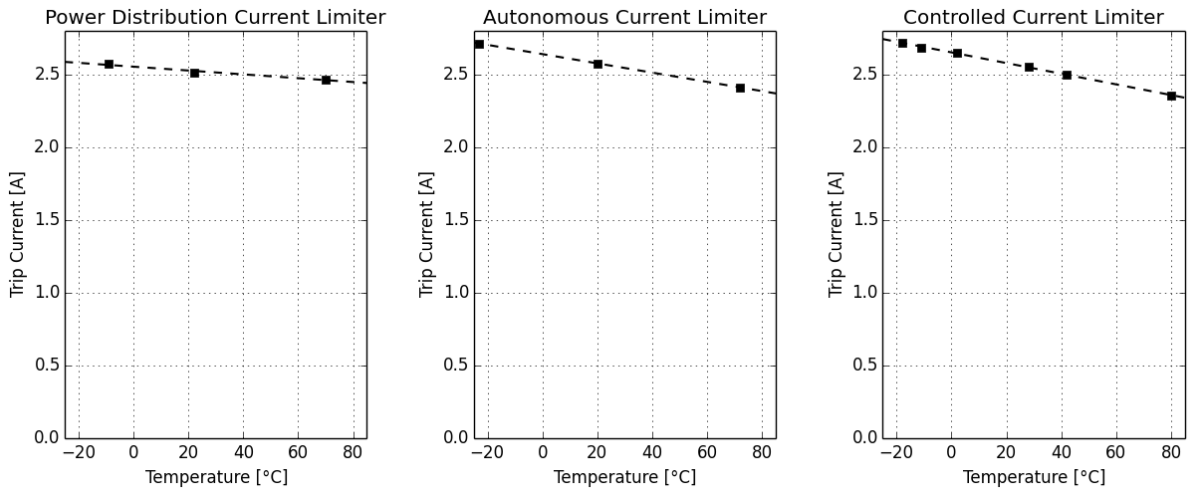


**Figure 26: Trip currents of the different current limiters with regards to temperature.**

# 5. Conclusion

The FDIR techniques presented in this paper, when combined, present a flexible and reliable starting point for designing nanosatellite missions for harsher environments. After evaluating the radiation environments encountered on such missions, specific failure modes were identified, focusing primarily on SEE effect. In order to ease the integration into the development process, the FDIR process was split into High and Low Level FDIR policies. For the Low Level FDIR policy, specific measures were identified, which rely on the use of an FPGA to protect satellite electronics. Specifically, the use of current limiters is proposed for managing power distribution as well as protecting components from destructive events, combined with the presented use of Watchdog timers and fault-tolerant logic design techniques.

The proposed approach was analyzed for tolerance to various SEE effects with regards to the specific parts on the satellite where they might occur. It was determined that the approach allows recovery from most faults. Further, an implementation of the presented current limiters and watchdog timers was implemented and tested for resource use and proper functionality. It was found that the logic resource overhead is minimal, while the current limiters and watchdog timers perform as expected, even within the extended temperature range. The current limiters were also implemented as miniaturized 3D circuits, which allows for more compact nanosatellite subsystem design and also facilitates design reuse.

Finally, it must also be noted that nanosatellites do not necessary need to be 100% reliable. Most of the advanced nanosatellite applications hinge on the fact that multiple nanosatellites in a swarm are used in conjunction in order to achieve a common scientific (or other) goal. As such, the failure of any single nanosatellite does not necessary (depending on the specific constellation) pose a problem to the mission. By using the presented approach the cost of a single nanosatellite can be reduced. As such, utilizing redundant satellites in the constellation can present an economically viable alternative to designing a higher reliability satellite. Though there are cases where even the failure of a single satellite is unacceptable (e. g. orbiting in GEO due to potential space debris issues, failure during deployment due to potential damage to other satellites), the authors believe that the FDIR concept

presented in this paper is robust enough to justify the use of nanosatellites even in single-satellite harsher-than-LEO applications.

