IMPLEMENTATION OF CLOUD DETECTION AND PROCESSING ALGORITHMS AND CCSDS-COMPLIANT HYPERSPECTRAL IMAGE COMPRESSION FOR CHIME MISSION

Yubal Barrios[(1)], Pedro Rodríguez [(2)], Antonio Sánchez[(1)],
M. Isabel González[(2)], Luis Berrojo[(2)] and Roberto Sarmiento [(1),]

(1) Institute for Applied Microelectronics, University of Las Palmas de Gran Canaria,
35001 Las Palmas de Gran Canaria, Spain
(2) Thales Alenia Space (TAS) in Spain, 28760 Tres Cantos, Madrid, Spain.

Corresponding authors: Yubal Barrios (ybarrios@iuma.ulpgc.es)
and Pedro Rodríguez (pedro.r@thalesaleniaspace.com).

ABSTRACT

Hyperspectral sensors are increasing their presence on-board satellites because they provide relevant information for the scientific community in many applications. This is the reason why ESA has included the Copernicus Hyperspectral Imaging Mission for Environment (CHIME) in the future Copernicus 2.0 program. CHIME shall provide continuous spectral coverage in VNIR and SWIR spectral domain (covering approximately 220 bands between 400nm and 2500nm). Acquisitions shall have typically 106Msps, with a dynamic range of 16 bits per sample and acquired in Band-Interleaved by Line (BIL) order, what leads to input data rates up to almost 2Gbps, which is the worst case value considered for the demonstrator. However, clouds are estimated to cover more than 54% of the Earth's land area and 68% of the oceans. Many scientific applications are useless in presence of clouds, making more than half of the acquired scenes unusable. At sensor level, on board detection of cloudy areas and compression combined are proposed to transmit sensor information to ground with a limited downlink bandwidth.

The work presented in this paper, done in CHIME (phase A/B1) framework, presents a demonstrator implementing cloud detection and processing algorithms and hyperspectral image compressor based on CCSDS 123.0-B-2 standard, recommended by the Consultative Committee for Space Data Systems.

1. DESIGN DESCRIPTION

Figure 1 shows a functional block diagram of the design implemented in the demonstrator. Main modules are:

- Image reception: The demonstrator will receive hyperspectral images at a data rate up to 2Gbps (16 bits samples captured with a 125MHz clock). The input image will be stored in a mass memory.
- Cloud detection over selected bands: At the same time the image is being received, the cloud detection algorithm is performed over some selected bands. The result will be a cloud mask to discriminate the clouds from the useful information.
- Cloud processing: The image stored in the memory is read back to perform the processing. All the samples will be read one by one: if the sample is cloud, it will be simplified; if it is not cloud, the complete useful sample will be considered. The result is a hyperspectral data cube, in the same format as the input one, with some losses in the cloud areas to improve the posterior compression.
- CCSDS compression: all these samples are compressed according CCSDS 123.0-B-2 standard. The compression can be lossless or near lossless.
- Data formatting and transmission: the compressed image is transmitted together with the cloud mask at the same data rates than the input image.
- TMTC module: This module allows to configure the design. The standard buses AXI and AMBA AHB will be used to transfer configuration information to the rest of the modules.

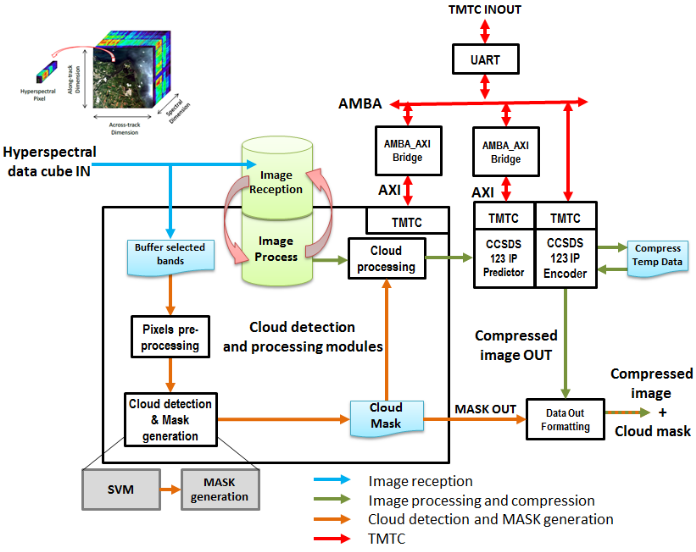


Figure 1. Demonstrator design block diagram

2. CLOUD ALGORITHMS

Cloud algorithms are composed by detection and processing stages. The cloud detection is performed by a Support Vector Machine approach (SVM). The SVM algorithm is pixel-based and it is followed by a simple filtering in order to reduce the false positive detection. The output of this algorithm is a spatial mask, including one bit per each hyperspectral pixel of the image, indicating if it is cloud (mask = ‘1’) or not (mask = ‘0’).

