

# LOSSLESS AND LOSSY HYPERSPECTRAL COMPRESSION IP CORE FOR SPACE MISSIONS

Antonio J. Sánchez<sup>1</sup>, Panagiotis Chatziantoniou<sup>2</sup>, Joan Bartrina-Rapesta<sup>3</sup>, Yubal Barrios<sup>1</sup>, Samuel Torres-Fau<sup>1</sup>, Luis Berrojo<sup>4</sup>, Lucana Santos<sup>5</sup>, and Roberto Sarmiento<sup>1</sup>

<sup>1</sup> *Institute for Applied Microelectronics (IUMA), University of Las Palmas de Gran Canaria (ULPGC)  
35017 Las Palmas de Gran Canaria, Spain*

<sup>2</sup> *Dept. of Informatics & Telecommunications, National and Kapodistrian University of Athens, Greece  
Athens, Greece*

<sup>3</sup> *Dept. of Information and Communications Engineering, Universitat Autònoma de Barcelona  
Campus UAB, 08193 Cerdanyola del Vallès, Spain*

<sup>4</sup> *Thales Alenia Space (TAS) in Spain. 28760 Tres Cantos, Madrid, Spain*

<sup>5</sup> *European Space Research and Technology Centre, European Space Agency. 2201 Noordwijk, The Netherlands*

## ABSTRACT

This work presents a new compression IP core for lossless and near-lossless compression of multi- and hyperspectral data, compliant with the CCSDS-123.0-B-2 compression standard. This IP core supports most of the standard features, including all entropy encoder options and the most common input sample arrangement formats. It features an adaptable preprocessing stage capable of working in both lossless and near-lossless regimes while supporting all prediction and local sum modes, as well as optimized implementations of all entropy encoder variants. Thanks to this, the IP core can achieve a performance of 1 clock cycle per sample for a subset of the compressor configuration space, even in the near-lossless regime. The goal is to have a technology-independent reusable IP core that can be used as a building block in multiple missions, devices and technologies. This IP core is being developed in the scope of the ESA project 4000136723/22/NL/CRS, and it will join the ESA portfolio of IP cores for space missions.

## 1. INTRODUCTION

Space missions are increasingly incorporating hyperspectral sensors as part of their payloads. The vast amount of information these sensors capture has multiple applications in geology, agriculture, defence and security, to name a few examples [1]. However, handling all those data is challenging due to the limited on-board storage capacity and transmission bandwidths, thus enforcing the adoption of data compression methods. Furthermore, with the increasing resolution of last generations of hyperspectral sensors, lossless compression methods may fall short in terms of required target compression ratios, which motivates the use of lossy compression methods.

In this context, the Consultative Committee for Space Data Systems (CCSDS) has conceived several data compression standards for application on space missions. In particular, the CCSDS-123.0-B-2 standard [2] defines a lossless to near-lossless compression method specifically aimed at multi- and hyperspectral data. Near-lossless compression is achieved by introducing a quantization loop in the prediction block of its predecessor, the CCSDS-123.0-B-1 lossless standard [3]. The quantization losses are controlled by error limits defined by the user. The CCSDS-123.0-B-2 standard also includes a Hybrid Entropy Coder, designed to provide higher compression performance for low-entropy data than the alternative options (i.e., the sample- and the block-adaptive encoders), particularly in near-lossless compression scenarios.

This work presents a new compression IP core for lossless and near-lossless compression of multi- and hyperspectral data that is being developed in the scope of an ESA project. This new IP core is compliant with the CCSDS-123.0-B-2 compression standard and supports most of the standard features, including all entropy encoder options (block-adaptive, sample-adaptive and hybrid), the most common input sample arrangement formats (BIP, BIL and BSQ) and a preprocessing stage capable of working in both lossless and near-lossless regimes while supporting all prediction and local sum modes. The goal is to have a technology-independent reusable IP core that can be used as a building block in multiple missions, devices and technologies.

