



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



EXPRO+ ESA AO/1-8032/14/NL/AK

CCSDS Lossless Compression IP-CORE Space Applications

Final Presentation Days

8th May 2017

Introduction and motivation

CCSDS lossless compressors overview

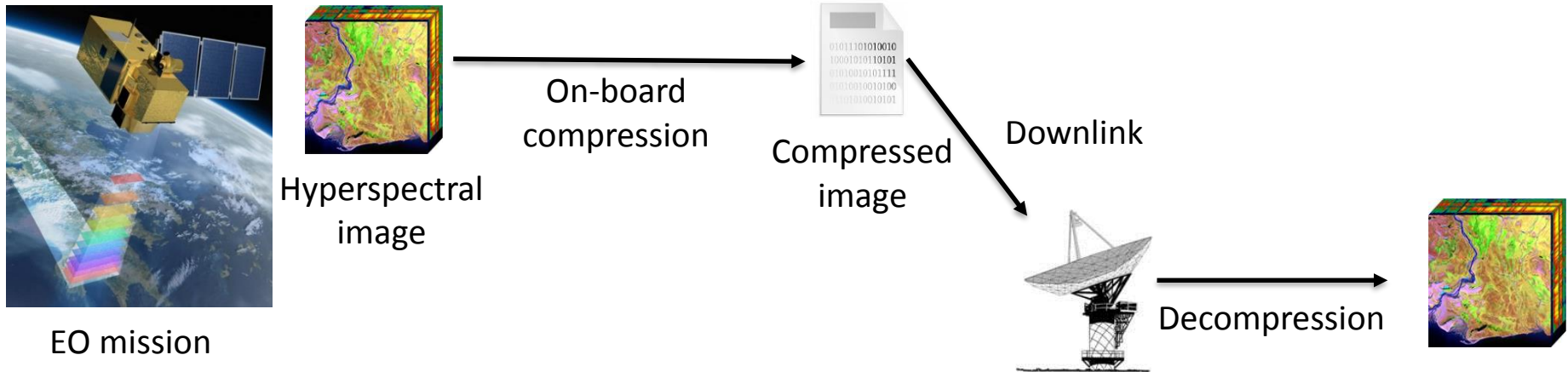
SHyLoC SystemC Model

SHyLoC VHDL Description and Verification

Technology Mapping to FPGA and ASIC

Conclusions

Why do we need on-board compression?



- ▶ While the resolution of the remote sensors, and consequently the data rates continue to increase, the available downlink bandwidth is comparatively stable.
- ▶ The solution offered is to apply compression **on-board the satellites** → payload data processors have to be able to accomplish this task.
- ▶ **Lossless compression** allows for reducing the data volume without compromising the data integrity (the image can be fully recovered after decompression).

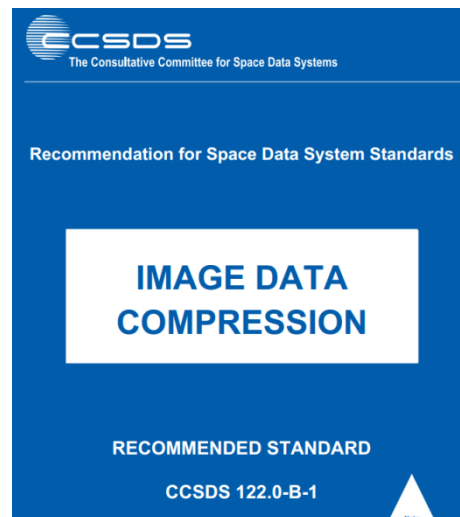
Standard Algorithms of the CCSDS

- ▶ CCSDS algorithms (Consultative Committee for Space Data Systems)