For the pixels detected as cloud, a pre-quantization is done, to improve the posterior compression in these less useful areas. Among the three cloud processing possibilities considered and compared in [5], the one retained is the pre-quantization. Typical performances of compression ratios are available in this reference.

Even for cloudy areas, some bands of the image can have scientific or commercial utility, therefore some interesting selected bands can be excluded from the processing.

Refer to [5] for more information on the definition of the cloud algorithms.

3. CCSDS 123.0-B-2 ADAPTATION TO CHIME

3.1. Algorithm overview

The compression solution implemented in the CHIME demonstrator is based on the recent CCSDS 123.0-B-2 standard, thought for the near-lossless compression of multi- and hyperspectral images [1]. The algorithm presented in this recommendation has backward compatibility with its predecessor, CCSDS 123.0-B-1 standard [2], being able to compress in lossless mode by setting the maximum error limit to 0. In addition, the CCSDS 123.0-B-2 includes new features not presented in the previous standard, such as the narrow local sums, avoiding to use the sample at the left of the current one during this prediction stage (favouring pipeline implementations in hardware); or the new hybrid encoder.

The predictor is based on a linear prediction, which estimates the value of each image sample based on the values of nearby samples in a small three-dimensional neighbourhood. Two new modules are introduced regarding the structure of its predecessor:

- Quantizer: this module introduces losses in the prediction chain, taking into account the maximum error limit specified by the user. Then, its output is mapped to obtain the mapped residuals δz (t), the input of the entropy coding stage.
- Sample representative: also known as local decompressor, this module reconstructs the samples, taking as input the quantized residual $q_z(t)$. Prediction calculations are performed using these reconstructed samples, so they can be replicated by the decompressor, which has no access to the original samples.

A general overview of the CCSDS 123.0-B-2 algorithm is shown in Figure 2.

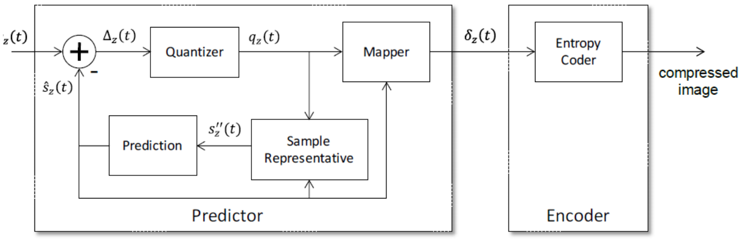


Figure 2. CCSDS 123.0-B-2 block diagram [1]

3.2. Implementation

The CCSDS 123.0.B-2 compression standard presents a great amount of configuration parameters that influence the compression performance. For the CHIME mission, most of these parameters have been set to fixed values to simplify the implementation, while at the same time optimizing the compressor performance according to the mission requirements. Some choices are imposed by the mission sensor, such as the spectral and spatial resolution per line, as well as the sample processing order in BIL. Other choices are motivated by the goal of obtaining a data throughput as high as possible, such as the settings for the number of previous bands used in prediction, local sum and prediction mode. For the rest of configuration parameters of the predictor, a parameter tuning has been performed in order to find a configuration which allows to maximize the compression ratio for the representative test vectors.

With respect to the quantizer settings, the compressor implemented in this work supports only the band-independent absolute error limit fidelity control mode, for simplicity. The absolute error is implemented with a resolution of 6 bits, which allows a range of values between 0 and 63, runtime configurable. The sample representative settings are all fixed with the only exception of the band-independent sample representative offset ψ_z. Lossless compression is enabled by setting both the absolute error limit and the sample representative offset to 0.

This module has been modelled with custom interfaces, based on a simple handshaking protocol to be easily connected with the entropy coder. The predictor is the master asking for a new sample when the previous one is already processed. The configuration of the run-time parameters is done by a dedicated AXI-Lite interface.

Regarding the entropy coding stage, the block-adaptive encoder proposed in the CCSDS 121.0.B-2 compression standard [3] is selected. This module was developed in VHDL in a previous work, named SHyLoC, a pair of IP cores for compressing hyperspectral images and currently available at the ESA IP Core library [4].

4. VHDL IMPLEMENTATION AND SYNTHESIS

Cloud detection and processing modules and the compression encoder have been coded by means of VHDL RTL description. In case of the predictor compliant with the CCSDS-123.0-B-2 standard, it was modelled in C language and implemented by using High Level Synthesis (HLS) techniques, that automatically generate an equivalent RTL description.
VHDL simulations have been done in workstations with Mentor Graphics QuestaSim (version 10.6, revision 2016.12). The VHDL has been synthesized with Synopsys Synplify Premier (version P-2019.03-SPI) and the place & route into the Xilinx device has been performed by Vivado(version 2016.4).
The design has been loaded into a Xilinx KU040 FPGA mounted in the demonstrator main board.