This new compression IP core has been developed as a technology agnostic, synthesizable VHDL code. The design is based in a previous development, the SHyLoC IP core [4], a hardware implementation of the CCSDS-123.0-B-1 lossless compression standard that already belongs to the ESA IP core repository for space missions. However, the predictor block has been redesigned, incorporating the modules to enable the near-lossless compression regime and a dedicated adaptive control that schedules prediction operations based on the dominant data dependencies for any given compressor configuration. The IP core also incorporates an implementation of

the Hybrid Entropy Coder, whose architecture is deeply pipelined providing 1 sample/cycle throughput in all processing orders [5]. Currently, this new IP core has been fully designed and verified through novel fuzzing verification techniques, that can reach for corner cases that could pass unnoticed otherwise.

The rest of this paper is organized as follows. Section 2 gives an overview of the CCSDS 123.0-B-2 compression standard. Then, 3 introduces the IP core. Section 4 reviews the project status, and finally Section 5 provides some conclusions.

## 2. CCSDS 123.0-B-2 COMPRESSION STANDARD

The CCSDS 123.0-B-2 compression standard defines a low-complexity predictor-based compression algorithm for multi- and hyperspectral data, conceived for space missions. This is an evolution of the CCSDS 123.0-B-1 standard, enabling near-lossless compression in addition to lossless. It is comprised by two parts, prediction and entropy coder.

In the prediction step, every input sample is estimated using a spacial and spectral vicinity of previously pre-processed samples, determined by the local sum and prediction modes selected. The impact of each neighbour sample in the final prediction is scaled by internal weight vectors that are constantly updated, trying to capture the statistical behaviour of the image. Predicted samples are compared with the corresponding input samples, generating prediction residuals that are finally mapped and encoded. In near-lossless compression, these prediction residuals are quantized before mapping and encoding. This allows to reach higher compression ratios at the cost of losing part of the captured information, with a maximum allowed error determined by user.

In lossless compression, input samples can be directly used for prediction. However, in the near-lossless compression regime it is compulsory to incorporate a local decompressor, where input samples are reconstructed from the quantized prediction residuals and some user-defined parameters. These reconstructed samples are used then as neighbour samples for prediction instead of the original samples. This introduces a closed loop with heavy data dependencies that slows computations down, making it harder for hardware designers to achieve high performances with this algorithm. The CCSDS acknowledged this and has introduced new local sum modes in the Issue 2 of the CCSDS 123 standard, which partially alleviate these data dependencies.

Regarding the entropy coding step, the CCSDS 123.0-B-2 standard allows several options. Block- and sample-adaptive entropy coders are inherited as is from Issue 1 of the standard. In addition, a new hybrid encoder is incorporated. This is a key feature of the standard, designed to provide excellent compression performance for low-entropy data, particularly in near-lossless compression scenarios. Independently from the selected entropy coder option, a compression header is appended at the beginning of the compressed bitstream, which specifies the input image characteristics and compression options selected.

The hybrid encoder includes two coding methods, for high- and low-entropy samples, respectively. In the high-entropy coding method, each input sample is individually encoded in a similar way than the sample-adaptive encoder. On the contrary, in the low-entropy method input samples are encoded using one of the 16 available dictionary-based variable-to-variable codes, where an output symbol may represent multiple input symbols. For every input sample, the encoding method is determined by some adaptive code selection statistics that are constantly updated. The compressed bitstream is completed by appending a compressed image tail, which includes the internal status of the encoder by the end of the compression, necessary for the decoding process.

## 3. DESIGN OVERVIEW

Our compression IP core is composed by two main processing blocks: a prediction block and an encoder block, respectively implementing the prediction and entropy coder steps of the CCSDS 123.0-B-2 standard. Both parts have been designed to work in tandem, but they can be separately implemented too, in order to combine them with other preprocessing and/or encoder cores. Several implementation options are offered for both processing blocks, next explained.