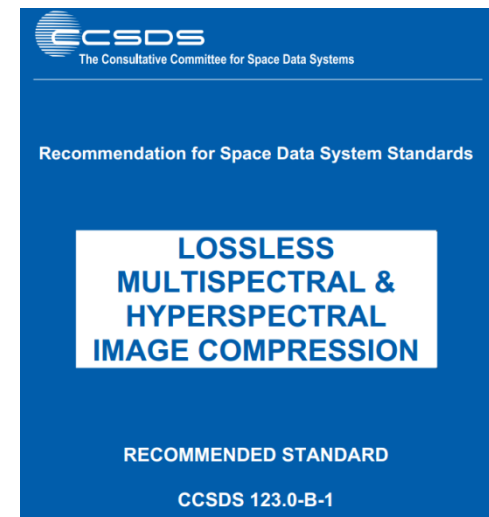
CCSDS 121

- ▶ Universal lossless based on Rice codes.



CCSDS 122

- ▶ Lossless or lossy 2D compressor based on DWT



CCSDS 123

- ▶ Multi/hyperspectral compressor based on prediction.

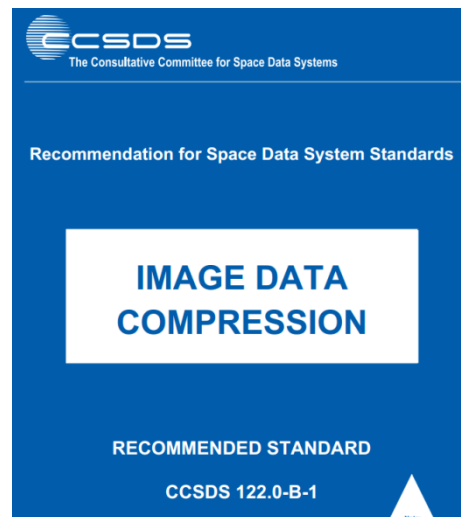
Standard Algorithms of the CCSDS

- ▶ CCSDS algorithms (Consultative Committee for Space Data Systems)



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.

The challenge

- ▶ Efficient hyperspectral image compression on-board the available hardware (maximum reduction of data with minimum requirements of on-board resources)

The goal

- ▶ Develop low-complexity high-throughput hardware architectures.
- ▶ Efficient implementation on space-qualified FPGAs and ASICs.

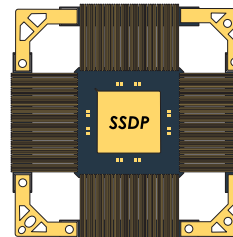
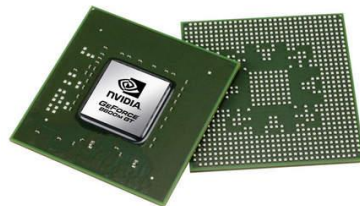
▶ Software

- CPU
- DSP
- GPU

(not space-qualified)

▶ Hardware

- ASIC
- FPGA





TRP activity
CCSDS Lossless Compression IP-CORE Space Applications (SHyLoC)



▶ Main objectives

- Model and implement of **two lossless compression IP cores**.
- **Described** in SystemC and VHDL.
- Compliant with the **CCSDS 121** and **CCSDS123** standards, including all configuration modes.
- To be part of **ESA's IP core's Repository**.
- Compatible with technologies:
 - One-time programmable FPGAs (Microsemi);
 - Reconfigurable FPGAs (Virtex5);
 - ASIC (DARE standard cell library)

Summarized Workplan



WP1	IP Core Definition (IUMA)
WP2	System IP Core Implementation and Design Space Exploration (IUMA)
WP3	VHDL IP Core Implementation (IUMA)
WP4	IP Core validation and deployment (TELETEL)
WP5	Technology mapping (TELETEL & TASE)

- ▶ The consortium for this Project is formed by:
 - The Institute for Applied Microelectronics (IUMA) from the University of Las Palmas de Gran Canaria (ULPGC), Spain.
 - TELETEL SA, Greece.
 - Thales Alenia Space Spain (TASE)