5. TEST VECTORS

For the validation of the design implemented in the demonstrator three test vectors have been chosen. A test vector is an ensemble of data needed to run the tests and to verify the results. Each test vector includes the input data cube and associated information, intermediate data and resulting output image.
The images for the demonstrator have been generated from the AVIRIS images, with and without clouds, extended by mirroring in order to achieve the size expected for the CHIME instrument. The images have 16 bits per sample and are sent in BIL order.
The three test vectors have the following characteristics:

- Test vector 1: Image of size 1024 lines x 1536 pixels x 239 bands to be compressed in lossless mode (the image can be seen in Figure 3).
- Test vector 2: Image of size 1024 lines x 2048 pixels x 231 bands to be compressed in lossless mode (Figure 4).
- Test vector 3: Same image than in test vector 1 but compressed in lossy conditions.

Figure 3. Image for test vectors 1 and 3

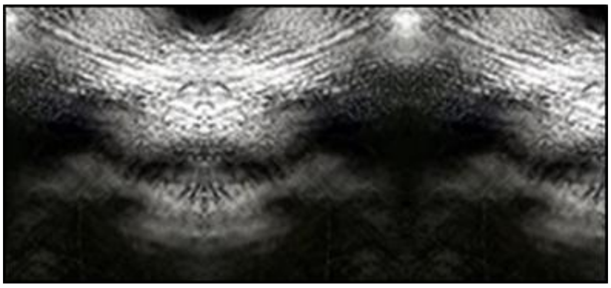


Figure 4. Image for test vector 2

6. VALIDATION PLAN

The Figure 5 provides a schematic representation of the validation tests done.

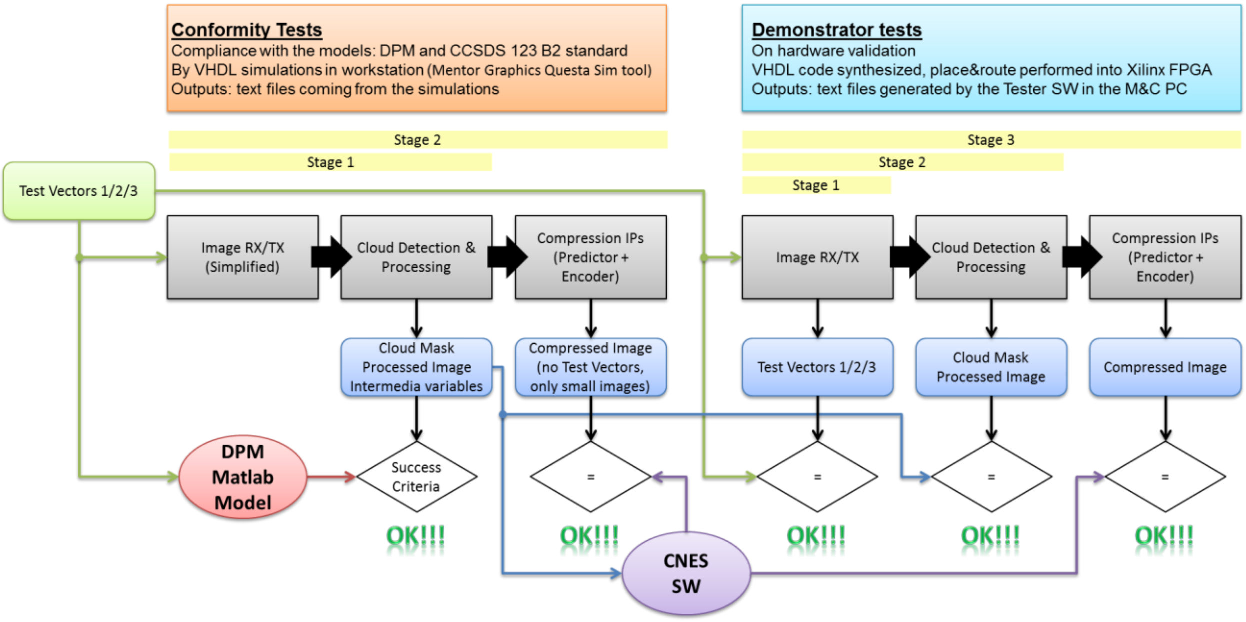

Conformity Tests
Compliance with the models: DPM and CCSDS 123 B2 standard
By VHDL simulations in workstation (Mentor Graphics Questa Sim tool)
Outputs: text files coming from the simulations
Demonstrator tests
On hardware validation
VHDL code synthesized, place&route performed into Xilinx FPGA
Outputs: text files generated by the Tester SW in the M&C PC
Stage 2
Stage 1
Stage 3
Stage 2
Stage 1
Test Vectors 1/2/3
Image RX/TX (Simplified)
Cloud Detection & Processing
Compression IPs (Predictor + Encoder)
Image RX/TX
Cloud Detection & Processing
Compression IPs (Predictor + Encoder)
Cloud Mask Processed Image Intermedia variables
Compressed Image (no Test Vectors, only small images)
Test Vectors 1/2/3
Cloud Mask Processed Image
Compressed Image
DPM Matlab Model
Success Criteria
=
=
=
=
OK!!!
OK!!!
OK!!!
OK!!!
OK!!!
CNES SW