As initial step in the IP core development, a design space exploration has been performed. In this phase, several important design decisions have been adopted. The most relevant one is to make the IP core natively supports Band-Interleaved by Line (BIL) and Band Sequential (BSQ) processing orders. Band-Interleaved by Pixel (BIP) format is supported by implementing a reordering module at the data input that performs BIP to BIL conversion

before compression. This is done to avoid the strong data dependencies that are present in BIP processing order in the preprocessing stage in near-lossless compression regime [6]. The sample arrangement format must be determined at compile-time, as different formats imply slight architectural changes in the predictor. In other words, every instance of the compression IP core is suited to compress images in a specific format only. In addition, several optimization strategies in the predictor datapath have been proposed and evaluated, aimed at mitigating the data dependencies and improving performance for a subset of the compressor configuration space [7].

The predictor block offers two different versions: a full compliant version that is capable of compressing both in lossless and near-lossless regimes; and a lightweight version, which omits the near-lossless components, saving some hardware resources but enabling just lossless compression. Finally, there is the possibility of allocating some predictor memory resources externally. Which memory element is externally allocated will depend on the chosen sample arrangement format.

Regarding the entropy coder block, all three encoding methods allowed in the CCSDS 123.0-B-2 compression standard are offered. However, entropy coder selection must be done at compile-time. This means that any implementation of the compression IP core will implement only one of the three encoding methods.

An overview diagram of the compression IP core is shown in Figure 1, including the implementation options mentioned. Predictor and entropy encoder blocks can be disabled, but not at the same time. This allows to implement any of the processing blocks in a standalone manner to be used along with other preprocessors or entropy coders. In addition, the compression IP core includes a configuration interface and unit to enable runtime configuration. This allows to launch compression runs with different configurations. The IP core provides a high diversity of configuration options, including some parameters that can be configured to different values for each band. To handle this, the configuration memory is subdivided in several regions, which are independently enabled or disabled based on which features implements each IP core instance. The configuration interface supports both AMBA AHB and AXI4-Lite protocols. This can be disabled if reconfiguration capabilities are not required, thus saving some hardware resources.

Additional details about the main processing blocks are given in the next subsections.

### 3.1. Predictor block

The predictor block implements the preprocessing stage of the CCSDS 123.0-B-2 compression standard. It is responsible for generating the image and predictor metadata sections of the compression header and then preprocessing the input samples as specified in the compression standard. This block is comprised by a datapath, which performs all operations involving the prediction algorithm; a control unit; a header generation module; and optionally an AMBA master, which stores some intermediate results of the prediction into an external memory.

A single datapath architecture has been devised for this IP core, versatile enough to support all combinations of configuration parameters and processing orders. With that aim, slight adjustments are done based on the selected compile-time IP configuration, such as the size and arrangement of internal memories. The datapath

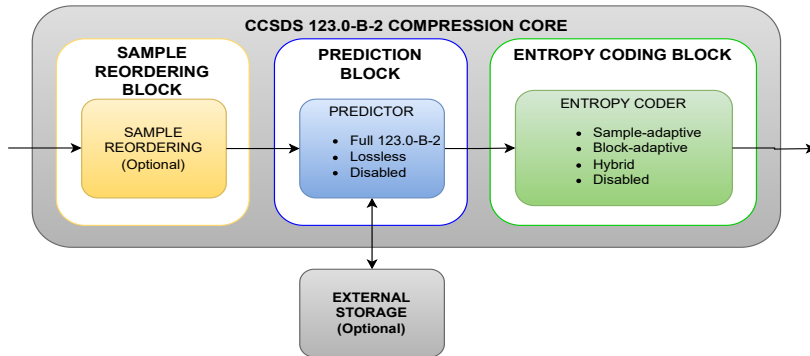


Figure 1. Overview of the CCSDS 123-B-2 compression IP core

architecture is depicted in Figure 2. Size and arrangement of storage elements are determined from compile-time configuration. In addition, which memory element is externally allocated depends on the chosen processing order. The quantizer, which has a configurable division step size, and sample representative modules enable near-lossless compression, and they can be removed when only lossless compression is needed, leaving just the datapath highlighted in green.

