UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA
Instituto Universitario de Microelectrónica Aplicada

IUMA

Review and comparison of design methodologies and hardware implementations on FPGA technologies. Case study: CCSDS compression algorithms for multispectral and hyperspectral images

Yúbal Barrios, Antonio Sánchez
Lucana Santos, Sebastián López, Roberto Sarmiento

10th April 2018

Outline

Introduction and background

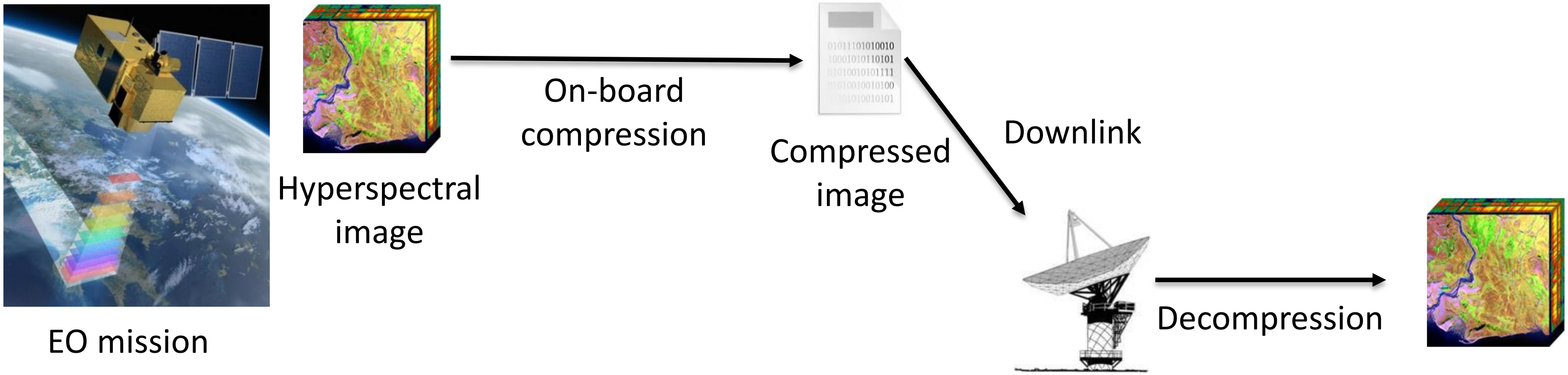
CCSDS algorithms description

Design methodologies

Implementation results

Conclusions

Why do we need on-board compression?



EO mission

Hyperspectral image

On-board compression

Compressed image

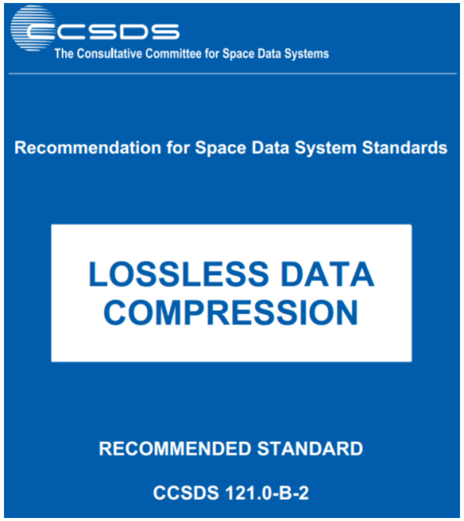
Downlink

Decompression

- While the resolution of the remote sensors, and consequently the data rates continue to increase, the available downlink bandwidth is comparatively stable.
- **Solution** → to apply compression **on-board the satellites.**
- **Lossless compression** allows for reducing the data volume without compromising the data integrity.
- **Lossy compression** yields higher compression ratios introducing losses in the data.

Standard Algorithms of the CCSDS

- CCSDS image compression algorithms (Consultative Committee for Space Data Systems)



CCSDS
The Consultative Committee for Space Data Systems

Recommendation for Space Data System Standards

LOSSLESS DATA COMPRESSION

RECOMMENDED STANDARD

CCSDS 121.0-B-2

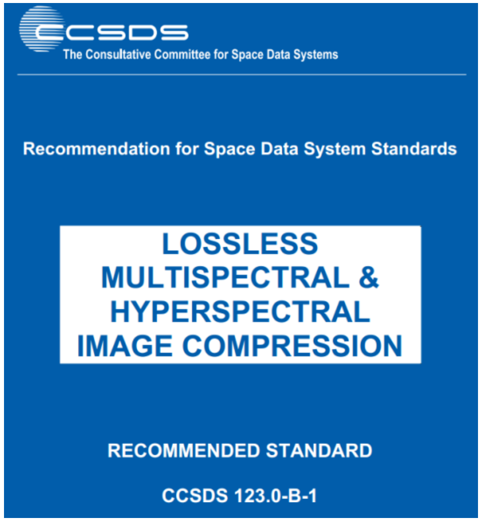
CCSDS
The Consultative Committee for Space Data Systems

Recommendation for Space Data System Standards

IMAGE DATA COMPRESSION

RECOMMENDED STANDARD

CCSDS 122.0-B-2



CCSDS
The Consultative Committee for Space Data Systems

Recommendation for Space Data System Standards

LOSSLESS MULTISPECTRAL & HYPERSPECTRAL IMAGE COMPRESSION

RECOMMENDED STANDARD

CCSDS 123.0-B-1

CCSDS 121

- Universal lossless based on Rice codes.

CCSDS 122

- Lossless or lossy 2D compressor based on DWT.

CCSDS 123

- Multi/hyperspectral compressor based on prediction.