Figure 5. Validation plan

The validation of the VHDL implementation of the CHIME design has been performed with two kind of tests in very different environments:

- Conformity tests: compliance with existing models or standards. This kind of tests has been performed by simulation of the VHDL code in a workstation. A verification plan based in a number of artificial images has been used for this purpose prior to use the test vector set mentioned in section §5.
- Demonstrator tests: on-hardware validation. Once the VHDL is verified by simulations, it has been loaded into the FPGA. Additional set of tests has been done to confirm the design correct behaviour in the real world (not only by simulations).

7. CONFORMITY TESTS RESULTS

The first set of tests performed to validate the VHDL has been called conformity tests and has been done by simulations of the VHDL and comparison with the models or standards. Two main software have been used to test different parts of the design:

- To validate the cloud algorithms the processed images are compared against the results of the so called Data Processing Model (DPM). This is a Matlab model developed by TAS implementing the cloud algorithms designed during the CHIME pre-developments phase.
- To verify the compression implementation, the CNES software that implements a compressor compliant with the CCSDS 123.0-B-2 standard has been used (more information about this software can be obtained from reference [6]). Both implementations of the same standard have to be bit to bit comparable.

Several analyses are done to assure that the VHDL code fulfil the requirements. In particular several success criteria have been defined as necessary to consider a test as successfully executed. The conformity with the models has been agreed assuming that these success criteria are met.

7.1. Stage 1: Cloud detection and processing conformity report

The first success criterion agreed to verify the implementation of the cloud algorithms is:

<u>SUCCESS CRITERION 1 for conformity tests:</u>
All the quantified variables in VHDL shall have as maximum a 2% error comparing with the corresponding variable in the DPM Matlab software.

Matlab variables are real numbers and in VHDL exists a limitation in the number of bits. Therefore, quantization has to be done and an error is produced. This error has to be limited.

The comparison between VHDL and DPM has been done for several relevant variables for all the selected bands. For each variable in each selected band the error is computed (in absolute value and in percentage).

The quantified variables selected for the analysis are:

- Pre-processing coefficients: Used to convert from input raw samples to radiance and after that to reflectance.
- Radiance: The values for the samples of the selected bands converted to radiance.
- Reflectance: The values for the samples of the selected bands converted to reflectance.
- Hyperplan coefficients: The coefficients computed according the Support Vector Machine approach.
- Total hyperplan result: the addition of all previously calculated coefficients.

The analysis of the quantization error has been performed for the test vectors 1 and 2 (test vector 1 and 3 are identical at cloud detection and processing module level). With this analysis it has been verified that all the errors are under the 2% admitted. Therefore, the requirement is met.

<u>SUCCESS CRITERION 2 for conformity tests:</u>
For the cloud mask a maximum of 1% of False Positives shall be admitted comparing with the mask obtained in the DPM software.

The cloud mask from the VHDL simulations has been compared with the one generated with the DPM model.

Three kind of errors have been defined:

- False positives: A no cloud pixel according to DPM is considered as cloud by the VHDL. This is the worst situation, as useful information will be processed and interesting data will be lost.
- No detected clouds: A cloud in DPM has not been detected as cloud by the VHDL. A big number of these errors will make worse the compression ratio, but no useful information will be lost.
- Total errors: the addition of false positives and no detected clouds.

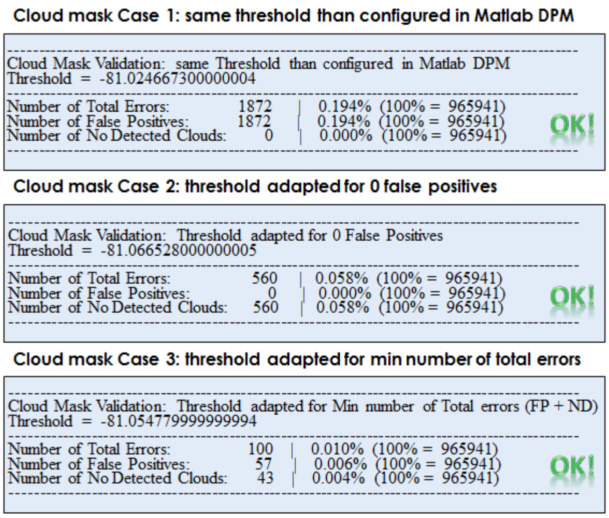
The comparison of the cloud mask has been performed in three cases. In the SVM algorithm a computed value, hyperplan coefficient, is compared with a fixed configured threshold. The pixels over this limit are defined as useful information and those below the threshold are detected as clouds.