The predictor core module encapsulates the dot product, prediction computation and weights update operations. This is a key component in the design, which may hinder the maximum achievable throughput. Because of this, several optimizations have been analyzed and implemented for this component. First, an alternative formula for weights updating has been implemented, which allows to reduce the loop of data dependencies while being mathematically equivalent to the original formula. Then, a simplified prediction computation can be conditionally implemented under certain circumstances, which allows to reduce the hardware consumption and latency for this operation. Finally, two different architectures for the prediction core have been developed. By default, a base pipelined architecture is implemented, which is compatible with the whole configuration space of the IP core but has a limited throughput due to existing data dependencies. However, this can be replaced by a high-performance architecture for a subset of the IP core configuration space, which provides a throughput of one sample per clock cycle at the cost of an increased hardware resources consumption.

To control this complex datapath an adaptable control unit has been developed, which schedules operations based on the runtime configuration selected for each compression run. Thus, the adequate operation mode for each case is enabled, among the following ones:

- **123B2 Baseline.** This operation mode is devised to deal with the strongest data dependencies in the compression algorithm. This enforces a serial processing, in which each sample must be fully preprocessed before starting to preprocess the next one. This has a high impact on processing throughput. However, it is the only operation mode that supports all predictor configurations. Therefore, this alternative should be used only when no other operation modes are feasible for the chosen configuration.
- **123B2 Lossless.** This operation mode is intended to compress in lossless mode in a way backward compatible with the CCSDS 123.0-B-1 compression standard, i.e., without using sample representatives. Therefore, it is compatible with the lightweight, lossless only version of the predictor block. In this situation, data dependencies are partially alleviated, which allows to implement a pipelined scheduling. However, the feedback loop of the weights update limits the maximum achievable throughput.