Introduction and motivation

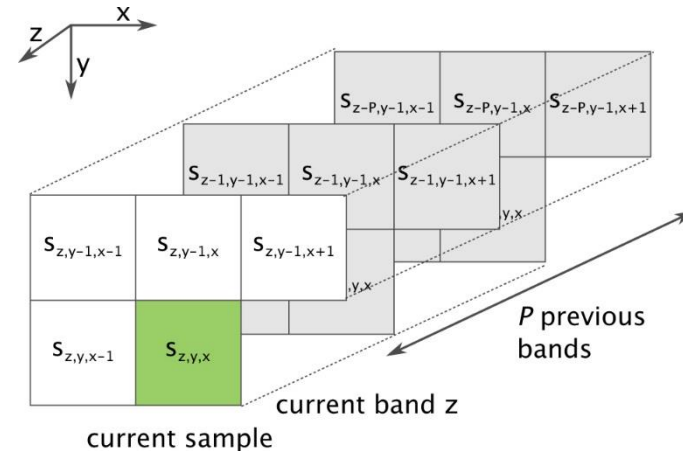
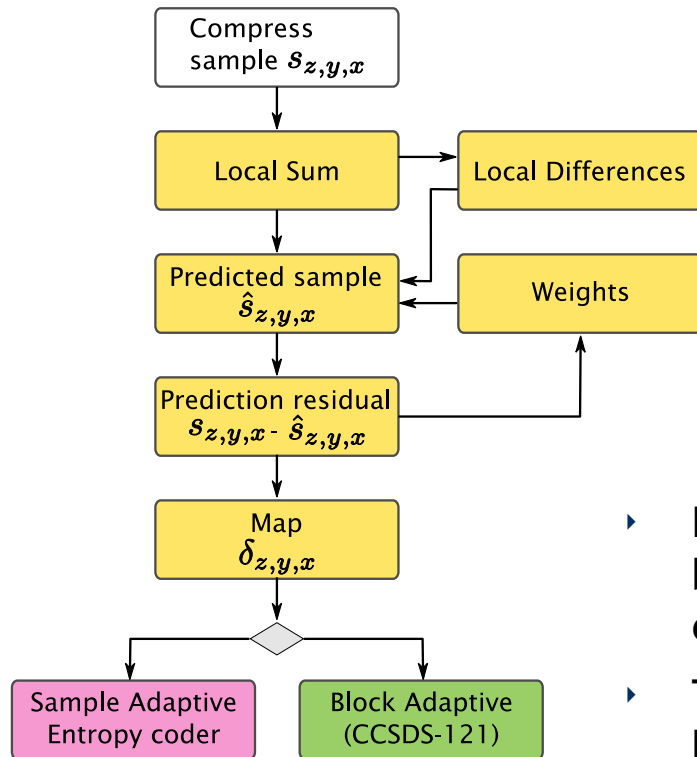
CCSDS lossless compressors overview

SHyLoC SystemC Model

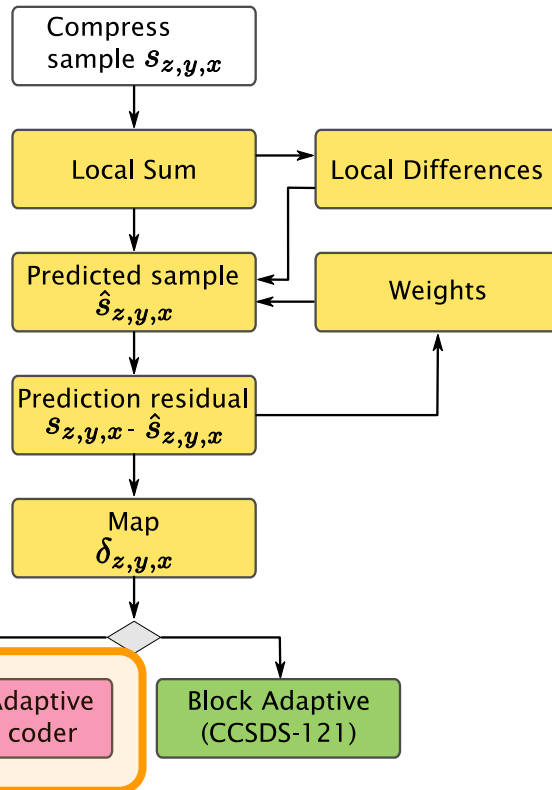
SHyLoC VHDL Description and Verification