In a first step the same SVM threshold has been configured in VHDL and in DPM. For this case, all the errors obtained in the simulations are false positives (for test vectors 1 and 2), which is the worst case as explained. The differences are due to the quantization of the variables. As these variables are in most cases unsigned signals the quantization always introduces an error in the same direction. The result is that the hyperplan coefficients in DPM are always higher than the corresponding values in VHDL. Consequently, there are some pixels that in DPM are detected as no clouds (hyperplan coefficient over the threshold) and in VHDL are under the threshold (and considered as clouds): being false positives. To solve this issue the threshold defined in the VHDL can be configured a little bit lower than in DPM to compensate the quantization effect. The adjustment has been done in two conditions (case 1 is the normal situation without threshold adjustment):

- Case 2: adjust the threshold to remove totally the false positives (as this is the worst situation)
- Case 3: adjust the threshold to minimise the total number of errors.

The nominal value of the threshold is -81,0246673 and the adjustment is under 0,04. These values show the very fine tuning required.

In all the cases, the absolute number of errors have been computed and also the percentage comparing with the total number of pixels detected as clouds in the image. The obtained values are shown in Figure 6.



Cloud mask Case 1: same threshold than configured in Matlab DPM

```
--------------------------------------------------------------------------------
Cloud Mask Validation:  same Threshold  than configured  in Matlab DPM
Threshold  =  -81.024667300000004
--------------------------------------------------------------------------------
Number  of  Total  Errors:            1872    |    0.194%  (100% =  965941)
Number  of  False  Positives:         1872    |    0.194%  (100% =  965941)      OK!
Number  of  No Detected  Clouds:         0    |    0.000%  (100% =  965941)
--------------------------------------------------------------------------------
```

Cloud mask Case 2: threshold adapted for 0 false positives

```
--------------------------------------------------------------------------------
Cloud Mask Validation:  Threshold  adapted for 0 False  Positives
Threshold  =  -81.066528000000005
--------------------------------------------------------------------------------
Number  of  Total  Errors:             560    |   0.058%  (100% =  965941)
Number  of  False  Positives:            0    |   0.000%  (100% =  965941)      OK!
Number  of  No Detected  Clouds:       560    |   0.058%  (100% =  965941)
--------------------------------------------------------------------------------
```

Cloud mask Case 3: threshold adapted for min number of total errors

```
--------------------------------------------------------------------------------
Cloud Mask Validation:  Threshold  adapted for Min  number  of Total  errors (FP + ND)
Threshold  =  -81.054779999999994
--------------------------------------------------------------------------------
Number  of  Total  Errors:             100    |  0.010%  (100% =  965941)
Number  of  False  Positives:           57    |  0.006%  (100% =  965941)      OK!
Number  of  No Detected  Clouds:        43    |  0.004%  (100% =  965941)
--------------------------------------------------------------------------------
```

Figure 6. Test vector 1 Cloud mask errors statistics

Figure 7 and Figure 8 show the cloud mask generated for test vector 1 without threshold adjustment and with tuning to have minimum number of total errors.

Pixels in white are the detected clouds and in light blue the no cloud information as computed by DPM. The red dots are false positives obtained in VHDL and the black ones are the no detected clouds. It has to be remarked that these coloured points are quite bigger than the real pixel size in order to make them visible, therefore the figures give a sensation of more errors than there are in the reality. The figures provide visual information on where the errors are located. They are always in the limits between cloud and no cloud areas, showing that the SVM algorithm is working correctly as cloud pixel discriminator.

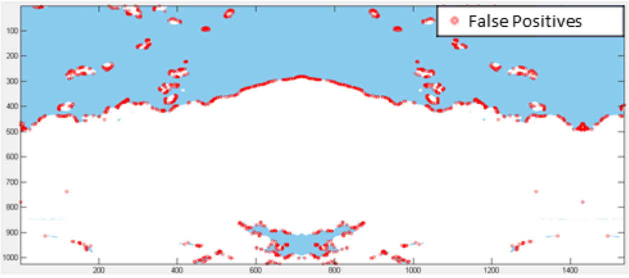


Figure 7. Test vector 1 cloud mask case 1: same threshold than in DPM

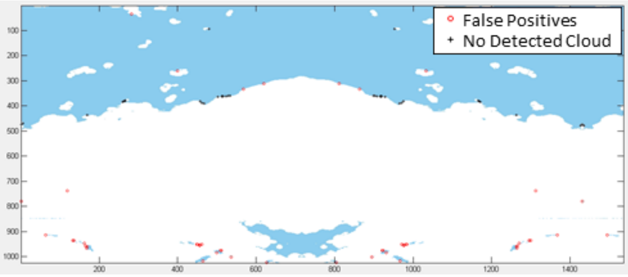


Figure 8. Test vector 1 cloud mask case 3: threshold adapted for minimum number of Total errors

From the results obtained it is concluded that threshold adjustment significantly improves the performance of the VHDL implementation. Same tests have been executed also for test vector 2 with same conclusions.