Standard Algorithms of the CCSDS

- CCSDS image compression algorithms (Consultative Committee for Space Data Systems)

CCSDS
The Consultative Committee for Space Data Systems

Recommendation for Space Data System Standards

LOSSLESS DATA COMPRESSION

RECOMMENDED STANDARD

CCSDS 121.0-B-2

CCSDS 121

- Universal lossless based on Rice codes.

CCSDS
The Consultative Committee for Space Data Systems

Recommendation for Space Data System Standards

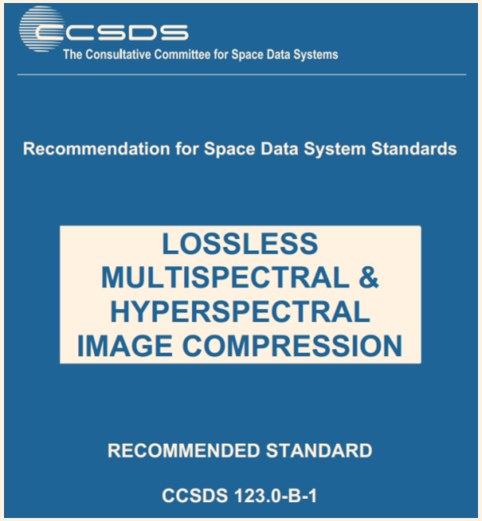
IMAGE DATA COMPRESSION

RECOMMENDED STANDARD

CCSDS 122.0-B-2

CCSDS 122

- Lossless or lossy 2D compressor based on DWT



CCSDS
The Consultative Committee for Space Data Systems

Recommendation for Space Data System Standards

LOSSLESS MULTISPECTRAL & HYPERSPECTRAL IMAGE COMPRESSION

RECOMMENDED STANDARD

CCSDS 123.0-B-1

CCSDS 123

- Multi/hyperspectral compressor based on prediction.

Compression IP cores developed

- Lossless compression
  - Data compression: CCSDS121.
  - Multispectral & Hyperspectral compression: CCSDS123
    - Block coder (Golomb).
    - Rice coder (CCSDS123).
  - Part of **ESA's IP core's Repository** and CoBham Gaisler IP library.
- Lossy compression
  - CCSDS123 lossy extension.
  - HyperLCA: IUMA lossy algorithm (including multispectral & Hyperspectral fusion).

Ongoing Projects

- ESA TRP: extension of the SHyLoC IP Cores (CCSDS121 and CCSDS123)
  - CCSDS 121 and CCSDS 1123 lossless compression IP cores.
  - On-going development:
    - Complete CCSDS121 IP to be able to compress independently.
    - New memory architectures for the CCSDS123 IP improving the throughput.
    - Compatible with SRAM-based FPGAs (Xilinx Virtex V, NanoXplore NG-MEDIUM).
- ENABLE-S3: Reconfigurable Video Processor for Space
  - Consortium: **GMV**, ITI, **TAS-E**, ULPGC, UPM.
  - Fault-tolerant and reconfigurable lossy compression over Xilinx Zynq UltraScale+.
- REBECCA:
  - HyperLCA compression using OpenCL heterogeneous computing over Altera Stratix and nVidia GPUs.

