






Article

Reliability Analysis of the SHyLoC CCSDS123 IP Core for Lossless Hyperspectral Image Compression Using COTS FPGAs

Luis Alberto Aranda ^{1,*}, Antonio Sánchez ², Francisco Garcia-Herrero ¹, Yubal Barrios ², Roberto Sarmiento ² and Juan Antonio Maestro ¹

¹ ARIES Research Center, Universidad Antonio de Nebrija, Pirineos 55, 28040 Madrid, Spain; fgarciahe@nebrija.es (F.G.-H.); jmaestro@nebrija.es (J.A.M.)

² Institute for Applied Microelectronics, University of Las Palmas de Gran Canaria, 35001 Las Palmas de Gran Canaria, Spain; ajsanchez@iuma.ulpgc.es (A.S.); ybarrios@iuma.ulpgc.es (Y.B.); roberto@iuma.ulpgc.es (R.S.)

* Correspondence: laranda@nebrija.es

Received: 2 September 2020; Accepted: 5 October 2020; Published: 14 October 2020



Abstract: Hyperspectral images can comprise hundreds of spectral bands, which means that they can represent a large volume of data difficult to manage with the available on-board resources. Lossless compression solutions are interesting for reducing the amount of information stored or transmitted while preserving it at the same time. The Hyperspectral Lossless Compressor for space applications (SHyLoC), which is part of the European Space Agency (ESA) IP core's library, has been demonstrated to meet the requirements of space missions in terms of compression efficiency, low complexity and high throughput. Currently, there is a trend to use Commercial Off-The-Shelf (COTS) on-board electronic devices on small satellites. Moreover, commercial Field-Programmable Gate Arrays (FPGAs) have been used in a number of them. Hence, a reliability analysis is required to ensure the robustness of the applications to Single Event Upsets (SEUs) in the configuration memory. In this work, we present a reliability analysis of this hyperspectral image compression module as a first step towards the development of ad-hoc fault-tolerant protection techniques for the SHyLoC IP core. The reliability analysis is performed using a fault-injection-based experimental set-up in which a hardware implementation of the Consultative Committee for Space Data Systems (CCSDS) 123.0-B-1 lossless compression standard is tested against configuration memory errors in a Xilinx Zynq XC7Z020 System-on-Chip. The results obtained for unhardened and redundancy-based protected versions of the module are put into perspective in terms of area/power consumption and availability/protection coverage gained to provide insight into the development of more efficient knowledge-based protection schemes.

Keywords: fault injection; FPGA; hyperspectral image compression; reliability; soft errors

1. Introduction

On-board hyperspectral image compression has been adopted by several space missions since it implies a significant data volume reduction [1]. A hyperspectral image is obtained by measuring the spectrum of each pixel in the scene. Therefore, hyperspectral images can be comprised of hundreds or even thousands of spectral bands, representing a large volume of data. In small satellite applications, in which the on-board memory to store this amount of information is limited, hyperspectral image compression is at the same time challenging and a necessity.

A hyperspectral image compressor for space applications must meet mission requirements in terms of compression efficiency, low complexity and high throughput. But the power consumption and reliability aspects of the image compressor have to be addressed also, particularly in small satellites [2]. Reliability in space applications is a major concern since the malfunction of some of the on-board devices may lead to the premature end of the mission. As an example, some of the detectors of the Aqua imaging spectroradiometer sensor stopped working, and the information retrieved by the sensor contains large areas of dead pixel stripes [3]. In small satellites, reliability is related to the COTS devices that are commonly used to reduce the overall cost of the spacecraft. COTS components such as FPGAs based on Static Random Access Memory (SRAM) are attracting attention because of its reconfigurable capabilities and low cost compared to Application Specific Integrated Circuits (ASICs) [4]. However, SRAM-based FPGAs are susceptible to soft errors since the configuration memory of these devices is made from SRAM cells [5]. Therefore, a soft error may cause bit flips in this memory, modifying the structure of the design, and thus its behavior until reconfiguration is performed. For example, a modification in the behavior of the on-board hyperspectral image compression module may lead to the corruption of several samples during the decompression process (see Figure 1). For this reason, the design implemented in the FPGA has to be protected against soft errors to avoid malfunctions.

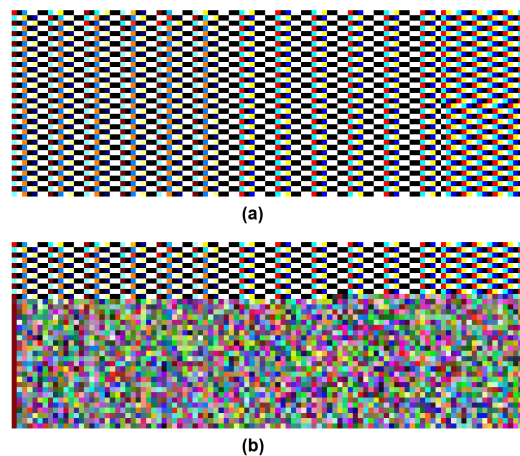


Figure 1. Decompressed hyperspectral image (a) without errors (b) corrupted by a malfunction error during the compression process.

There are plenty of alternatives in the literature to protect FPGA designs against configuration memory errors, but those based on modular redundancy are usually implemented due to their conceptual simplicity and high protection coverage [6]. However, these techniques have a major drawback related to the large area overhead and, therefore, large power consumption. As it was aforementioned, this aspect must be carefully analyzed to meet mission requirements in small satellites.

In this paper, we present a reliability analysis of a hardware implementation of the CCSDS 123.0-B-1 lossless compression standard [7] motivated by the results obtained from the radiation experiments presented in a previous work [8]. In this previous work, the robustness of the CCSDS 121.0-B-2 standard was evaluated, drawing some lessons learned. The first and most immediate conclusion was that soft errors lead to the malfunction of the compression module implemented in the SRAM-based FPGA and, eventually, to the corruption of the processed image (as illustrated in Figure 1). But it was also identified that experiments performed in radiation facilities have some disadvantages such as complexity and, the most important, difficulty to replicate the experiments. In this paper, a fault-injection-based approach is followed in order to have more control over the experiments. In this manner, the most sensitive regions or modules to protect can be identified in an easier way.

The hardware implementation of the CCSDS 123.0-B-1 is part of the SHyLoC, a configurable system that supports different multispectral and hyperspectral data arrangements and dimensions [9]. The CCSDS-123 Intellectual Property (IP) core has been implemented on the SRAM-based FPGA part of a Xilinx Zynq XC7Z020 System-on-Chip and tested against configuration memory errors through the mentioned fault-injection-based experimental set-up. From these experiments, the intrinsic reliability of the unprotected design is estimated to get an insight into its behavior in a real deployed satellite. Moreover, two designs protected with a modular redundancy approach have also been tested to put into perspective the protection achieved by the traditional techniques and the penalties added in terms of area overhead, power consumption and availability. As a result of this information, future ad-hoc protection techniques based on the system knowledge can be developed.