Technology Mapping to FPGA and ASIC

Conclusions



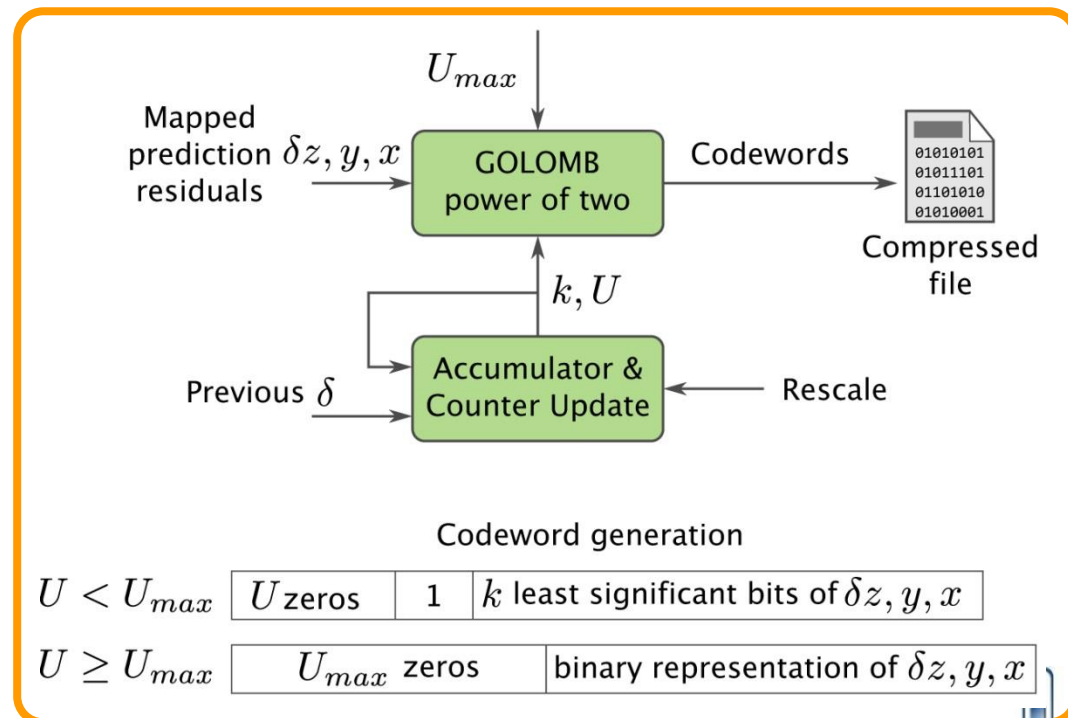
- ▶ Prediction-based using neighboring samples in the same band and in P previous bands (local sum and local differences).
- ▶ The prediction is computed from the dot product (\hat{d}) between the local differences vector (U) and a weight vector (W) $\hat{d} = W_{z,y,x}^T \cdot U_{z,y,x}$
- ▶ The weight vector is updated according to the prediction error.
- ▶ Prediction residuals are mapped and then encoded.