Outline

Introduction and background

CCSDS algorithms description

Design methodologies

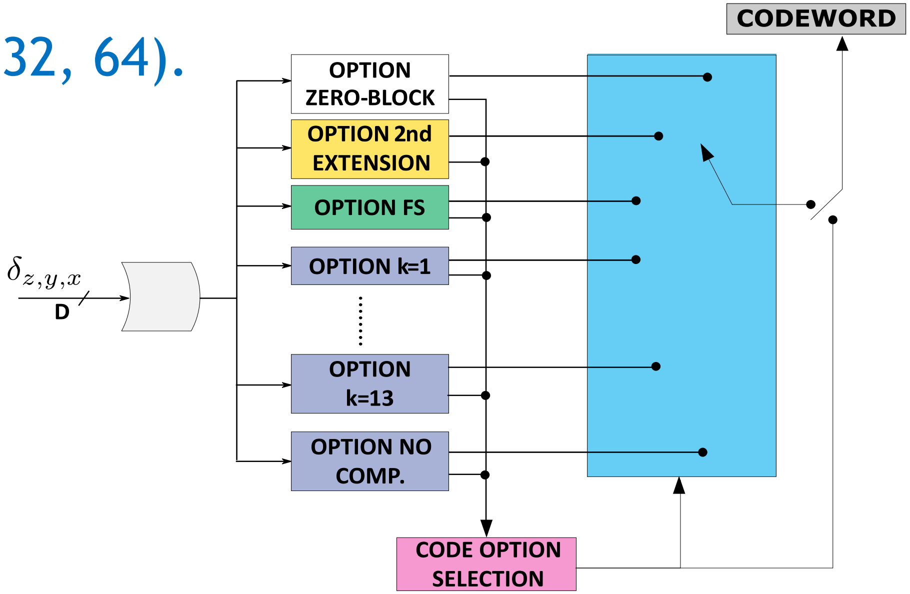
Implementation results

Conclusions

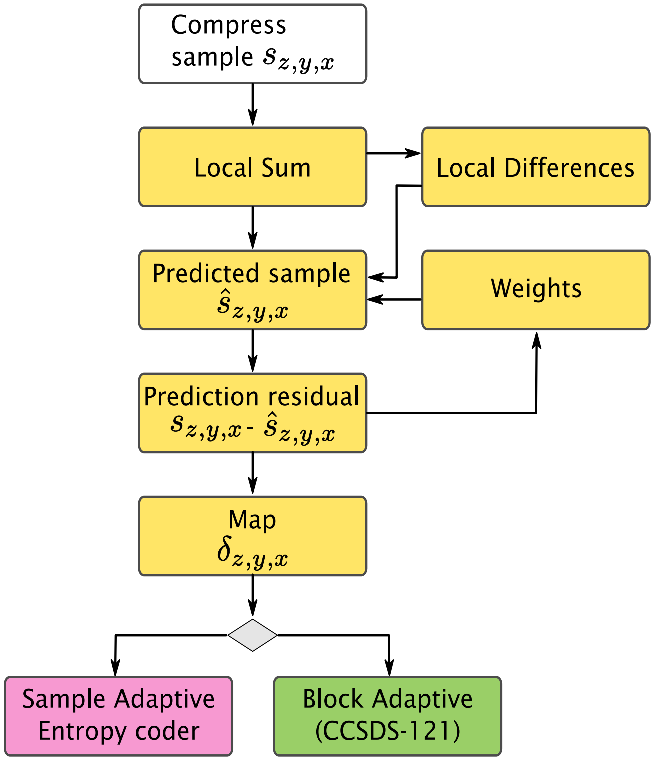
CCSDS121 Standard

- Block-adaptive encoder:
  - A variable-length code that utilizes Rice’s adaptive coding technique.
  - For a block of J samples, the coder evaluates the option that yields the shortest codeword.
  - J is a configurable value (8, 16, 32, 64).
  - Basic code: FS codeword.

Preprocessed Sample Values, δ_i	FS Codeword
0	1
1	01
2	001
.	.
.	.
.	.
.	.
$2^n - 1$	000…0001 ($2^n - 1$ zeros) 1



CCSDS123 Standard



Compress sample $s_{z,y,x}$

Local Sum

Local Differences

Predicted sample $\hat{s}_{z,y,x}$

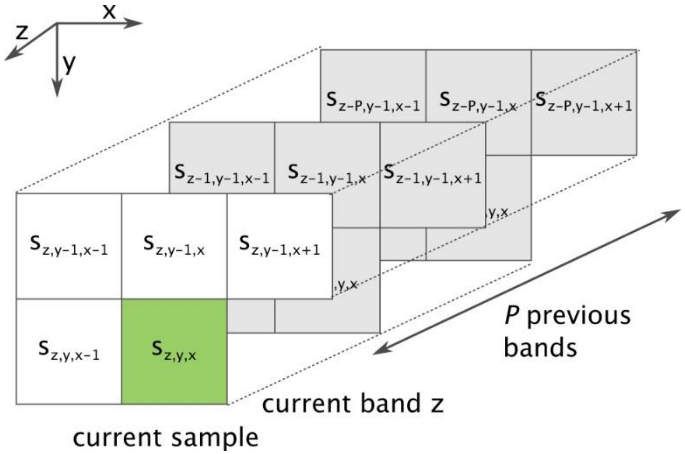
Weights

Prediction residual $s_{z,y,x}$ - $\hat{s}_{z,y,x}$

Map $\delta_{z,y,x}$

Sample Adaptive Entropy coder

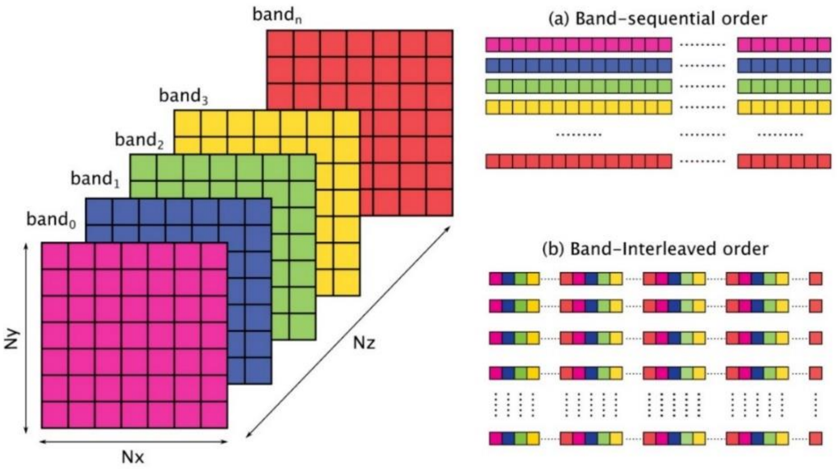
Block Adaptive (CCSDS-121)



- Prediction-based using neighboring samples in the same band and in *P* previous bands (local sum and local differences).
- The prediction is computed from the dot product ($\hat{d}$) between the local differences vector (U) and a weight vector (W) $\hat{d} = W^T{}_{z,y,x} \cdot U_{z,y,x}$
- Prediction residuals are mapped and then encoded using a variable-length binary codeword.
- The variable-length codes are adaptively selected based on statistics that are updated after each sample is encoded.