The rest of the paper is structured as follows—Section 2 presents the related work. Section 3 provides an overview of the CCSDS-123 standard algorithm and its hardware implementation. In Section 4 the experimental set-up used for the reliability analysis is explained in detail, while the results of the fault injection campaign are shown and discussed in Section 5. Finally, Section 6 concludes the paper.

2. Related Work

Over the years, great progress has been made in remote analysis of surfaces through hyperspectral sensors. In particular, the presence of hyperspectral sensors in space applications for detection and identification tasks (e.g., Earth Observation missions) reduces the number of sensors to be deployed on the land, saving time, bandwidth, material and human resources, and providing higher flexibility [10].

The main drawback of hyperspectral imaging is that these distance measurements generate a high volume of information that come from the light captured by the sensor in hundreds of different wavelengths. Besides, as technology improves, the amount of information from each wavelength and the total number of wavelengths (also named as bands) are increasing together with the improvement of the sensor resolution, requiring more bandwidth, more memory and more power consumption for processing this data.

To manage efficiently all this spectral information, data compression methods are necessary. Recently, researchers have introduced different architectures to implement lossless compression methods [11–13]. The focus of these works is on providing different solutions with both high compression efficiency and low-power/low-area implementations. High compression efficiency is required as the bandwidth of the downlink is limited, while low-power/low-area is essential in space applications since the number of on-board hardware resources is tightly constrained. An alternative solution is the use of lossy [14–17] or perceptually lossless compression methods [18], which achieve higher compression ratios than lossless techniques, at the expense of a penalty in terms of resources utilisation, not fitting well with the current on-board electronics. This is because these techniques are based on transform approaches, even reusing some concepts from video compression solutions, which are more complex (in terms of computational performance) than prediction-based lossless compression methods [19].

In Reference [8] two IP cores for hyperspectral and multispectral image compression were described in VHDL, which have been included in the portfolio of the European Space Agency (ESA)'s IP cores for space missions. These technology-independent cores implement compression solutions following the CCSDS 121.0-B-2 [20] and the CCSDS 123.0-B-1 [7] standards. Afterwards, both IPs were extended in Reference [9] in order to include standard-compliant functionality not implemented in the first version of these designs, improving at the same time results in terms of throughput. In Reference [9], an in-depth comparison of the SHyLoC CCSDS-123 IP core with other FPGA implementations of the CCSDS 123.0-B-1 compression algorithm, targeting both Radiation Hardened By Design (RHBD) [11,21,22] and COTS [23,24] devices, is addressed in terms of timing capabilities (i.e., maximum frequency and throughput) and resources utilization. In this way, a general overview about the IP features and its viability to be implemented on-board satellites is provided.

Although these IP cores are very efficient in terms of power consumption, meet real-time requirements imposed by the sensors and can achieve high throughput compared to the alternatives available in the literature [25], the necessity of applying mitigation techniques to increase the robustness of the compressor was stated. Otherwise, the compressors would not work in critical environments exposed to radiation [26], if COTS devices were used. To solve this problem, low-cost solutions based on implementing RHBD strategies can be applied. Different RHBD techniques have been developed over the years to protect FPGA designs against radiation providing good results [27–29].

This low-cost solution was also adopted in Reference [26] implementing a Triple Modular Redundancy (TMR) scheme to the SHyLoC-121 IP, which showed good results in terms of hardening under heavy ion radiation. However, it was concluded that a better trade-off between protection and area/power consumption needs to be achieved so that it is important not to lose the area/power efficiency of the unhardened implementation when we try to protect it.

In this work, the trade-off between area/power consumption and protection against radiation effects is analyzed for different unprotected and protected hardware implementations of the CCSDS 123.0-B-1 standard. In the next section, an overview of the algorithm and the IP core architecture of the CCSDS 123.0-B-1 implementation is explained in detail.

3. Overview of the CCSDS123-IP for Hyperspectral Image Lossless Compression

3.1. Algorithm Description

The CCSDS 123.0-B-1 standard [7] defines a lossless compression solution for multispectral and hyperspectral images, focusing on reaching a high level of data compression while maintaining the image quality and without incurring a high area occupancy. This algorithm includes several parameters, that should be configured to tune the compression efficiency. It is comprised of two main stages: a predictor and an entropy coder.

3.1.1. Predictor

The predictor is used to reduce the correlation among input samples. It estimates the value of a new input sample by using a reduced set of neighboring pixels, both in the spatial and the spectral domains (bands). The predictor implements the algorithm detailed in the next lines.

First, for each input sample $s_{z,y,x}$ located in spatial coordinates (y, x) and band z , a local sum ($\sigma_{z,y,x}$) is calculated for the current band as well as for the previous bands. The vicinity used for computing the local sum depends on the selected type: neighbor-oriented or column-oriented, which takes into account all the previously processed adjacent samples or only the sample right above, respectively. The equation for calculating the neighbor-oriented local sum is shown in Equation (1) while the column-oriented type is expressed in Equation (2).

$$\sigma_{z,y,x} = s_{z,y-1,x-1} + s_{z,y,x-1} + s_{z,y-1,x} + s_{z,y-1,x+1} \quad (1)$$

$$\sigma_{z,y,x} = 4 \cdot s_{z,y-1,x}. \quad (2)$$

The next step is to calculate the local differences ($d_{z,y,x}$) by subtracting the value of the neighboring pixels from the computed local sums. Both the central and the directional differences are computed according to the equations defined by the standard [7] and summarized in Equations (3)–(6) respectively. Under the reduced mode, only the central difference with the previous bands is computed, while the full mode also considers the directional differences in the current band. These differences are arranged in the local differences vector ($U_{z,y,x}$), whose arrangement depends on the selected prediction mode, as shown in Figure 2.

$$d_{z,y,x} = 4s_{z,y,x} - \sigma_{z,y,x} \tag{3}$$

$$d_{z,y,x}^N = \begin{cases} 4s_{z,y-1,x} - \sigma_{z,y,x}, & y > 0 \\ 0, & x > 0, y = 0 \end{cases} \tag{4}$$

$$d_{z,y,x}^W = \begin{cases} 4s_{z,y,x-1} - \sigma_{z,y,x}, & x > 0, y > 0 \\ 4s_{z,y-1,x} - \sigma_{z,y,x}, & x = 0, y > 0 \\ 0, & x > 0, y = 0 \end{cases} \tag{5}$$

$$d_{z,y,x}^{NW} = \begin{cases} 4s_{z,y-1,x-1} - \sigma_{z,y,x}, & x > 0, y > 0 \\ 4s_{z,y-1,x} - \sigma_{z,y,x}, & x = 0, y > 0 \\ 0, & x > 0, y = 0 \end{cases} \tag{6}$$

$$U_{z,y,x}^T = \left[\begin{array}{c|ccc|ccc} & \text{Central local diffs.} & & & \text{Directional diffs.} & & & \\ \hline & d_{z-1,y,x} & d_{z-2,y,x} & \dots & d_{z-p,y,x} & d_{z,y,x}^N & d_{z,y,x}^W & d_{z,y,x}^{NW} \\ \hline & \text{Reduced prediction mode} & & & & & & \end{array} \right]$$

Figure 2. Construction of the local differences vector $U_{z,y,x}$.