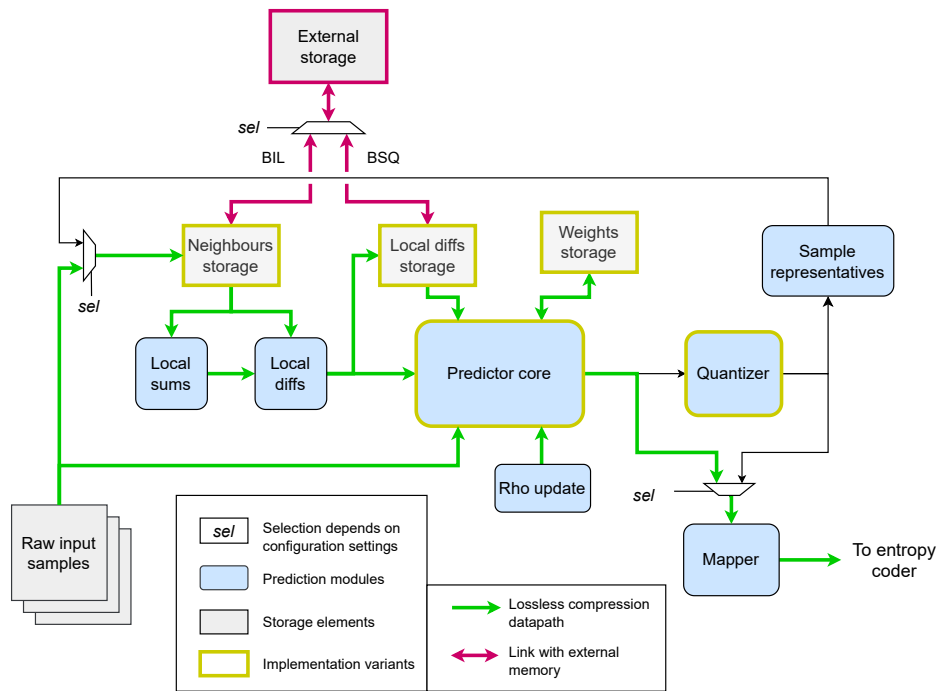


Figure 2. Predictor datapath architecture

- **123B2 Main.** This operation mode is enabled when the combination of local sums and predictor mode allows avoiding the use of the sample representative of the previous sample in the predictor computations. Again, data dependencies are partially alleviated, and a pipelined scheduling is implemented. This allows using a similar scheduling as in the lossless operation mode, but when compressing in near-lossless mode and/or using sample representatives.
- **123B2 High-performance.** This operation mode allows to reduce data dependencies to the minimum and provides a processing throughput of one sample per clock cycle. This mode requires to use the high-performance version of the predictor core, and therefore it is reserved for a subset of the IP core configuration space.

Except for the components that enable near-lossless compression, most of the modules in the prediction block have been reused from the SHyLoC CCSDS 123-IP predictor design. However, those modules have been upgraded to support the new features introduced in Issue 2 of the CCSDS 123 standard.

### 3.2. Entropy coder block

The entropy coding block implements the entropy coding stage of the CCSDS 123.0-B-2 compression standard. It is in charge of, first, building and packaging the compression header by combining the section coming from a previous preprocessing stage with the encoder section internally generated. And then, encoding the incoming preprocessed samples using one of the three encoding options defined in the CCSDS 123.0-B-2 compression standard.

Block-adaptive entropy coder is inherited from SHyLoC CCSDS-121 IP core with minimal changes. Likewise, sample-adaptive encoder is inherited from SHyLoC CCSDS-123 IP core. In addition, a hardware implementation of the hybrid encoder has been incorporated. This implementation distinguishes high- and low-entropy data based on code statistics maintained and updated along the encoding process. High-entropy data are encoded with a codeword from a Golomb-style family of codes, while low-entropy data are encoded with codewords selected out of 16 codetables containing variable-length family of codes, implemented in the device memory as a tree-like data-structure for efficient lookups. A top-level block design of the Hybrid Entropy Coder architecture is presented in Figure 3, displaying all the main functional units of the design.

The architecture of Hybrid Entropy Coder supports the most common pixel orderings (BIL, BIP and BSQ), without loss in performance data-rate. This mainly affects the Adaptive Code Selection Statistics (ACSS) unit of the design. The processing order, selected at compile time, determines the architecture for ACSS unit among three possible options. This selection affects mainly the feedback path of the ACSS. The simplest architecture corresponds to BIP ordering, where the feedback path just implements a FIFO. For BSQ, an elastic buffer is implemented in addition. Finally, BIL has the most complex architecture, requiring a dedicated control logic in addition to the FIFO and the elastic buffer.

The proposed implementation provides a performance of 1 sample per clock cycle, thanks to a deeply pipelined architecture exploiting a systolic, latency insensitive design pattern, with elastic buffers serving as pipeline registers [5]. This design pattern avoids additional controllers for flow control or excessive buffering to handle sink side stalls.