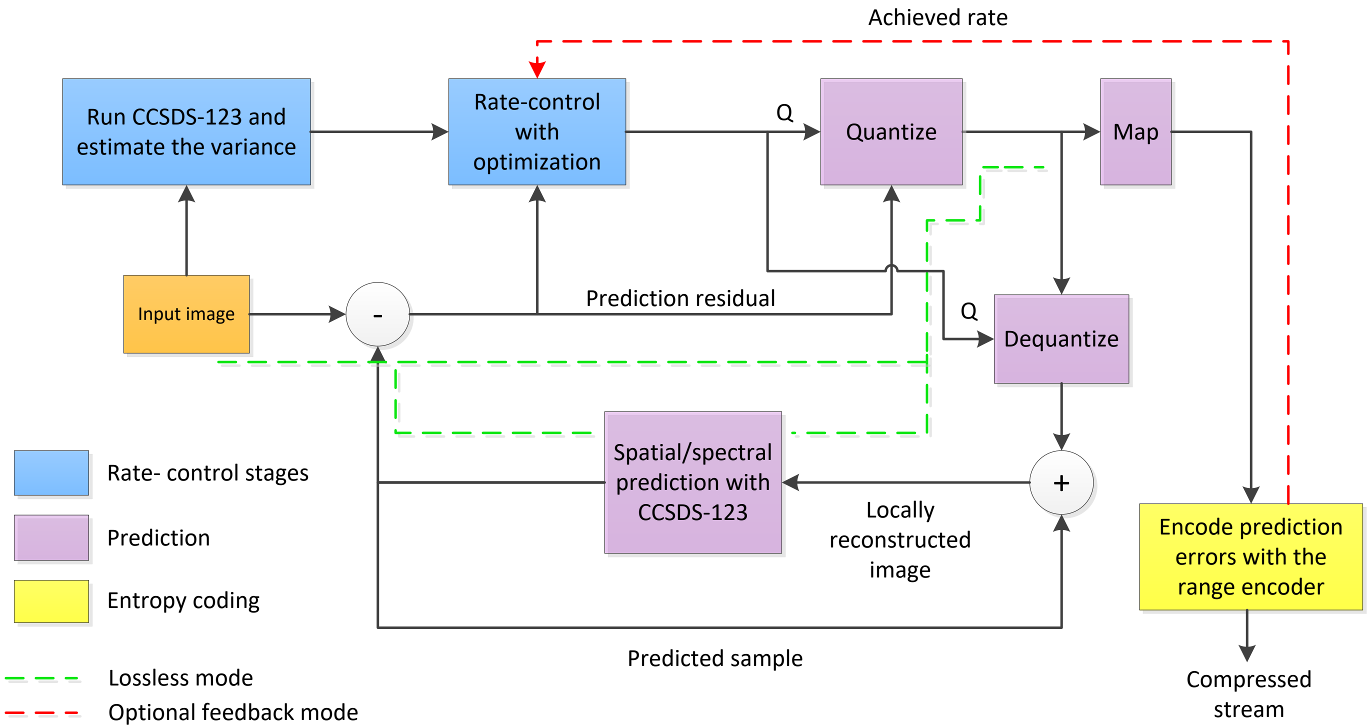
CCSDS123 architectural solutions

- Different architectures, depending on the sensor type: BIP, BSQ, BIL.
- For BIP and BIL orders, two different memory approaches:
  - Mem architecture: uses external memory to store intermediate values for compression → Lower resource utilization.
  - Base architecture: stores the intermediate results only in the FPGA internal memory → Better throughput.
- Different storage requirements depending on the compression order, image size and *P* (number of bands used for prediction).
  - Different achievable throughput:
    - BIP → allows for parallelization of prediction operations of a sample in all bands.
    - BSQ → prediction finished before starting the compression of samples in the same band.
    - BIL → mixed situation.



CCSDS123 Lossy Extension

- Hyperspectral Compression algorithm, works in near-lossless to lossy range*.
- Able to adapt losses according to the user-selected bit rate (rate control).
- Leverages predictor and entropy coder from the CCSDS-123.0 lossless compressor.
- Rate control will be included in the standard as an option.


Achieved rate
Run CCSDS-123 and estimate the variance
Rate-control with optimization
Q
Quantize
Map
Input image
-
Prediction residual
Q
Dequantize
Spatial/spectral prediction with CCSDS-123
+
Locally reconstructed image
Encode prediction errors with the range encoder
Predicted sample
Compressed stream
Rate- control stages
Prediction
Entropy coding
Lossless mode
Optional feedback mode

*D. Valsesia and E. Magli, "A Novel Rate Control Algorithm for Onboard Predictive Coding of Multispectral and Hyperspectral Images," in IEEE Transactions on Geoscience and Remote Sensing, vol. 52, no. 10, pp. 6341-6355, Oct. 2014.