# 6. References

[1] E. Buchen, D. DePasquale, 2014 Nano/Microsatellite Market Assessment, SpaceWorks Enterp. IncSEI Atlanta Tech Rep. (2014) 1–18.
[2] H. Heidt, J. Puig-Suari, A. Moore, S. Nakasuka, R. Twiggs, CubeSat: A new generation of picosatellite for education and industry low-cost space experimentation, (2000).
[3] M. Swartwout, The first one hundred cubesats: A statistical look, J. Small Satell. 2 (2014) 213–233.
[4] R. Rose, J. Dickinson, A. Ridley, CubeSats to NanoSats; Bridging the gap between educational tools and science workhorses, in: Aerosp. Conf. 2012 IEEE, IEEE, 2012: pp. 1–11.
[5] J.E. Mazur, W.R. Crain, M.D. Looper, D.J. Mabry, J.B. Blake, A.W. Case, M.J. Golightly, J.C. Kasper, H.E. Spence, New measurements of total ionizing dose in the lunar environment, Space Weather. 9 (2011).
[6] Y. Shin, S. Yoon, Y. Seo, H. Jin, J. Seon, Radiation effect for a CubeSat in slow transition from the Earth to the Moon, Adv. Space Res. 55 (2015) 1792–1798.
[7] D. Sinclair, J. Dyer, Radiation Effects and COTS Parts in SmallSats, in: Proc. AIAAUSU Conf. Small Satell., 2013.
[8] Y.M. Seo, Y.H. Kim, S.H. Park, J. Seon, Cumulative ionizing effect from solar-terrestrial charged particles and cosmic rays for CubeSats as simulated with GEANT4, Curr. Appl. Phys. 12 (2012) 1541–1547.
[9] J. Likar, S. Stone, R. Lombardi, K. Long, Novel Radiation Design Approach for CubeSat Based Missions, in: Proc. AIAAUSU Conf. Small Satell., 2010.
[10] Mathias Rousselet, Jérôme Boch, Laurent Dusseau, Frédéric Saigne, Muriel Bernard, Axel Rodriguez, Jean-Roch Vaille, Pierre-François Peyrard, Radiation Hardness Assurance for nanosatellites, in: Proc. 4S Symp. 2014, Blau Porto Petro Resort, Majorca, Spain, 2014.
[11] I. Jun, Effects of secondary particles on the total dose and the displacement damage in space proton environments, Nucl. Sci. IEEE Trans. On. 48 (2001) 162–175.
[12] K.A. LaBel, M. Gates, Single-Event-Effect from a System Perspective, IEEE Trans. Nucl. Sci. 43 (1996).
[13] M.A. Viscio, N. Viola, S. Corpino, F. Stesina, S. Fineschi, F. Fumenti, C. Circi, Interplanetary CubeSats system for space weather evaluations and technology demonstration, Acta Astronaut. 104 (2014) 516–525.
[14] R. Staehle, D. Blaney, H. Hemmati, M. Lo, P. Mouroulis, P. Pingree, T. Wilson, J. Puig-Suari, A. Williams, B. Betts, Interplanetary CubeSats: opening the solar system to a broad community at lower cost, Final Rep. NIAC Phase. 1 (2011).
[15] O. Durou, V. Godet, L. Mangane, D. Pérarnaud, R. Roques, Hierarchical fault detection, isolation and recovery applied to COF and ATV avionics, Acta Astronaut. 50 (2002) 547–556.
[16] F. Bruhn, J. Schulte, P. Selin, J. Freyer, NJORD: A Plug-and-Play based Fault Tolerant CubeSat Architecture, in: ESACNES Small Satell. Serv. Syst. 4S Symp. 4-8 June 2012 Portoroz Slov. CubeSat Workshop V 7, 2012.
[17] W. Haebel, A new approach to provide high-reliability data systems without using space-qualified electronic components, Acta Astronaut. 55 (2004) 563–571.
[18] G. Innocenti, J.F. Arrigo, A Compact Power Controller for Microsat Applications, in: Aerosp. Conf. 2008 IEEE, IEEE, 2008: pp. 1–10.
[19] H. Ashida, K. Fujihashi, S. Inagawa, Y. Miura, K. Omagari, N. Miyashita, S. Matunaga, T. Toizumi, J. Kataoka, N. Kawai, Design of Tokyo Tech nano-satellite Cute-1.7+ APD II and its operation, Acta Astronaut. 66 (2010) 1412–1424.