Then, a weighted sum of the elements in the local differences vector is computed to calculate the predicted sample $\hat{s}_{z,y,x}$. This sum is done using a weight vector, which is independently maintained for each band. These weights are updated after a new sample is processed, taking into account the prediction residual, the local differences, and some user-defined parameters. Finally, the prediction residuals are mapped into unsigned integer values ($\delta_{z,y,x}$), which are the input of the entropy coder.

3.1.2. Entropy Coder

In the CCSDS 123.0-B-1 standard, two options are defined for encoding purposes: the block-adaptive and the sample-adaptive entropy coders [7]. The block-adaptive encoder is defined in the CCSDS 121.0-B-2 standard [20] and it is based on adaptive Rice coding [30], a set of the Golomb coding that employs a power of two as a tunable parameter, making its hardware implementation easier. This alternative processes the input samples in groups of pixels, applying all the possible compression options simultaneously, and selecting the one that produces the shortest codeword. An identifier is attached to each compressed block to know which of the compression options has been used. On the other side, the sample-adaptive encoder is a tweaked version of a Rice-based coder. In this case, the input samples are compressed individually, one by one. The codeword for each pixel depends on the value of some image statistics (a counter and an accumulator) independent for each band, which are updated after every new processed sample.

As it is also reflected in Reference [31], the sample-adaptive encoder reaches, in general terms, higher compression ratios than the block-adaptive one, being at the same time simpler in terms of hardware complexity.

Finally, as a summary, a flowchart of the CCSDS 123 algorithm is illustrated in Figure 3.

From the explanation of the algorithm, it can be inferred that the amount of hardware required to implement the predictor will be larger than the hardware required for the entropy coder. This is because the former has to perform more arithmetic operations than the latter. As it will be shown in Section 5, this will lead to the occurrence of more critical bits in the predictor module of the IP core.

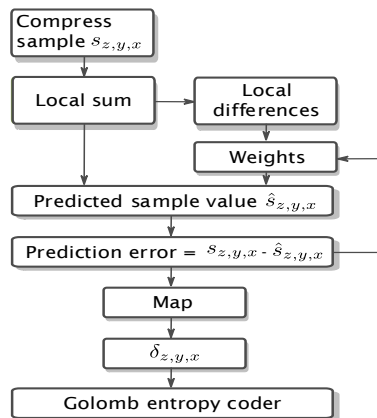


Figure 3. Flowchart of the CCSDS 123 algorithm.

3.2. IP Core Architecture

As it was aforementioned, the CCSDS-123 IP core under analysis in this work is part of SHyLoC [9], a couple of IP cores written in technology-independent VHDL language, and compliant with the CCSDS 123.0-B-1 and CCSDS 121.0-B-2 lossless compression standards. These IPs are currently available at the ESA IP Cores library [32], offering a technology-independent hardware solution for compressing generic data and hyperspectral images on-board satellites. Thanks to these two IPs, different configurations and solutions are possible (see Figure 4). Both IPs can be used together for compressing multispectral and hyperspectral images (Figure 4b), but it is also possible to use the CCSDS-121 IP in standalone mode to compress generic data (Figure 4c). The selected architecture in this study is the one reflected in Figure 4a, where the CCSDS-123 IP is used in standalone mode, because it can reach high throughput while using a reduced amount of FPGA resources. From the reliability point of view, the characterization and protection of the CCSDS-123 IP core against soft errors is relevant since it is the input module in Figure 4b implementation, and a malfunction in this IP would lead to the malfunction of the whole system.

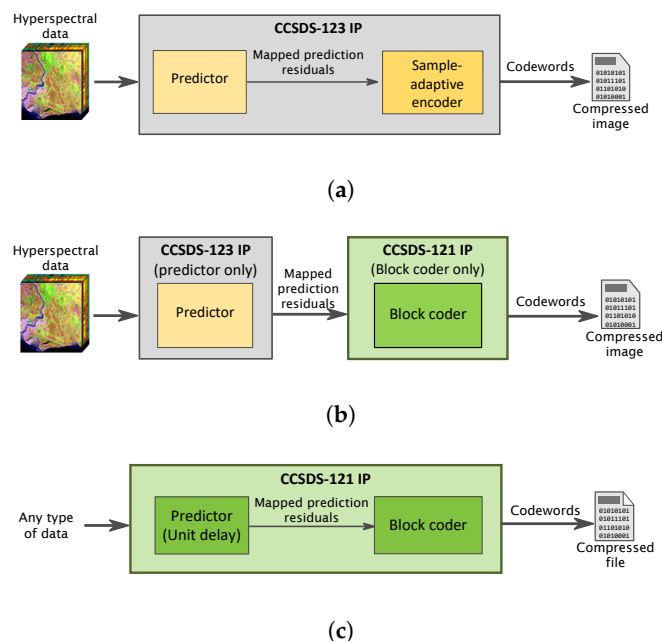


Figure 4. General overview of SHyLoC 2.0. (a) CCSDS-123 IP standalone; (b) CCSDS-123 predictor with CCSDS-121 entropy coder; (c) CCSDS-121 IP standalone.

The SHyLoC CCSDS-123 IP includes both the predictor and the sample-adaptive entropy coder described in the previous subsections. It can be drawn from the description of these algorithms that the predictor can be a critical module due to its arithmetic complexity. In Reference [33], a high amount of constants and runtime configuration parameters are presented for this module, including the local sum and prediction modes or the initial value of the weights vector. However, an interesting option for this reliability study is the possibility of implementing Error Detection And Correction (EDAC) in the dedicated memory, asserting an error signal in case this mechanism reports an issue. In this way, the reliability of this module could be increased. As will be discussed in Section 5, the implementation of the EDAC will imply a small percentage of error detections in the predictor module.