Outline

Introduction and background

CCSDS algorithms description

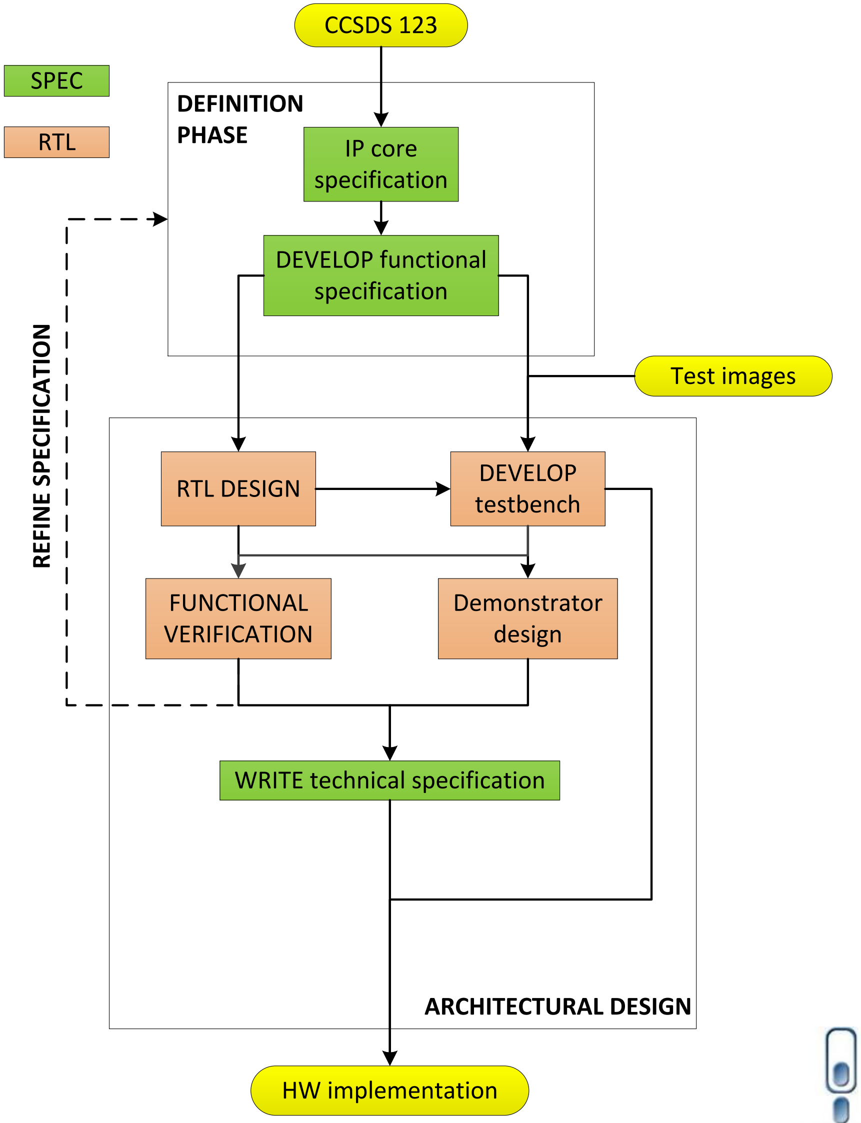
Design methodologies

Implementation results

Conclusions

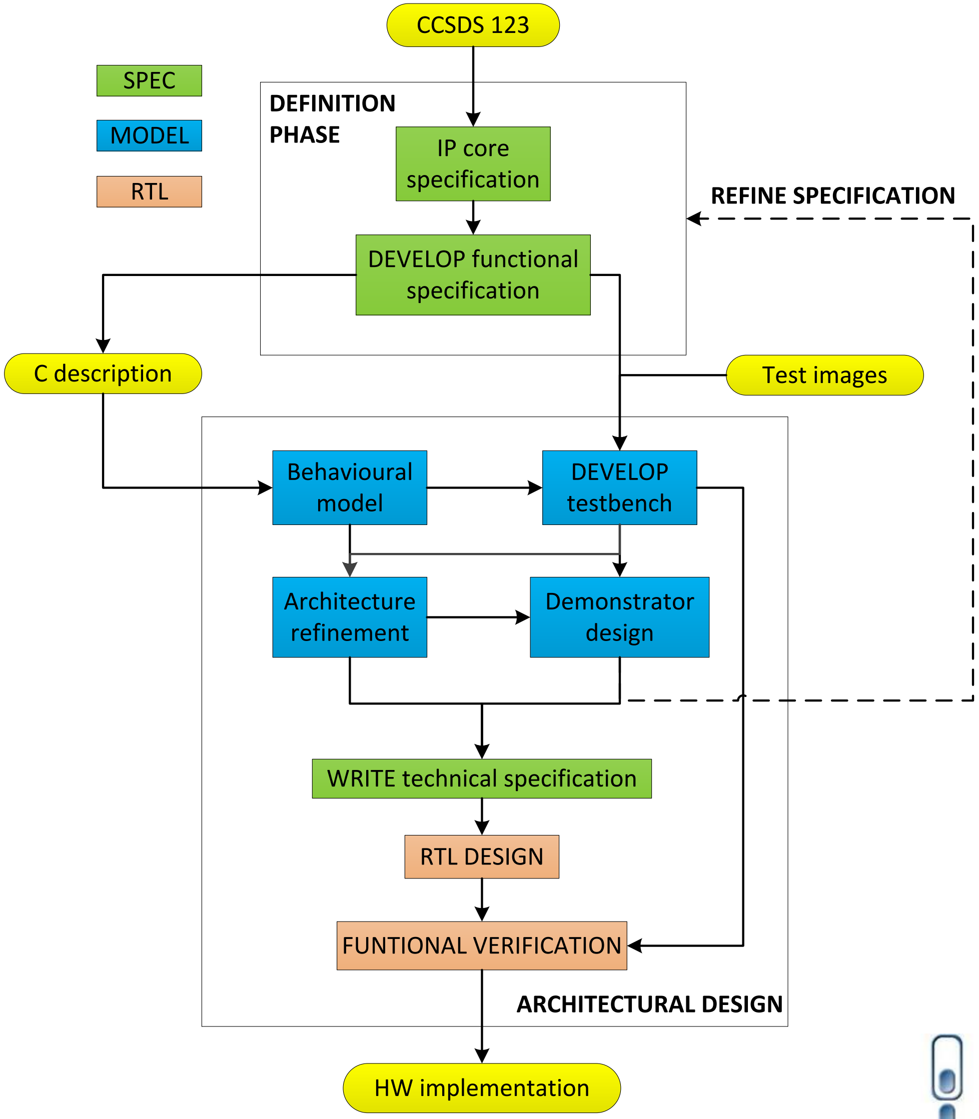
RTL design flow

- All the design and verification steps are performed at RTL level:
  - Full control of all implementation details (cycle-accurate design).
  - Validation and optimization after RTL design.
  - Large design times for complex systems.
  - Costly specification refinement (necessity of partially re-designed).