In addition to the analysis performed for the cloud mask generation, also the processing of the detected cloud pixels has been tested. The processing of the cloud samples is an algorithm that does not depend on quantization. Considering the same cloud mask, the processed cloud samples should be exactly the same in VHDL and in the DPM. Therefore, in this case the success criterion to consider the processing correctly implemented in VHDL is that the error between Matlab and VHDL has to be always zero.

The results obtained in the simulations show that no differences exist between the processed information coming from DPM and from the VHDL and consequently the VHDL implementation is correct.

SUCCESS CRITERION 3 for conformity tests:

The Data rate for the Cloud algorithms shall be up to 2Gbps. Processing one 16 bits sample each 125MHz clock cycle.

Data latency and throughput have been measured and the results obtained are:

- The clouds detection is performed in real time at the same time the image is being received (no extra delay is inserted due to the application of the detection algorithms).
- The clouds processing introduces a single clock cycle of 125 MHz delay.

For cloud detection and cloud processing it has been validated by simulations that both can be performed with the required data rates without problems.

7.2. Stage 2: CCSDS 123.0-B-2 compression modules conformity report

For the validation of the compression modules two main success criteria have been defined.

SUCCESS CRITERION 4 for conformity tests:
The compressed data shall be bit to bit comparable to the data obtained from the CNES software.

To verify that this criterion is met, some simulations have been performed. By means of these simulations a file containing the input samples to the compressor (inputs to the predictor module) is generated. At the same time the outputs of the compression module (outputs of the encoder) are also captured into an output file. These stimuli are used for the comparison against the CNES software. The input file is also the same input vector used for the CNES software. The output file from the VHDL simulations has been compared against the output results of CNES software.
The tests to verify the compression block at simulation level have been done only with small images, due to constraints in the capability of the simulation tools. Simulating the big images considered in the defined test vectors would produce very long simulations, that were incompatible with the project needs. The complete images have been validated on the hardware in the demonstrator tests.

SUCCESS CRITERION 5 for conformity tests:
The Data rate for the compression modules shall be up to 2 Gbps. Compression of one 16 bits sample each 125 MHz clock cycle.

The objective for the design is to reach one clock cycle per sample. For the CHIME demonstrator, the predictor has been implemented by means of the HLS tool that generates automatically the VHDL code from a high level language (C code). This automatically-generated VHDL cannot reach the same performances and therefore for this demonstrator the compression core needs more than one clock cycle per sample.
The encoder is implemented in VHDL RTL and can work at one clock/sample, as it has been validated by simulations of this module isolated from the predictor.
Analyses have been done during the project to verify the bottle necks in the predictor implementation and it has been concluded that with an ad hoc VHDL RTL implementation the requirement can be met

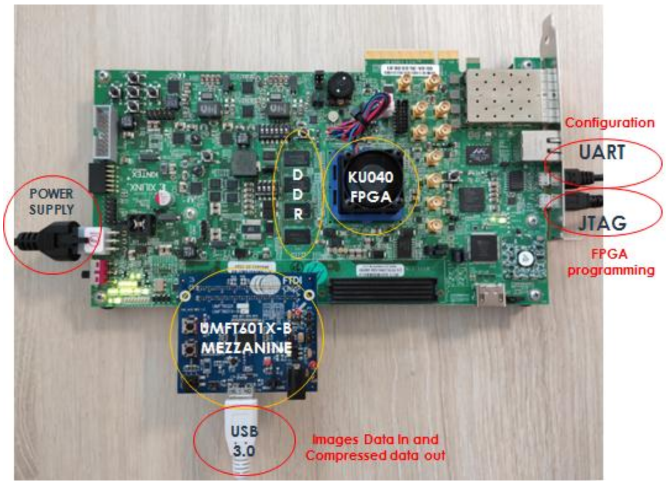


Figure 9. HW description

8. DEMONSTRATOR SET-UP

Figure 9 includes a picture of the boards used in the demonstrator. In this picture, the main hardware components of the boards and the external interfaces are marked. Besides the boards a standard laptop complete the set-up.
The main elements of the demonstrator are summarised in the following list:

- DUT board: The KCU105 evaluation board from Xilinx has been used to implement the Design-Under-Test (DUT). This board contains a Kintex UltraScale FPGA, the XCKU040-2FFVA1156E device, where the design is loaded for its validation.
  This board contains all the hardware needed, excepting the interfaces to send and receive the images. To implement the data interfaces extra hardware was added by using a FMC connector.
- Test Board: This mezzanine board has been mounted in the DUT board. The UMFT601X-B is an evaluation/development module with FMC connector for interfacing FTDI's FT601Q USB 3.0 integrated circuit. This ASIC allows for bridging the USB3.0 host (connected to the Monitoring and Control PC) to a FIFO bus. The FPGA in the DUT board interfaces this FIFO to receive and transmit the samples. The USB 3.0 to FIFO bridge allows to send the images and receive the compressed information at a maximum data rate of 2,1 Gbps, compatible with the needs of the project.
- Monitoring and Control PC (M&C PC): A standard laptop that includes some software tools to perform several functions:
  - Allows to program the FPGA through the JTAG link.
  - Provides a Graphical User Interface to control and monitor the test execution.
  - Reads the files containing the images to be sent and controls the USB interface to transmit this information to the test mezzanine.