SHyLoC was designed focusing on versatility, which is the reason why all data arrangements in multispectral and hyperspectral applications are supported—Band-Interleaved By Pixel (BIP), Band-Interleaved By Line (BIL) and Band Sequential (BSQ). In this reliability study, the BIP architecture without access to external memory is considered since it is the only option that achieves a throughput of one sample per clock cycle.

In addition to the predictor and sample-adaptive encoder blocks, the CCSDS-123 IP includes other modules for control and configuration purposes. The configuration module receives the user-defined configuration parameters, validates and disseminates them to the rest of the compression engine, while the dispatcher module controls the output data stream. For clarification purposes, a high-level block diagram of the CCSDS-123 IP core is illustrated in Figure 5.

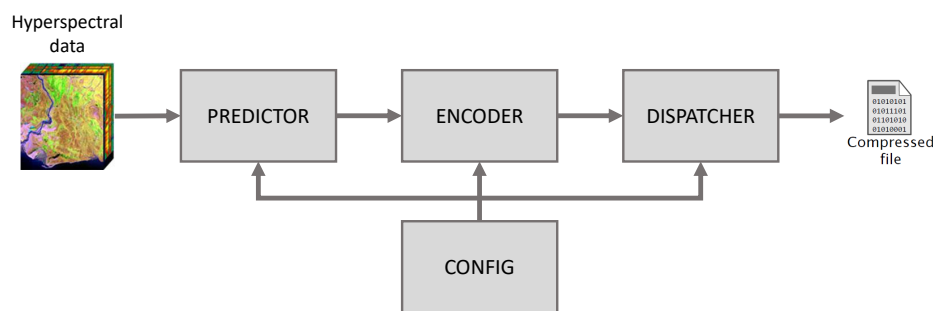


Figure 5. Block diagram of the CCSDS-123 IP core architecture

4. Experimental Set-Up

To perform the reliability analysis of the SHyLoC CCSDS-123 IP core described previously, the experimental set-up shown in Figure 6 has been used. As a result of this set-up, the intrinsic reliability of the different modules of the IP core can be tested. In addition, the protection coverage of traditional protection schemes can also be evaluated and compared to provide insight into the development of more efficient ad-hoc protection techniques for the hyperspectral compression system.

This set-up consists of an FPGA, two Digilent USB-UART peripheral modules (PMODs), and a MATLAB script running in a computer to control the experiments. In particular, the Designs Under Test (DUT) have been implemented on the XC7Z020-CLG484 FPGA part, a COTS device integrated in the Xilinx Zynq-7000 SoC family, together with the Xilinx Soft Error Mitigation (SEM) IP Controller. The SEM IP is a hardware module that can be used to inject and correct errors in the configuration memory of an SRAM-based FPGA through its Internal Configuration Access Port (ICAP) by reading and writing the content of the configuration memory. As mentioned at the beginning of the paragraph, two Digilent USB-UART PMODs are required with this set-up. One PMOD is used to control the status of the SEM IP to inject and later correct the injected bit flip, and the second PMOD is used to receive in the computer the output of the DUT. This helps us to determine whether the injected bit flip has produced a critical error in the design or not.

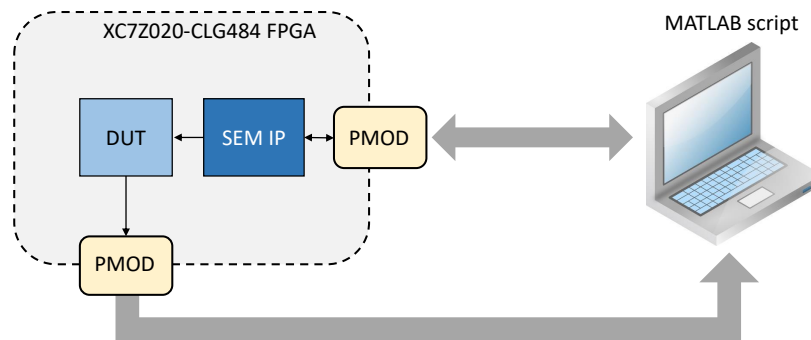


Figure 6. Experimental set-up used to test the reliability of the IP core.

In addition to the PMODs and the MATLAB communication script, the addresses where the bit flips will be injected by the SEM IP have to be obtained. To do so, the Automatic Configuration Memory Error-injection (ACME) tool was used [34]. ACME is an open-source tool that translates the configuration memory bits of an SRAM-based FPGA region into injection addresses for the SEM IP. It uses the Xilinx EBD file created by Vivado together with the coordinates of the FPGA region where the DUT is placed to generate a list with the injection addresses. In our case, the coordinates of the different modules of the CCSDS-123 IP core are given to the ACME tool, together with the EBD file of the whole design. Then, the tool uses this information to locate the configuration bits of the different modules and generates a list with the injection addresses that can be directly used by the SEM IP to perform bit flips. To avoid injection side-effects, Vivado placement constraints were used to exclude the SEM IP resources from the DUT area.

Once the experimental set-up shown in Figure 6 is implemented and the injection addresses are obtained with the ACME tool, the next step is to execute the MATLAB script to perform the fault injection campaign, obtaining the reliability results presented in Section 5. The steps of the MATLAB script are summarized below:

1. A golden execution where the DUT compresses the hyperspectral image in the absence of errors is first performed to obtain the correct outcome of the design. This outcome is stored for later comparisons when the bit flip is injected.
2. An injection address read from the list is obtained with the ACME tool, and a command is sent to the SEM IP to inject a single bit flip in that specific location.
3. The DUT compresses the hyperspectral image and the result is compared to the golden result obtained in step 1 to classify the bit as critical or non-critical. If the DUT has some kind of protection (e.g., the redundancy-based schemes presented in the next section), the status of the protection technique is also registered to determine if the error has been detected or not.
4. The results from the previous step are logged in a text file.
5. A command is sent to the SEM IP to correct the injected bit flip.
6. Steps 2–5 are repeated until all the desired configuration memory bit flips are injected.

It should be mentioned that the previously explained steps can be used to perform exhaustive fault injection campaigns by sequentially injecting in all the bits obtained by the ACME tool, or statistical fault injections by injecting in a random subset of these bits. In this paper, both approaches have been followed. Exhaustive fault injection campaigns have been done to characterize the unprotected SHyLoC CCSDS-123 IP core and analyze the sensitivity of each of the modules shown in Figure 5. On the other hand, statistical campaigns have been followed to test the reliability of the IP core protected with two traditional redundancy-based techniques to reduce the fault injection campaign runtime. In particular, 27,000 random bit flips have been tested in each campaign. This value implies a

confidence interval of 95% with an error margin of approximately 0.6% according to Cochran's sample size formula [35].