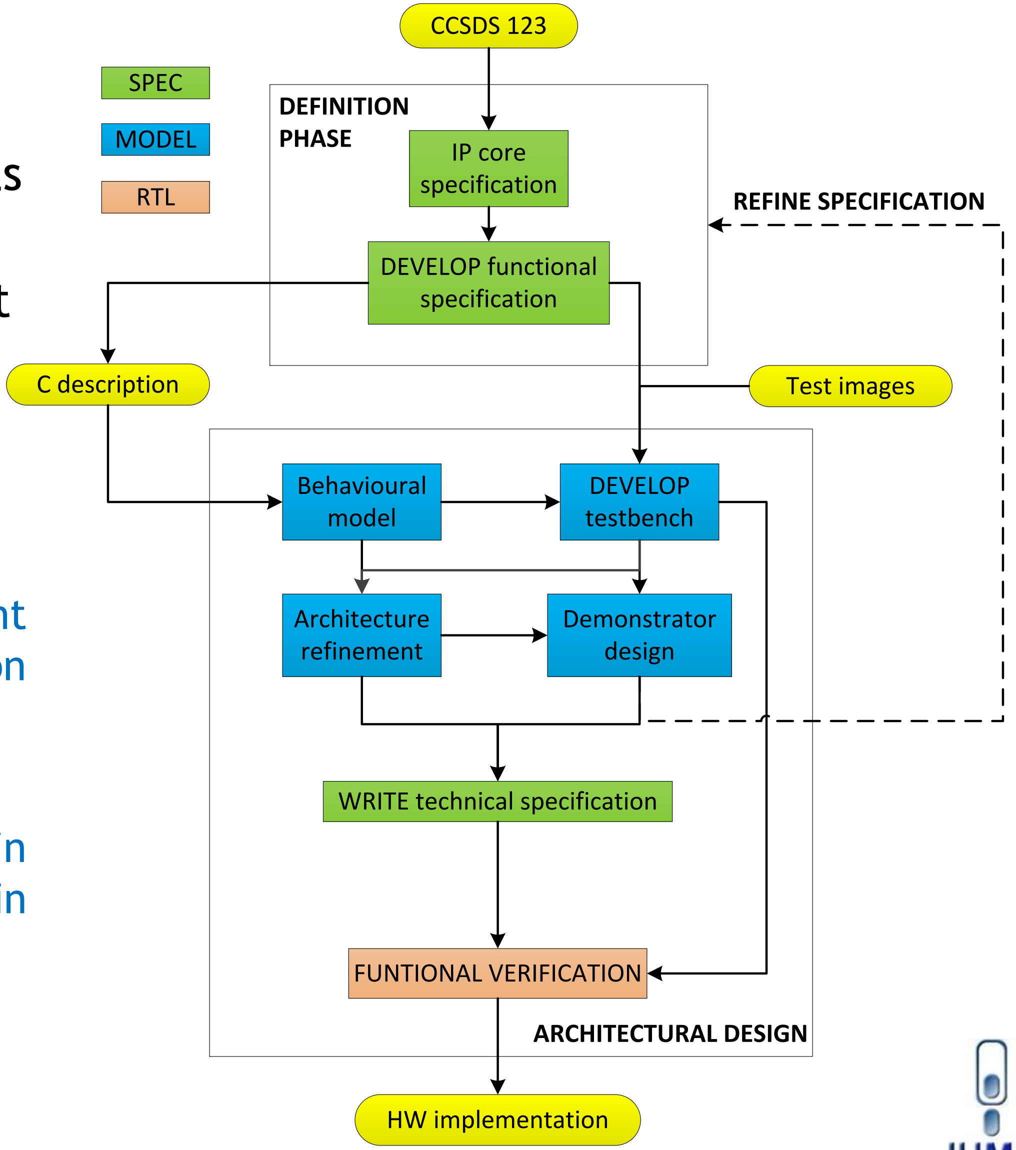
Mixed design flow

- IP cores modelled in a high-level programming language (C/C++) as previous stage of a VHDL implementation.
- Advantages of having a higher abstraction model:
  - Numerous iterations for architecture exploration in a short term.
  - Generate specifications that lead to efficient implementations.
  - Design optimization at an early stage.
  - Validate hardware against software before the RTL description is modelled.
  - HW/SW co-design: HW and SW are not developed in isolation.
- Implementation is still done at RTL level.