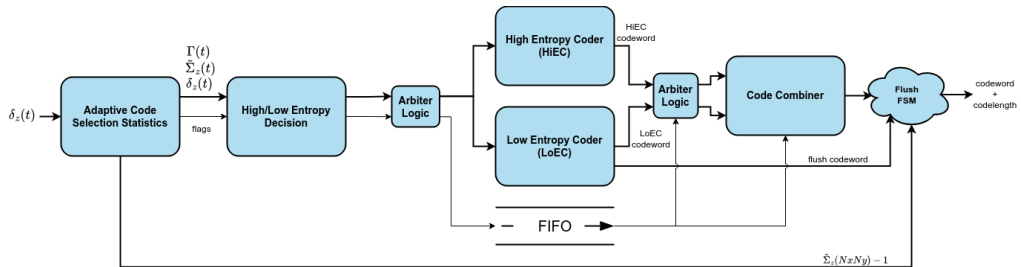


Figure 3. Hybrid Entropy Coder top-level block diagram

## 4. PROJECT STATUS

The proposed compression IP core is being developed in the scope of the ESA project 4000136723/22/NL/CR. The project is currently in its final stages. The IP core VHDL description and verification are completed, and only the validation and technology mapping steps are left. Once all project tasks are finished, the compression IP core will join the ESA portfolio for space missions.

As previously introduced, a design space exploration has been performed prior to the IP core design itself, which was mainly focused on the evaluation of different implementation strategies for the predictor block. In this step, it has been decided the processing orders that are natively supported, the implementation of the quantization module (iterative division against look-up table based approach) and different optimization strategies for the most critical operations in the predictor datapath.

The compression IP core has been developed as a technology agnostic soft core, described in VHDL. Around the IP core, several automation scripts have been developed for an easy integration with some of the most common EDA tools, such as Mentor Questasim, Xilinx Vivado or NanoXplore Impulse. The IP core has been extensively verified following a twofold approach. First, a traditional verification campaign with predefined testcases has been conducted, in order to prove the project requirements and assess code coverage. This approach has been used to verify individual processing blocks, including the sample reordering block and AMBA interfaces, as well as at system level. This traditional verification has been complemented with an alternative approach through fuzzing techniques. These approaches can generate thousands of different configuration sets and input patterns in an automated manner to exercise the design under test in multiple ways, so they can reach for corner cases that could pass unnoticed otherwise, considering the large configuration space of the IP core. With this approach we managed to run around one million tests, in both lossless and near-lossless compression regimes. For the IP core verification, a group of self checking test benches has been developed. These test benches compare the output of the design with a golden reference generated by software. For this purpose, CCSDS 123.0-B-2 reference software from CNES has been used, along with in-house software developments for verification of individual processing blocks.

Once fully verified, a preliminary synthesis of the IP core has been performed using Synopsys Synplify software. Two different IP core configurations have been selected: one that uses the base configuration of the predictor core, and another one with the high-performance version. Differences in configuration parameters between both instances are summarized in Table 1. However, both share some commonalities too: BIL processing order, predictor block with near-lossless compression capabilities, hybrid encoder, configuration interface enabled (AHB interface), no external memory, band-independent parameters, maximum number of bands used for prediction equal to 3, 32 bits maximum output width, maximum dynamic range of 16 bits, and maximum image dimensions of 680 columns x 512 rows x 256 bands (i.e., targeting AVIRIS sensor).

Parameter	Base	High-performance
Prediction modes	Full + reduced	Reduced only
Local sum modes	Wide + narrow	Narrow only
Max. weight resolution	13	19
Register size range	[40,48]	48
Weight update scaling exponent max. range	[-3,1]	[-1,3]
Weight update scaling exponent max. interval	11	6
Quantization fidelity methods allowed	Absolute + relative	Absolute only
Max. error limit bitdepth	8	10
Max. sample representative resolution	2	4
Max. Initial count exponent	1	1
Max. Rescaling counter size	4	6
Max. Unary length limit	16	16

Table 1. IP core configurations for preliminary synthesis

Results of preliminary synthesis for the Xilinx UltraScale technology (XCKU060) are shown in Table 2. According to these results, hardware resources consumption is kept at a reasonable footprint, making this design to easily fit in FPGAs commonly used on space missions. Memory resources usage depends greatly on the maximum image dimensions allowed, and also will increase if band-dependent parameters are enabled. It should be noted that, despite the clock frequency achieved for the high-performance configuration is slightly lower than the base one, only the high-performance is able to achieve a throughput of one sample per clock cycle. Therefore, it achieves a better performance at the cost of an increased hardware resources consumption, especially DSPs.