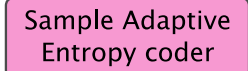


Entropy coding:

- Sample-adaptive
- Block-adaptive (CCSDS-121)

Sample-adaptive entropy coding





- Sample-adaptive

- Block-adaptive (CCSDS-121)

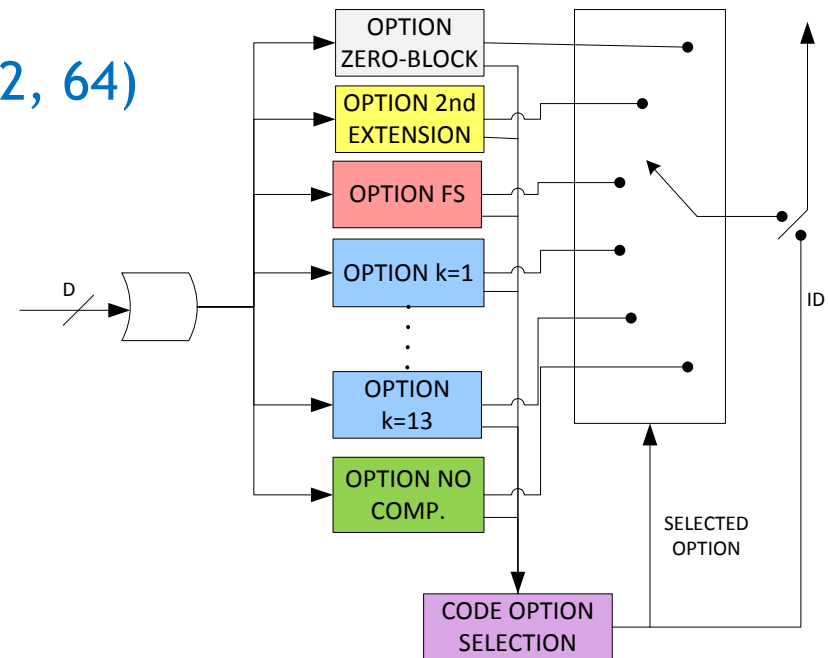
Block-adaptive entropy coding



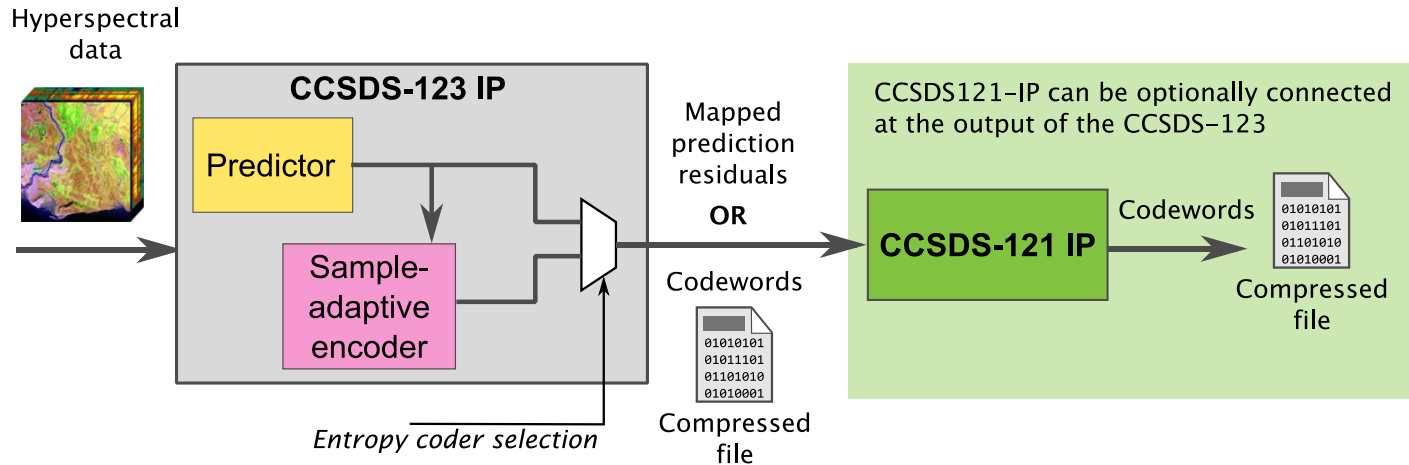
► Block-adaptive encoder:

- 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
	$(2^n - 1 \text{ zeros}) 1$