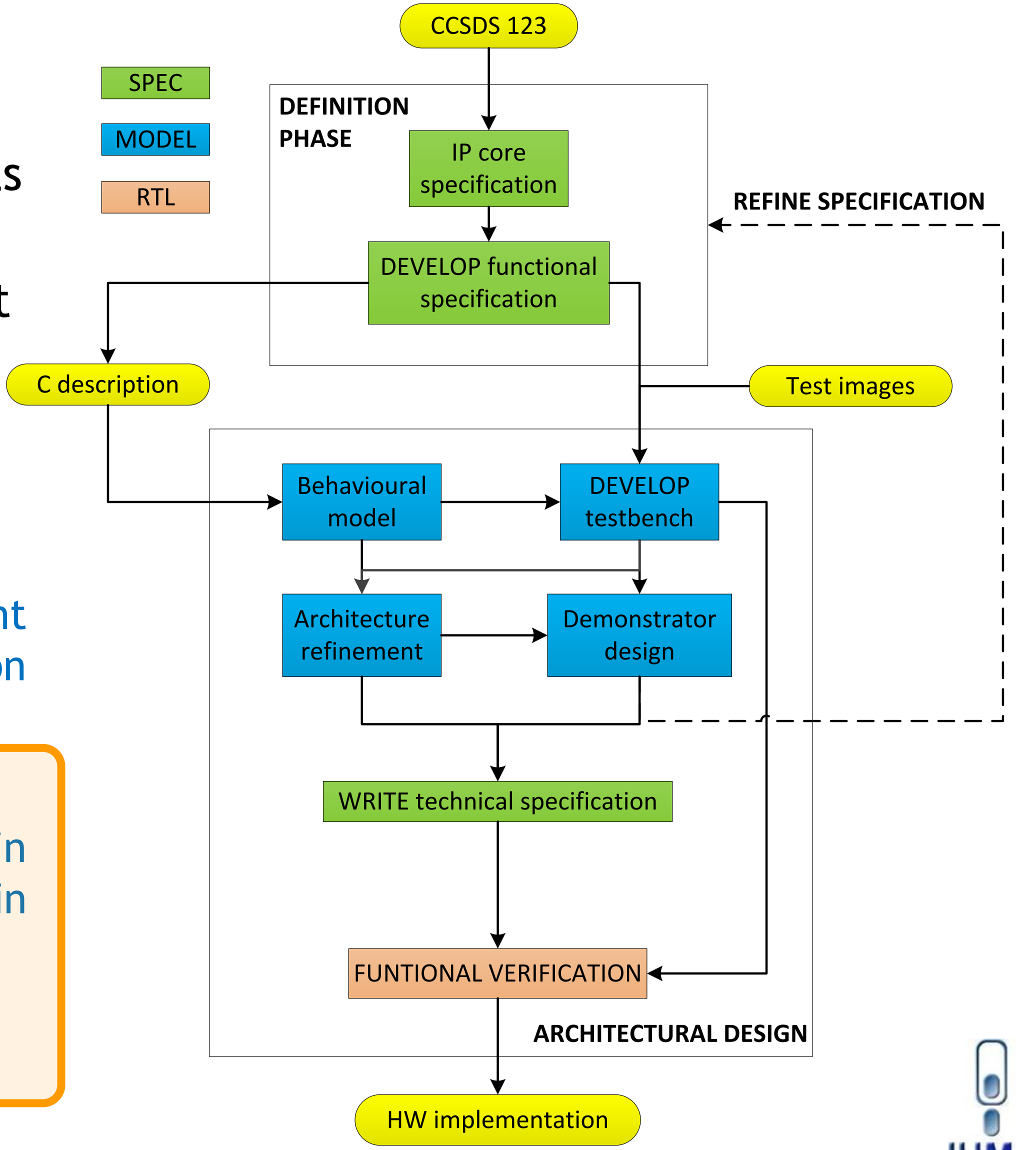
HLS design flow

- IP cores modelled in C directly transformed into RTL.
- Implementations by automated tools (CatapultC, Vivado HLS).
- C codes are adapted for an efficient hardware implementation.
- Advantages of HLS design:
  - Minimal design at RTL level.
  - Reduced Time-to-Market.
  - Fast exploration of different architectures and parallelization approaches.
- Methodologies in this work:
  - CatapultC: CCSDS123 predictor in HLS, entropy coder and interfaces in VHDL.
  - Vivado HLS: full CCSDS123 lossy compressor.



HLS design flow

- IP cores modelled in C directly transformed into RTL.
- Implementations by automated tools (CatapultC, Vivado HLS).
- C codes are adapted for an efficient hardware implementation.
- Advantages of HLS design:
  - Minimal design at RTL level.
  - Reduced Time-to-Market.
  - Fast exploration of different architectures and parallelization approaches.
- Methodologies in this work:
  - CatapultC: CCSDS123 predictor in HLS, entropy coder and interfaces in VHDL.
  - Vivado HLS: full CCSDS123 lossy compressor.



Outline

Introduction and background

CCSDS algorithms description

Design methodologies

Implementation results

Conclusions

CCSDS123 mapping

- Results for Microsemi RTG4.

Resources	BIP	BIP-mem	BSQ	BIL
MACC	7 (2%)	7 (2%)	7 (2%)	7 (2%)
RAM64x18_RT	31 (15%)	33 (16%)	25 (12%)	38 (19%)
RAM1K18_RT	4 (2%)	0 (0%)	1 (1%)	10 (5%)
LUTs	4799 (4%)	5996 (4%)	5973 (4%)	5211 (4%)
Max. Freq. (MHz)	78.3	**78.3**	68.5	75.2

LANDSAT

Resources	BIP	BIP-mem	BSQ	BIL
MACC	13 (3%)	13 (3%)	11 (3%)	13 (3%)
RAM64x18_RT	62 (30%)	64 (31%)	36 (18%)	65 (31%)
RAM1K18_RT	129 (62%)	1 (0%)	1 (0%)	135 (65%)
LUTs	7174 (5%)	7569 (5%)	7123 (5%)	7572 (5%)
Max. Freq. (MHz)	61	69.3	70.1	61.1