[20] M. Pajusalu, E. Ilbis, T. Ilves, M. Veske, J. Kalde, H. Lillmaa, R. Rantsus, M. Pelakauskas, A. Leitu, K. Voormansik, Design and pre-flight testing of the electrical power system for the ESTCube-1 nanosatellite, in: Proc Est. Acad Sci, 2014: pp. 232–241.
[21] R. Mahmood, K. Khurshid, Q.U. Islam, Institute of Space Technology CubeSat: ICUBE-1 subsystem analysis and design, in: Aerosp. Conf. 2011 IEEE, IEEE, 2011: pp. 1–11.
[22] J. Bouwmeester, G.F. Brouwer, E.K.A. Gill, G.L.E. Monna, J. Rotteveel, Design status of the Delfi-Next nanosatellite project, in: 61st Int. Astronaut. Congr. Prague Czech Repub. 27 Sept.-1 Oct. 2010, International Astronautical Federation, 2010.
[23] D. Evans, M. Merri, OPS-SAT: An ESA Nanosatellite for Accelerating Innovation in Satellite Control, in: SpaceOps Conf., 2014.
[24] X. Yu, J. Zhou, CubeSat: A candidate for the asteroid exploration in the future, in: Manip. Manuf. Meas. Nanoscale 3M-NANO 2014 Int. Conf. On, IEEE, 2014: pp. 261–265.
[25] J. Bouwmeester, N. Santos, Analysis of the Distribution of Electrical Power in Cubesats, in: Proc. 4S Symp. 2014, 2014.
[26] Lars Alminde, Morten Bisgaard, Fjolnir Gudmundsson, Claus Kejser, Toke Koustrup, Christian Lodberg, Tor Viscor, Gert K. Andersen, Power Supply Unit for the AAU-Cubesat, 2001.
[27] R.W. Kingsbury, F.H. Schmidt, K. Cahoy, D.A. Sklair, W.J. Blackwell, I. Osarentin, R.S. Legge Jr, R.S. Legge Jr, TID tolerance of popular cubesat components, (2013).
[28] F.W. Sexton, Destructive single-event effects in semiconductor devices and ICs, IEEE Trans. Nucl. Sci. 50 (2003) 603–621.
[29] L.S. Parobek, Research, Development and Testing of a Fault-Tolerant FPGA-Based Sequencer for CubeSat Launching Applications, DTIC Document, 2013.
[30] C. Carmichael, Triple module redundancy design techniques for Virtex FPGAs, Xilinx Appl. Note XAPP197. 1 (2001).
[31] K.S. Morgan, D.L. McMurtrey, B.H. Pratt, M.J. Wirthlin, A Comparison of TMR With Alternative Fault-Tolerant Design Techniques for FPGAs, IEEE Trans. Nucl. Sci. 54 (2007) 2065–2072. doi:10.1109/TNS.2007.910871.
[32] W. Xu, J. Wang, Y. Hu, J.-Y. Lee, F. Gong, L. He, M. Sarrafzadeh, In-place FPGA retiming for mitigation of variational single-event transient faults, Circuits Syst. Regul. Pap. IEEE Trans. On. 58 (2011) 1372–1381.
[33] Y. Bentoutou, M. Djaifri, A.M. Si-Mohammed, Design implementation of a quasi-cyclic codec for random access memories on-board Alsat-1, Acta Astronaut. 66 (2010) 954–961.
[34] B. Taylor, C. Underwood, A. Dyer, C. Ashton, S. Rason, J. Browning, The micro radiation environment monitor (MuREM) and SSTL radiation monitor (SSTL RM) on TechDemoSat-1, Nucl. Sci. IEEE Trans. On. 59 (2012) 1060–1065.
[35] T. Vladimirova, X. Wu, A Reconfigurable System-on-Chip Architecture for Pico-Satellite Missions., in: WoTUG-30 Proc. 30th WoTUG Tech. Meet., 2007: pp. 493–502.
[36] C. Ziemke, T. Kuwahara, I. Kossev, An integrated development framework for rapid development of platform-independent and reusable satellite on-board software, Acta Astronaut. 69 (2011) 583–594.