Finally, regarding the input hyperspectral images selected to check the behavior of the compression core, the IP core has been configured first to compress a synthetic image of $8 \times 7 \times 17$, and then to compress a crop of $20 \times 20 \times 224$ of an AVIRIS image from flight f080927t01p00r10. Hence, two different implementations of the IP core has been tested in the experiments presented below.

5. Experimental Results

In this section, the reliability results obtained by applying the experimental set-up explained before are presented and discussed together with area overhead and power consumption results. In particular, an unhardened version of the SHyLoC CCSDS-123 IP core, a second version of the IP core protected with a Dual Modular Redundancy (DMR) scheme, and a third version protected with a Triple Modular Redundancy (TMR) scheme have been chosen. These three designs have been chosen to obtain reliability metrics from different points of view. On the one hand, the unprotected design has been tested to measure the intrinsic reliability of the IP core, as well as to identify the most vulnerable modules of the design. On the other hand, two redundancy-based approaches have been selected to put into perspective the effectiveness of traditional protection schemes. In this manner, an upper limit will be defined in order to compare future ad-hoc protection techniques developed. These three designs have been configured to process the synthetic image and also the crop of the AVIRIS image mentioned at the end of the previous section. Therefore, six different implementations have been tested in this work. The results are discussed in Section 5.4.

5.1. Unhardened Design: Intrinsic Characterization

The first set of experiments has been carried out with the unprotected IP core architecture to measure its intrinsic reliability and to determine the most sensitive module to single bit flips in the configuration memory. As mentioned in Section 3.2, the SHyLoC CCSDS-123 IP consists of a predictor, a sample-adaptive entropy coder, and configuration and control modules. These four modules of the IP core (see Figure 5) have been studied by performing exhaustive fault injection campaigns in each module with the set-up shown in Figure 6. The intrinsic reliability results for the four modules configured to compress the synthetic $8 \times 7 \times 17$ image are summarized in Table 1, while the results for the crop of $20 \times 20 \times 224$ of the AVIRIS scene can be seen in Table 2. In these tables, the number of critical and non-critical bits is presented. It should be mentioned that a critical bit is a configuration memory bit that, when flipped, jeopardize the reliability of the design by modifying its outcome.

Table 1. Intrinsic reliability of the different modules of the CCSDS 123.0-B-1 IP core (synthetic image).

	Predictor	Entropy Coder	Config	Dispatcher
Critical bits	177,087 (20.54%)	29,753 (12.63%)	8385 (14.06%)	3919 (17.71%)
Non-critical bits	685,047 (79.46%)	205,731 (87.37%)	51,263 (85.94%)	18,211 (82.29%)
Design bits	862,134 (100%)	235,484 (100%)	59,648 (100%)	22,130 (100%)

Table 2. Intrinsic reliability of the different modules of the CCSDS 123.0-B-1 IP core (AVIRIS image).

	Predictor	Entropy Coder	Config	Dispatcher
Critical bits	186,652 (20.81%)	34,960 (13.98%)	8730 (12.26%)	3616 (16.57%)
Non-critical bits	710,173 (79.19%)	215,072 (86.02%)	62,497 (87.74%)	18,201 (83.43%)
Design bits	896,825 (100%)	250,032 (100%)	71,227 (100%)	21,817 (100%)

First, it can be observed that, as stated earlier in Section 3.1, the predictor is the module with the highest number of critical bits. This is reasonable since its arithmetic complexity is higher than the rest of the modules and uses most of the FPGA resources out of the whole design (see Tables 3

and 4). However, its intrinsic reliability (the percentage of critical bits out of the total number of bits) is worse than the rest of the modules. This means that the predictor is the most sensitive module of the SHyLoC CCSDS-123 IP core. This conclusion, together with the fact that the predictor is the module that uses most of the FPGA resources of the IP core, makes it the perfect candidate to develop an ad-hoc protection technique that uses fewer resources (and thus consumes less power) than traditional redundancy-based techniques and can keep the protection of the IP core at a reasonable level.

Table 3. FPGA resources used by each module of the CCSDS 123.0-B-1 IP core (synthetic image).

	Predictor	Entropy Coder	Config	Dispatcher
LUTs	4609	1262	300	155
FFs	3010	413	48	152
BRAMs	3.5	0.5	0	0
DSPs	6	0	0	0

Table 4. FPGA resources used by each module of the CCSDS 123.0-B-1 IP core (AVIRIS image).

	Predictor	Entropy Coder	Config	Dispatcher
LUTs	4606	1237	353	154
FFs	3010	419	53	152
BRAMs	10	0.5	0	0
DSPs	6	0	0	0

Finally, it should be mentioned that the predictor can detect a small percentage of critical errors due to the EDAC implemented in the dedicated memory (see Section 3.2). This feature can also be exploited in future works to improve the protection of the IP core. The number of detected errors is discussed later in Section 5.4.

5.2. Dual Modular Redundancy (DMR) Design: Error Detection

A DMR scheme, also known as Duplication With Comparison (DWC), is a widely used technique in which the outcome of two identical copies of the same module that are performing the same operations are used to detect malfunctions. The outcome of both modules is constantly checked to detect faults by a comparator. In DMR designs for space applications, the comparator is commonly duplicated since it is the most vulnerable part. Then, the comparator of the duplicated comparators is protected (e.g., manufacturing it in a rad-hard technology) to ensure the reliability of the design [36]. As additional protection, the output of one of the duplicated comparators can be inverted before feeding it to the final comparator. The described DMR scheme is the design that has been tested in this paper during the fault injection experiments. It is shown in Figure 7 for clarification purposes.

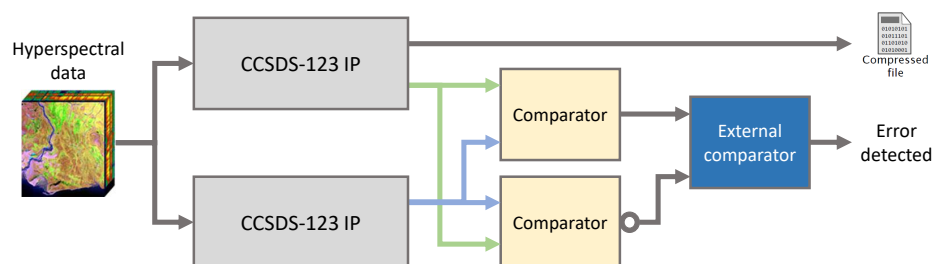


Figure 7. Dual Modular Redundancy (DMR) protection scheme.