AVIRIS

Lossy CCSDS123 compression ratios

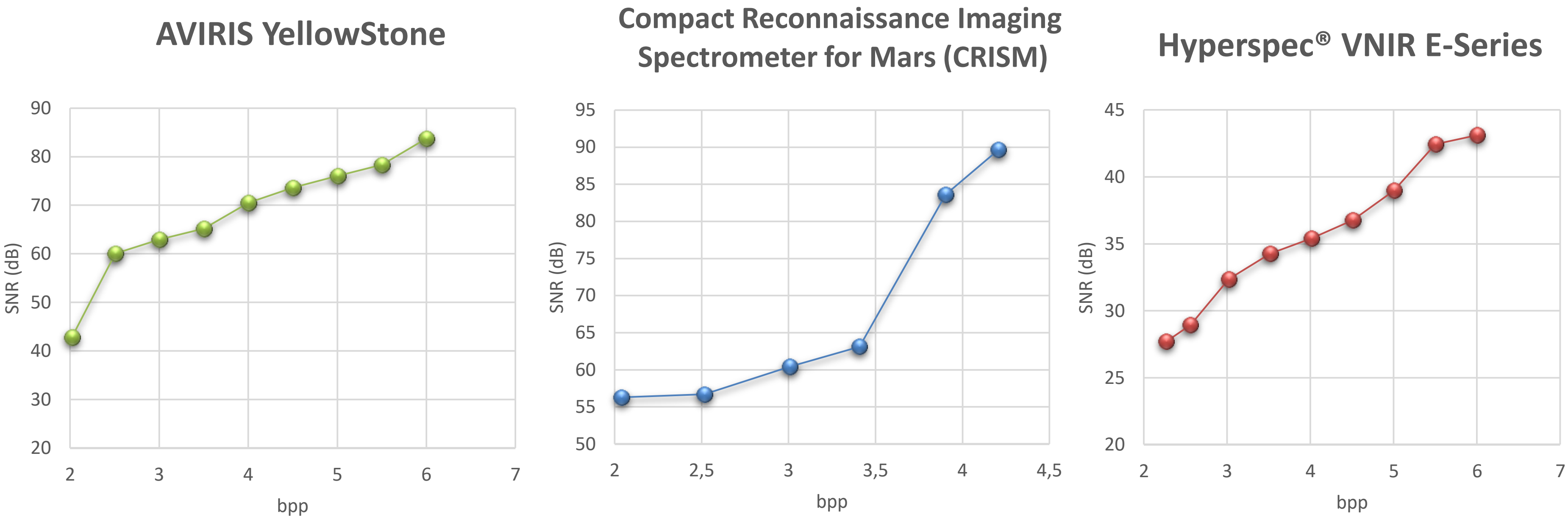


Image	Nx	Ny	Nz	bpp	Signed	Endianness	State
AVIRIS	512	680	224	16	No	BIG	Preprocessed
CRISM	90	135	1501	12	No	LITTLE	Calibrated
HYPERSEC E-SERIES	400	400	300	16	Yes	LITTLE	Raw

Lossy CCSDS123 mapping

- Synthesis performed for Xilinx Zynq XC7Z020.
- Baseline predictor configuration with:
  - BIL order processing.
  - Number of bands for prediction, P = 5.
  - Weight component resolution W = 13.
  - Neighbor oriented and full prediction.
  - Sample-adaptive encoder is always implemented.
  - Bit width of output buffer W_BUFFER = 64.

Implementation	Resources				Max. Frequency (MHz)
	BRAM	DSP48	LUT	FF	
HLS (Catapult C)	28,21%	5%	24,77%	6,34%	72
HLS (Vivado HLS)	25,36%	8%	11,92%	4,7%	80,2
VHDL (SHyLoC)*	50,36%	4%	14,11%	5%	70

*Lossless CCSDS-123 implementation (part of ESA's IP core's Repository).

CCSDS121 mapping

- Synthesis results for Xilinx Virtex-V, Microsemi RTG4 and NanoXplore NG-MEDIUM.
- Baseline encoder configuration with:
  - Block size J = 32.
  - Dynamic range of input samples, D = 16.
  - Bit width of output buffer W_BUFFER = 32.

Device	Resources				Max. Frequency (MHz)
	BRAM	DSP48	LUT	FF	
Virtex XC5VFX130T	3	1	3657	1501	Clk_AHB → 289,2 Clk_S → 118
Microsemi RTG4 150	11	3	5419	1347	Clk_AHB → 121,8 Clk_S → 51,4
NanoXplore NG-MEDIUM*	11	5	9371	1639	Clk_AHB → 79,8 Clk_S → 30,7

*Using 2.8.3 version of NanoXplore NanoXmap

Outline

ULPGC

Introduction and background

CCSDS algorithms description

Design methodologies

Implementation results

Conclusions

IUMA

Conclusions

- We have presented different hardware implementations which perform lossless compression as specified by the CCSDS-121 and a proposed lossy extension for the CCSDS-123 standard.
- Different architectures have been tested for the CCSDS-123 standard, depending on the compression order and the selected parameters.
- HLS and RTL design methodologies have been applied, enabling a comparison between them.
- Mapped to different FPGA devices: the proposed lossy CCSDS-123 algorithm over Xilinx Zynq 7000-Series; the CCSDS-121 standard over NanoXplore NG-MEDIUM, Microsemi RTG4 and Xilinx Virtex-V.
- Feasibility of the lossy implementation on Zynq XC7Z020 (maximum throughput 80 Msamples/s) and low complexity maximum 25% of LUTs and 28% of BRAMs.
- First approximation to the use of the first medium-capacity, high-performance, radiation-hardened re-programmable European FPGA (NG-MEDIUM).

Questions?

Yúbal Barrios

E-mail: ybarrios@iuma.ulpgc.es

Antonio Sánchez

E-mail: ajsanchez@iuma.ulpgc.es