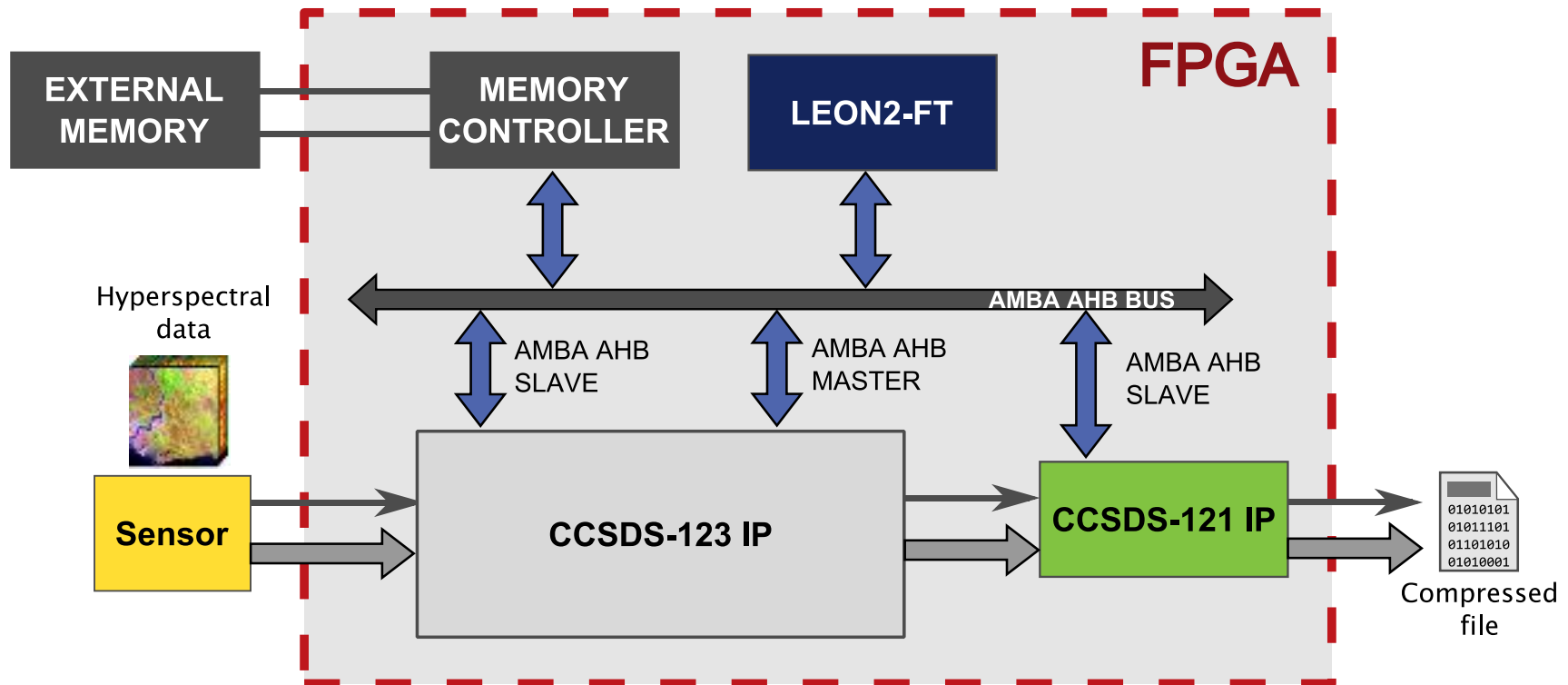
SHyLoC Concept - Requirements



- ▶ The IP cores can be combined into a logical single entity.
- ▶ CCSDS-121 IP can work independently as well as block-adaptive coder of the CCSDS-123 IP.
- ▶ Compression parameters selectable at runtime.
- ▶ Capable of accepting samples in any of the possible arrangements: band-sequential (BSQ), band-interleaved per pixel (BIP) or band-interleaved per line (BIL).