It should be mentioned that the comparator of the duplicated comparators (identified as “External comparator” in Figure 7) has been placed outside the injection area to simulate a real case scenario in which the comparator of the duplicated comparators is immune to bit flips.

Finally, it should be remarked that a DMR scheme cannot correct errors by itself. In an SRAM-based FPGA, where configuration memory errors are persistent, an alternative is to use the “Error detected” signal to trigger a reconfiguration of the device to remove the error. The effectiveness of this approach and its impact on the availability of the image compression system is discussed and compared in Section 5.4.

5.3. Triple Modular Redundancy (TMR) Design: Error Masking

TMR is a well-known error correction technique based on triplicating the module to protect and comparing the three outputs through a majority voter. In this manner, a single error affecting one of the modules can be masked by the voter. As with the DMR scheme, the voter is the most vulnerable part of the design. For this reason, it is traditionally triplicated in space missions and other critical applications to enhance the reliability of the circuit. Then, the outputs of the voters are connected to a final voter, which is hardened against radiation. To simulate this, the majority voter identified in Figure 8 as “External majority voter” has been placed outside the injection region, which means that no configuration memory bit flips are performed in this voter during the fault injection campaigns.

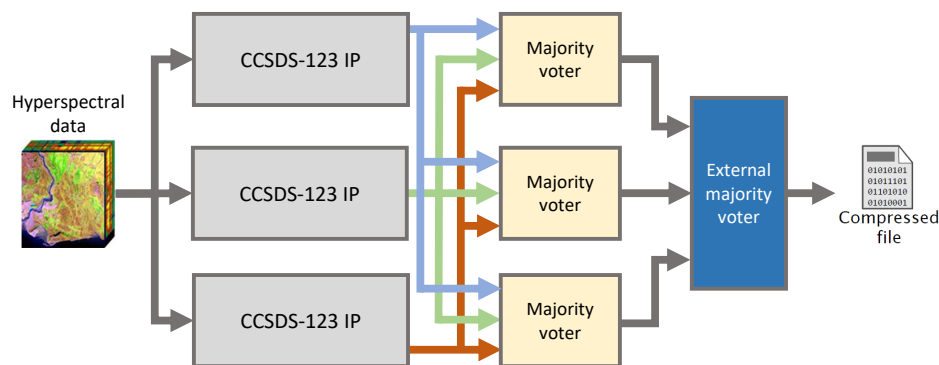


Figure 8. Triple Modular Redundancy (TMR) protection scheme.

Finally, it should be mentioned that a TMR implemented in an SRAM-based FPGA is almost immune to a single bit flip. In addition, unlike the DMR, no reconfiguration is performed to remove the error since it is masked but not detected by the voter. Therefore, the proper operation of the TMR technique has to be preserved by adding an error removal technique (e.g., scrubbing) to prevent the accumulation of errors.

5.4. Reliability Results and Discussion

In Table 5, the area overhead, the maximum frequency achieved and the power consumption of the implementations presented in previous subsections (unhardened CCSDS-123 IP core, DMR, and TMR schemes) are shown for each hyperspectral image test. Throughput results (measured in MSamples/s) are not included in Table 5, since it is equal to the maximum clock frequency. This is because the BIP architecture is able to process one sample per clock cycle, as it was mentioned in Section 3.2.

It can be observed in Table 5 that the IP core designs configured to compress the AVIRIS image of $20 \times 20 \times 224$ require more resources than their counterparts configured to compress the $8 \times 7 \times 17$ synthetic image. This is because the IP core requires more internal memory to process larger images (the size of some FIFOs directly depends of the number of bands N_z), which has a direct impact on the power consumption of the design (see Table 5).

Table 5. Area overhead and power consumption of the different unprotected and protected architectures.

	Synthetic Image			AVIRIS Image		
	Unprotected	DMR	TMR	Unprotected	DMR	TMR
LUTs	6409	12,827	19,489	6354	12,878	19,589
FFs	3645	7293	10,935	3662	7327	10,986
BRAMs	4	8	12	10.5	21	31.5
DSPs	6	12	18	6	12	18
Freq. (MHz)	64.3	63.9	63.1	63.6	63.0	62.8
Power (mW)	27	37	56	29	69	144

Regarding the reliability analysis, six fault injection campaigns (one for each design) were executed. In particular, 27,000 single bit flips were randomly injected in each campaign as explained in Section 4. The logged results have been classified as detected/undetected or masked/unmasked errors depending on the protection technique implemented. As explained before, detected/undetected errors are related to the DMR scheme while masked/unmasked errors are related to the TMR design. However, it is worth remembering that the unprotected design has an EDAC implemented in the dedicated memory, so a small percentage of errors were also detected. The results for each fault injection campaign are summarized in Tables 6 and 7.

Table 6. Reliability results for the different unprotected and protected architectures (synthetic image).

	Unprotected	DMR	TMR
Detected/masked	6875 (3.14%)	447,504 (99.71%)	717,552 (99.81%)
Undetected/unmasked	212,269 (96.86%)	1314 (0.29%)	1354 (0.19%)
Total critical bits	219,144 (100%)	448,818 (100%)	718,906 (100%)

Table 7. Reliability results for the different unprotected and protected architectures (AVIRIS image).

	Unprotected	DMR	TMR
Detected/masked	5016 (2.14%)	468,136 (99.69%)	729,698 (99.80%)
Undetected/unmasked	228,942 (97.86%)	1441 (0.31%)	1462 (0.20%)
Total critical bits	233,958 (100%)	469,577 (100%)	731,160 (100%)

First, it can be observed in the reliability tables that both DMR and TMR do not protect 100% of the design bits. This is because there are some configuration bits related to the input routing that can be affected by bit flips and cannot be avoided. The direct impact of these bit flips is that for example, in the DMR, both copies produce the same output (and therefore the comparators do not detect a mismatch) but the outputs are not correct since the input values have been modified by the routing error. A similar effect is produced in the TMR scheme due to the input routing. These undetected/unmasked errors are quite harmful since they are not detected by the protection techniques and can create silent data corruption. The presence of these errors in FPGAs is inevitable under the proposed mitigation techniques and will require an additional mechanism to remove them such as scrubbing-based or similar techniques. Finally, the percentages of detected/masked errors are more than 99.7% so it can be concluded that both DMR and TMR techniques are working properly in the presence of configuration memory errors.

A further observation is that the total number of critical bits differs from one design to another, with the highest value being in the TMR architecture configured to compress the AVIRIS image. This implies that the number of critical bits is related to the area overhead, as can be inferred from Table 5.