Resource	Base	High-performance
LUTs	15365 ( $\sim 4\%$ )	18212 ( $\sim 5\%$ )
FFs	6530 ( $< 1\%$ )	7544 ( $\sim 1\%$ )
BRAMs	140 ( $\sim 12\%$ )	138 ( $\sim 12\%$ )
DSPs	21 ( $< 1\%$ )	56 ( $\sim 2\%$ )
Clk frequency (MHz)	130.9	129.8

Table 2. Preliminary synthesis results for XCKU060

## 5. CONCLUSIONS

A new compression IP core has been presented in this work. This is aimed at multi- and hyperspectral image compression, compliant with the CCSDS 123.0-B-2 compression standard. It supports most of the standard features and modes, constituting an all-purpose compression solution for this kind of data. The IP core implements a predictor block that provides lossless to near-lossless compression capabilities. This has been designed with flexible architecture and control to adapt the hardware resources usage and performance to the selected configuration, even reaching a throughput of 1 sample per clock cycle for a subset of the IP configuration space. The IP core also implements all entropy coder choices defined in Issue 2 of the compression standard, including the brand new hybrid encoder, which allows to reach higher compression ratios when compressing in the near-lossless regime. The IP core natively supports BIL and BSQ processing orders, while BIP is supported by implementing a reordering module at data input that performs BIP to BIL conversion.

This IP core has been developed in the scope of ESA project 4000136723/22/NL/CR, which is in its final stages. Currently, the design description is completed, and it has been extensively verified using a mix of traditional verification approach with pre-generated testcases along with automated fuzz testing approach to exercise possible corner cases. Preliminary synthesis results demonstrate that the design has a moderate hardware footprint. Once the project finishes, this compression IP core will join the ESA portfolio and will become available as building block for space missions.

## ACKNOWLEDGEMENT

The research leading to these results has been funded by ESA contract 4000136723/22/NL/CRS. The authors want to thank CNES for lossless and near-lossless software that has been used for generation of reference compressed images for verification purposes.

## REFERENCES

1. S.-E. Qian, “Hyperspectral satellites, evolution, and development history,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 7032–7056, 2021.
2. Consultative Committee for Space Data Systems, *Low-Complexity Lossless and Near-Lossless Multispectral and Hyperspectral Image Compression*, *CCSDS 123.0-B-2*. CCSDS, February 2019.
3. —, *Lossless Multispectral and Hyperspectral Image Compression*, *Recommended Standard CCSDS 123.0-B-1*. CCSDS, May 2012.
4. Y. Barrios, A. Sánchez, L. Santos, and R. Sarmiento, “SHyLoC 2.0: a versatile hardware solution for on-board data and hyperspectral image compression on future space missions,” *IEEE Access*, vol. 8, pp. 54 269–54 287, 2020.
5. P. Chatziantoniou, A. Tsigkanos, D. Theodoropoulos, N. Kranitis, and A. Paschalis, “An efficient architecture and high-throughput implementation of ccstds-123.0-b-2 hybrid entropy coder targeting space-grade sram fpga technology,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 58, no. 6, pp. 5470–5482, 2022.
6. A. Sánchez, I. Blanes, Y. Barrios, M. Hernández-Cabronero, J. Bartrina-Rapesta, J. Serra-Sagrístà, and R. Sarmiento, “Reducing Data Dependencies in the Feedback Loop of the CCSDS 123.0-B-2 Predictor,” *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022.
7. Y. Barrios, J. Bartrina-Rapestà, M. Hernández-Cabronero, A. Sánchez, I. Blanes, J. Serra-Sagrístà, and R. Sarmiento, “Removing Data Dependencies in the CCSDS 123.0-B-2 Predictor Weight Updating,” *IEEE Geoscience and Remote Sensing Letters*, vol. 21, pp. 1–5, 2024.