SHyLoC Concept - Requirements

- ▶ Throughput up to 1 Gbps (~60 Msamples/s for 16-bit input) when implemented on a Virtex5 FX130.
- ▶ Include AMBA AHB interfaces.
- ▶ Compatible with GRLIB and LEON2-FT.



Introduction and motivation

CCSDS lossless compressors overview

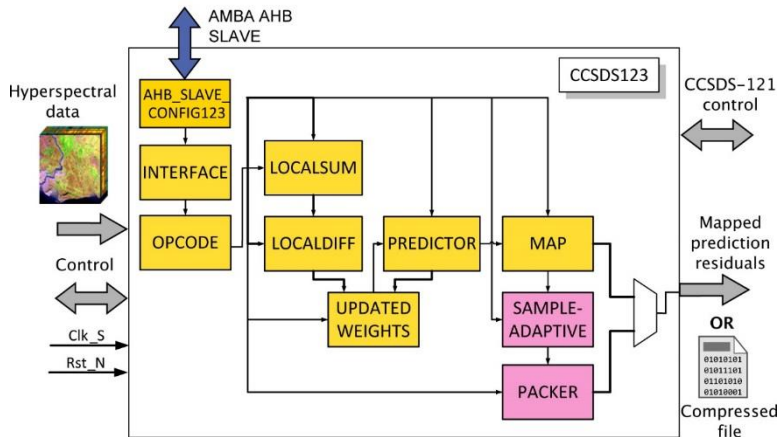
SHyLoC SystemC Model

SHyLoC VHDL Description and Verification

Technology Mapping to FPGA and ASIC

Conclusions

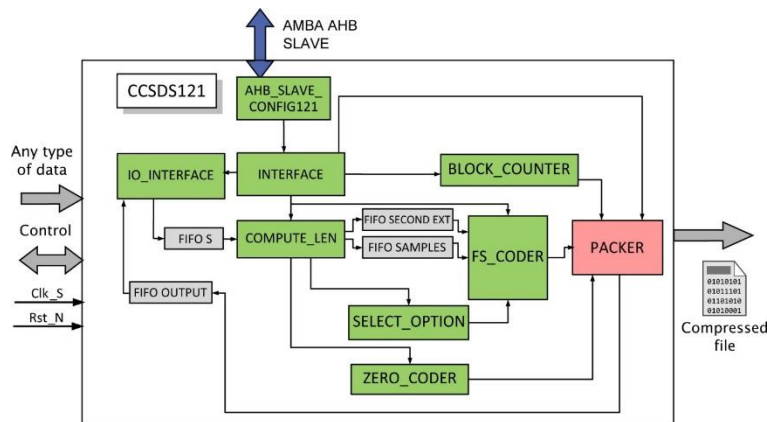
CCSDS123 SystemC Model



- Models have the same interfaces and behaviour as the VHDL counterpart:

- Compatible I/O interfaces enable plug and play connectivity between the IP cores/SystemC models.
- AHB interfaces (TLM).
- Configuration at compile-time and run-time.