Regarding the area overhead shown in Table 5, the DMR scheme requires fewer resources than the TMR, which also implies lower power consumption. However, as mentioned in Section 5.2, DMR has lower availability because its “Error detected” signal triggers a reconfiguration of the device to restore its correct behavior. In a hyperspectral image compression system, the image is usually processed on-the-fly by temporarily storing parts of it. Therefore, a reconfiguration of the device during a compression procedure means that the image has to be discarded and captured again. In space missions, this implies complex and time-consuming maneuvers to capture a similar image, even waiting to complete a whole orbit to acquire the same scene again.

On the other hand, TMR has more area overhead and more power consumption so it may not be suitable for low-power applications but, since it is a technique based on masking the errors by means of a voter, it has more availability than DMR. However, an additional mechanism to remove the configuration memory errors is required to avoid the accumulation of errors that may eventually lead the TMR to failure. As mentioned before, there are undetected/unmasked errors in all the studied designs, so an additional error correction mechanism would be required.

For the reasons outlined before, it can be concluded that the TMR scheme is not the perfect alternative to protect the CCSDS-123 IP core design for small satellite applications. This is because power consumption is a tight restriction in these spacecraft and, as can be observed in Table 5, this design for the AVIRIS image requires almost $5\times$ more power than the unprotected design and $2\times$ more than the DMR scheme. In terms of error correction, both protected designs achieve a similar protection coverage but, since the TMR masks the bit flips, it suffers from accumulated errors that may eventually lead to a system malfunction. On the other hand, this phenomenon will be less common in the DMR scheme due to its reconfiguration behavior. In order to avoid error accumulation, the traditional TMR scheme would have to be modified by adding an error detection signal that triggers the correction mechanism (e.g., scrubbing). However, the area overhead and power consumption of the correction mechanism will also increase the overall values.

6. Conclusions

In this work, a reliability analysis of a hardware implementation of the standalone CCSDS-123 IP core has been presented. The intrinsic reliability of this hyperspectral image processing module has been tested by performing configuration memory fault injection campaigns. To determine the best alternative to protect it against these soft errors, two traditional redundancy-based techniques (i.e., DMR and TMR) have also been analyzed in terms of area overhead, power consumption, protection coverage and availability.

The main conclusion drawn from the experimental analysis performed in this paper is that the TMR scheme is not the perfect alternative to protect the CCSDS-123 IP core design in small satellite applications due to its high power consumption compared to the DMR scheme. In scenarios where the expected frequency of bit flips in the configuration memory is low, the availability of the TMR will be higher than the DMR due to the absence of a reconfiguration signal, allowing the image compression system to continue processing the image. The higher availability of the TMR means that the image being compressed does not have to be discarded and captured again. However, in more harmful scenarios, the occurrence of accumulated errors may be a problem for the TMR design, while the DMR scheme would be less affected by this phenomenon.

Since both protection techniques achieve a similar protection coverage, but the power consumption of the DMR scheme is half of the TMR, the DMR technique would be preferred to protect this module for small satellite applications. In order to reduce even more the power consumption while still keeping a reasonable level of fault tolerance, the protection of the most vulnerable module of the IP core (i.e., the predictor) could be considered by applying selective hardening or ad-hoc protection techniques. In a future work, the results from the intrinsic reliability analysis presented in this paper will be used to develop more advanced ad-hoc protection techniques to reduce power consumption.

Author Contributions: Conceptualization, L.A.A., A.S., F.G.-H., Y.B., R.S., and J.A.M.; methodology, L.A.A., A.S., F.G.-H., Y.B., R.S., J.A.M.; software, L.A.A., A.S., Y.B.; validation, L.A.A., F.G.-H., J.A.M.; formal analysis, L.A.A., A.S., F.G.-H., Y.B.; data curation, L.A.A., F.G.-H., J.A.M.; writing—original draft preparation, L.A.A., A.S., Y.B.; writing—review and editing, F.G.-H., R.S., J.A.M.; supervision, R.S., J.A.M.; project administration, L.A.A., R.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kiely, A.B.; Klimesh, M.; Blanes, I.; Ligo, J.; Magli, E.; Aranki, N.; Burl, M.; Camarero, R.; Cheng, M.; Dolinar, S.; et al. The new CCSDS standard for low-complexity lossless and near-lossless multispectral and hyperspectral image compression. In Proceedings of the ESA On-Board Payload Data Compression Workshop, Matera, Italy, 20–21 September 2018; pp. 1–6.
2. George, A.D.; Wilson, C.M. Onboard processing with hybrid and reconfigurable computing on small satellites. *Proc. IEEE* **2018**, *106*, 458–470. [[CrossRef](#)]
3. Shen, H.; Li, X.; Zhang, L.; Tao, D.; Zeng, C. Compressed Sensing-Based Inpainting of Aqua Moderate Resolution Imaging Spectroradiometer Band 6 Using Adaptive Spectrum-Weighted Sparse Bayesian Dictionary Learning. *IEEE Trans. Geosci. Remote Sens.* **2014**, *52*, 894–906. [[CrossRef](#)]
4. Siegle, F.; Vladimirova, T.; Iltstad, J.; Emam, O. Availability analysis for satellite data processing systems based on SRAM FPGAs. *IEEE Trans. Aerosp. Electron. Syst.* **2016**, *52*, 977–989. [[CrossRef](#)]
5. Asadi, G.; Tahoori, M.B. Soft Error Rate Estimation and Mitigation for SRAM-Based FPGAs. In *Proceedings of the 2005 ACM/SIGDA 13th International Symposium on Field-Programmable Gate Arrays*; Association for Computing Machinery: New York, NY, USA, 2005; pp. 149–160. [[CrossRef](#)]
6. Hoque, K.A.; Mohamed, O.A.; Savaria, Y. Dependability modeling and optimization of triple modular redundancy partitioning for SRAM-based FPGAs. *Reliab. Eng. Syst. Saf.* **2019**, *182*, 107–119. [[CrossRef](#)]
7. CCSDS. *Lossless Multispectral and Hyperspectral Image Compression, Recommended Standard CCSDS 123.0-B-1*; Blue Book; CCSDS: Washington, DC, USA, 2012.
8. Santos, L.; Gómez, A.; Sarmiento, R. Implementation of CCSDS standards for lossless multispectral and hyperspectral satellite image compression. *IEEE Trans. Aerosp. Electron. Syst.* **2019**, *56*, 1120–1138. [[CrossRef](#)]
9. Barrios, Y.; Sánchez, A.; Santos, L.; Sarmiento, R. SHyLoC 2.0: A versatile hardware solution for on-board data and hyperspectral image compression on future space missions. *IEEE Access* **2020**, *8*, 54269–54287. [[CrossRef](#)]
10. Landgrebe, D. Hyperspectral image data analysis. *IEEE Signal Process. Mag.* **2002**, *19*, 17–28. [[CrossRef](#)]
11. Tsigkanos, A.; Kranitis, N.; Theodorou, G.A.; Paschalis, A. A 3.3 Gbps CCSDS 123.0-B-1 multispectral & Hyperspectral image compression hardware accelerator on a space-grade SRAM FPGA. *IEEE Trans. Emerg. Top. Comput.* **2018**, in press.
12. Ferraz, O.; Silva, V.; Falcao, G. 1.5 GBIT/S 4.9 W Hyperspectral Image Encoders on a Low-Power Parallel Heterogeneous Processing Platform. In Proceedings of the ICASSP 2020–2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 1693–1697.
13. Nascimento, J.M.; Vestias, M.P.; Martin, G. Hyperspectral Compressive Sensing with a System-On-Chip FPGA. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, in press. [[CrossRef](#)]
14. Du, Q.; Fowler, J.E. Hyperspectral Image Compression Using JPEG2000 and Principal Component Analysis. *IEEE Geosci. Remote Sens. Lett.* **2007**, *4*, 201–205. [[CrossRef](#)]
15. Blanes, I.; Serra-Sagrasta, J. Cost and Scalability Improvements to the Karhunen–Loève Transform for Remote-Sensing Image Coding. *IEEE Trans. Geosci. Remote Sens.* **2010**, *48*, 2854–2863. [[CrossRef](#)]
16. Guerra, R.; Barrios, Y.; Díaz, M.; Santos, L.; López, S.; Sarmiento, R. A New Algorithm for the On-Board Compression of Hyperspectral Images. *Remote Sens.* **2018**, *10*, 428. [[CrossRef](#)]
17. Báscones, D.; González, C.; Mozos, D. Hyperspectral Image Compression Using Vector Quantization, PCA and JPEG2000. *Remote Sens.* **2018**, *10*, 907. [[CrossRef](#)]
18. Kwan, C.; Larkin, J. New Results in Perceptually Lossless Compression of Hyperspectral Images. *J. Signal Inf. Process.* **2019**, *10*, 96–124. [[CrossRef](#)]