- Recovers the compressed data from the USB link and saves them into an output file to be post-processed.
- Transmits and also reads the configuration information to/from the DUT board through the UART serial configuration bus.
- Performs automation of all the tasks needed to run one test.
- Allows to run the DPM Matlab model, to verify the cloud algorithm results.
- Allows to run the CNES software to validate the compressor modules results.

9. DEMONSTRATOR TESTS RESULTS

Due to the complexity of the design the VHDL verification in the demonstrator has been performed following an incremental method. Three stages have been defined for the validation:

- Stage 1: Image reception into the demonstrator and transmission back to the Monitoring and Control PC validation (image RX/TX).
- Stage 2: Image RX/TX + cloud detection and processing algorithms validation.
- Stage 3: Image RX/TX + cloud algorithms + CCSDS 123.0-B-2 compressor validation.

These three validation steps are also those detailed in the Figure 5, that provides the validation plan definition. Figure 10 shows a block diagram of the demonstrator set-up highlighting the elements validated in each stage.

9.1. Stage 1: Image RX/TX validation

In the first stage of the VHDL verification, the correct behaviour of the procedure to send a hyperspectral image to the demonstrator has been tested. Also, the process to transmit the image from the demonstrator back to the M&C PC to be analysed a posteriori.

In this phase of the test campaign the cloud processing and compression modules have not been included yet and therefore the image transmitted is the same image received.

This first validation stage is planned because the procedure of sending and receiving the images in the demonstrator set-up is very complex, involving a lot of elements including software, hardware and firmware. Also, the large size of the hyperspectral images defined for the tests and the high data rates needed to characterize realistically the functionality of the real CHIME detector imply very high performances that entail quite great added difficulties.

For these tests next success criterion has been defined:

SUCCESS CRITERION demonstrator tests stage 1:
The image received back by the M&C PC from the demonstrator board shall be equal to the image sent previously from the control computer to the KCU105 evaluation board for all the defined test vectors.

The test has been executed for the test vectors 1 and 2 and it was possible to receive the same image sent in all the cases, so the tests have been carried out successfully

9.2. Stage 2: image RX/TX + cloud detection and processing algorithms validation

In the second stage of the on-hardware verification of the demonstrator design, the cloud detection and processing algorithms have been added to the VHDL.

In this case, the image transmitted will not be the same than the image received, as the cloud pixels have been modified by pre-quantization. The results obtained in the simulations of the VHDL performed for the conformity tests have been used as reference.

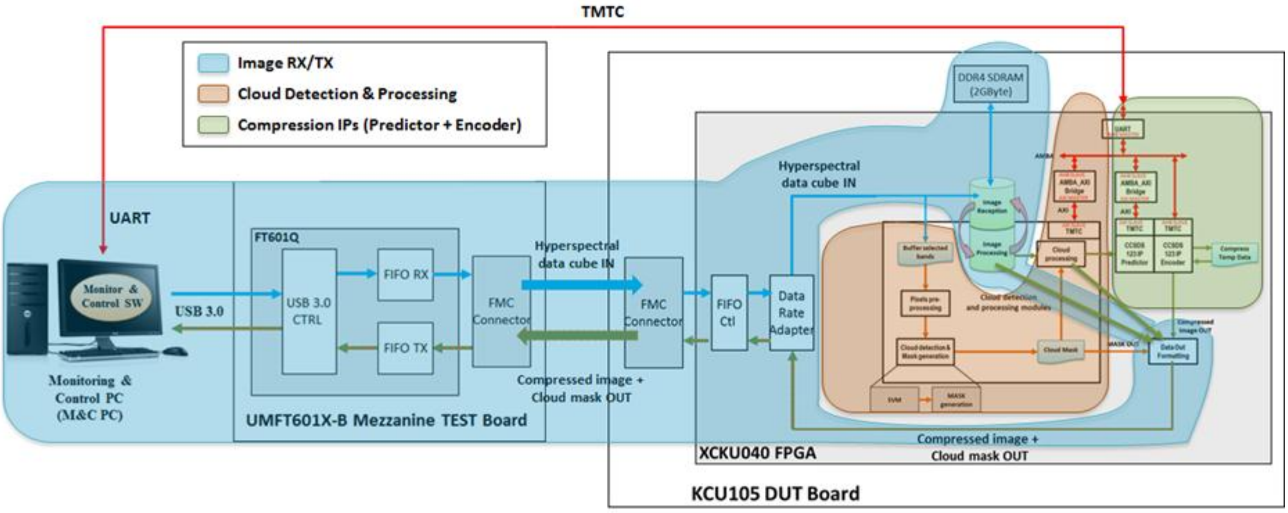


Figure 10. Demonstrator tests stages

According to this, the success criterion defined for the second stage is listed in the next squared text:

SUCCESS CRITERION demonstrator tests stage 2:
The processed image and the cloud mask received back by the M&C PC from the demonstrator board shall be exactly the same than those obtained in the VHDL simulations executed for the conformity tests.

Test vectors 1 and 2 have been tested. For both, the same cloud mask and the same processed image have been received than in the corresponding conformity simulations, so the success criterion is met.

It is remarkable that, as these files obtained from the demonstrator are the same than those generated by the simulations, the fulfilment of the success criterion for the demonstrator tests stage 2 implies necessarily the fulfilment of the success criteria defined in §7.1.

9.3. Stage 3: image RX/TX + cloud algorithms + CCSDS 123.0-B-2 compressor

In the third and final validation stage, the CCSDS 123.0-B-2 compression modules have been inserted in the VHDL project: both predictor and encoder. For the compression the success criterion is the same than the one fixed for the conformity tests of these modules:

SUCCESS CRITERION demonstrator tests stage 3:
The compressed data shall be bit to bit comparable to the data obtained from the CNES software.

In this stage, the verification has been done with small images initially for debugging. After that, the three test vectors have been verified. test vectors 1 and 2 define lossless compression of different images. Test vector 3 performs a lossy compression.

In all the cases the output files obtained in the demonstrator are totally equal to the files produced with the CNES software. Therefore, the execution of these tests has been considered as successful and consequently the VHDL implementation has been probed to be in accordance with the requirements.

10. CONCLUSIONS

The implementation of the cloud detection and processing algorithms, together with the CCSDS 123.0-B-2 compression modules (predictor and encoder) has been done into a Xilinx KU040 FPGA. The verification has been performed in comparison with the models.

Two different kind of tests have been performed and some clear success criteria have been defined and tested. All the criteria have been met with two remarks:

- For cloud mask generated by the SVM algorithm a threshold adjustment improves drastically the results. This will need further analysis in posteriors phases of the CHIME detector implementation in order to assure that this adjustment is feasible and compatible with all the detector's images.
- Predictor implementation by Xilinx HLS tool is not optimal and does not meet the data rate. Therefore, an ad hoc VHDL RTL implementation shall be required.

Taking into account these two remarks the implementation in the CHIME demonstrator and the validation can be considered as successful.

The phase A/B1 pre-developments have been planned as de-risking activities for the final CHIME development. The work done in this pre-development and explained in the present paper has demonstrated that the cloud algorithms and the compression implementation will be feasible for the real detector and compliant with the required performances.

The results obtained allow to have an estimation of the FPGA resources needed for the implementation. This information will be used to select the flight FPGA that better fits with the design, which in turn is very relevant for the hardware architecture definition of the processing unit of the CHIME detector.

Acknowledgments: The research leading to these results has been financed by ESA in the frame of the CHIME phase A/B1 pre-developments. The authors want to thank Mickael Bruno and Mathieu Albinet from CNES for the compression software. Also thanks to Roberto Camarero and Raffaele Vitulli from ESA and Michel-François Foulon and Dimitri Lebedeff from Thales Alenia Space in France for their contributions

[1] The Consultative Committee for Space Data Systems. *Low-Complexity Lossless and Near-Lossless Multispectral and Hyperspectral Image Compression, CCSDS 123.0-B-2*, Blue Book ed.; CCSDS, 2019.

[2] The Consultative Committee for Space Data Systems., *Lossless Multispectral and Hyperspectral Image Compression, CCSDS 123.0-B-1*, Blue Book ed.; CCSDS, 2012.

[3] The Consultative Committee for Space Data Systems, *Lossless Data Compression, CCSDS121.0-B-2*, Blue Book ed.; CCSDS, 2012.

[4] ESA ,"*SHyLoC IP core*," https://www.esa.int/Enabling_Support/Space_Engineering_Technology/Microelectronics/SHyLoC_IP_Core, accessed:2020-09-07

[5] OPBDC2020 "On-board Cloud compression: CCSDS standard and proposed evolutions for hyperspectral missions", Dimitri Lebedeff, Michel François Foulon, Roberto Camarero, Raffaele Vitully and Yves Bobichon.

[6] Software implementing CCSDS-121.0-B-3 and CCSDS-123.0-B-2 standards, project DAJ/AR/TN-2017-14924 "Low-Complexity Near-Lossless Multispectral & Hyperspectral Data Compression", Universitat Autònoma de Barcelona (UAB) and Centre National d'Études Spatiales (CNES).