19. Hussain, A.; Al-Fayadh, A.; Radi, N. Image compression techniques: A survey in lossless and lossy algorithms. *Neurocomputing* **2018**, *300*, 44–69. [[CrossRef](#)]
20. CCSDS. *Lossless Data Compression, Recommended Standard CCSDS 121.0-B-2*; Blue Book; CCSDS: Washington, DC, USA, 2012.
21. Santos, L.; Berrojo, L.; Moreno, J.; Lopez, J.F.; Sarmiento, R. Multispectral and Hyperspectral Lossless Compressor for Space Applications (HyLoC): A Low-Complexity FPGA Implementation of the CCSDS 123 Standard. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2016**, *9*, 757–770. [[CrossRef](#)]
22. Bascones, D.; Gonzalez, C.; Mozos, D. Parallel Implementation of the CCSDS 1.2.3 Standard for Hyperspectral Lossless Compression. *Remote Sens.* **2017**, *9*, 973. [[CrossRef](#)]
23. Fjeldtvedt, J.; Orlandic, M.; Johansen, T.A. An Efficient Real-Time FPGA Implementation of the CCSDS-123 Compression Standard for Hyperspectral Images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2018**, *11*, 3841–3852. [[CrossRef](#)]
24. Orlandic, M.; Fjeldtvedt, J.; Johansen, T.A. A Parallel FPGA Implementation of the CCSDS-123 Compression Algorithm. *Remote Sens.* **2019**, *11*, 673. [[CrossRef](#)]
25. Rodríguez, A.; Santos, L.; Sarmiento, R.; De La Torre, E. Scalable hardware-based on-board processing for run-time adaptive lossless hyperspectral compression. *IEEE Access* **2019**, *7*, 10644–10652. [[CrossRef](#)]
26. Sánchez, A.; Barrios, Y.; Santos, L.; Sarmiento, R. Evaluation of TMR effectiveness for soft error mitigation in SHyLoC compression IP core implemented on Zynq SoC under heavy ion radiation. In Proceedings of the 2019 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), Noordwijk, The Netherlands, 2–4 October 2019; pp. 1–4.
27. Aranda, L.A.; Reviriego, P.; Maestro, J.A. Towards a Fault-tolerant Star Tracker for Small Satellite Applications. *IEEE Trans. Aerosp. Electron. Syst.* **2020**, in press. [[CrossRef](#)]
28. Hogan, J.A.; Weber, R.J.; LaMeres, B.J. Reliability analysis of field-programmable gate-array-based space computer architectures. *J. Aerosp. Inf. Syst.* **2017**, *14*, 247–258. [[CrossRef](#)]
29. Banteywalu, S.; Khan, B.; De Smedt, V.; Leroux, P. A novel modular radiation hardening approach applied to a synchronous buck converter. *Electronics* **2019**, *8*, 513. [[CrossRef](#)]
30. Rice, R.F. *Some Practical Universal Noiseless Coding Techniques*; JPL Publication: Pasadena, CA, USA, 1979.
31. CCSDS. *Lossless Multispectral and Hyperspectral Image Compression, Informational Report CCSDS 120.2-G-1*; Green Book; CCSDS: Washington, DC, USA, 2015.
32. ESA. ESA HDL IP Cores Portfolio Overview. Available online: https://www.esa.int/Enabling_Support/Space_Engineering_Technology/Microelectronics/ESA_HDL_IP_Cores_Portfolio_Overview (accessed on 11 June 2019).
33. University of Las Palmas de Gran Canaria. SHyLoC Product Datasheet. 2017. Available online: https://amstel.estec.esa.int/tecedm/ipcores/SHyLoC_Datasheet_v1.0.pdf (accessed on 7 May 2020).
34. Aranda, L.A.; Sánchez-Macián, A.; Maestro, J.A. ACME: A tool to improve configuration memory fault injection in SRAM-based FPGAs. *IEEE Access* **2019**, *7*, 128153–128161. [[CrossRef](#)]
35. Cochran, W.G. *Sampling Techniques*, 3rd ed.; John Wiley & Sons: New York, NY, USA, 1977.
36. Kibar, O.O.; Mohan, P.; Rech, P.; Mai, K. Evaluating the Impact of Repetition, Redundancy, Scrubbing, and Partitioning on 28-nm FPGA Reliability Through Neutron Testing. *IEEE Trans. Nucl. Sci.* **2019**, *66*, 248–254. [[CrossRef](#)]

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).