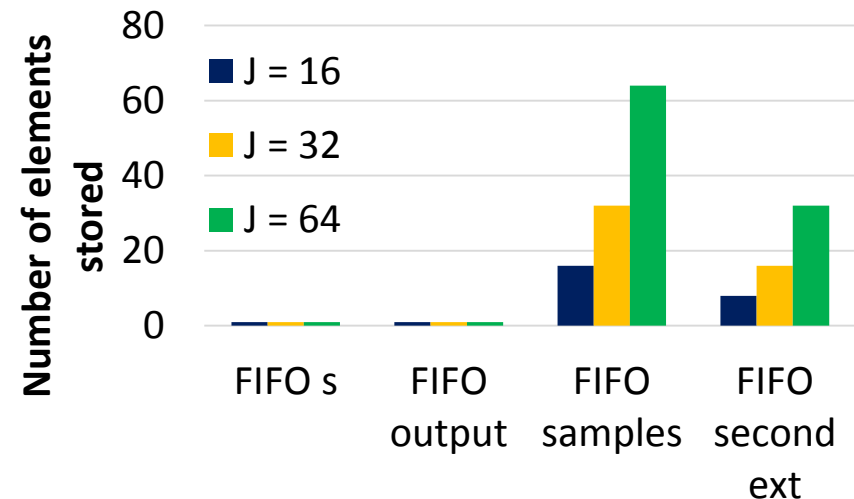
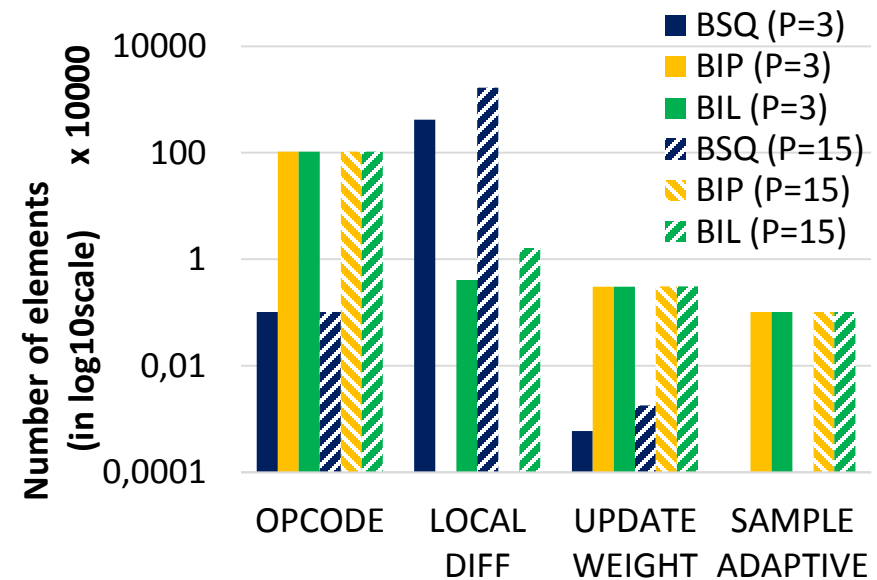
CCSDS121 SystemC Model



- Exploration allowed to identify:
 - Relationships between configuration parameters and hardware complexity.
 - Data dependencies limiting throughput.
 - Potential need of storage element external to the FPGA.

SystemC Modelling findings

- Findings of the CCSDS-123 models:
 - Complexity depends on image size, P and compression order.
 - Need for different architectures for each compression order.
 - Optimize design for baseline P value ($P = 3$).
 - High throughput in BIP; lower in BSQ and BIL.
 - Need for external memory.
- Findings of the CCSDS-121 model
 - Influence of J in storage and latency.
 - High throughput.



Introduction and motivation

CCSDS lossless compressors overview

SHyLoC SystemC Model

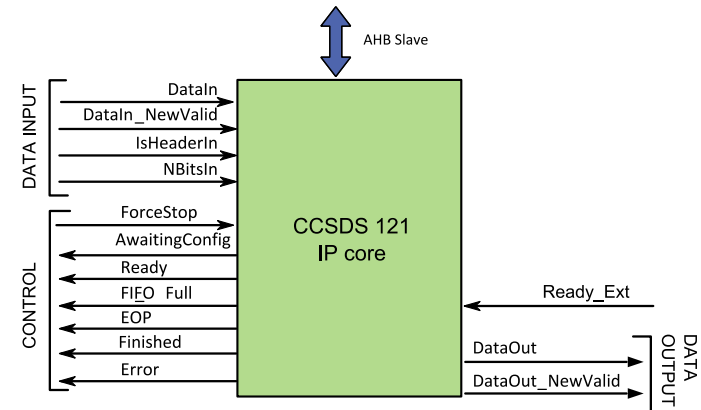
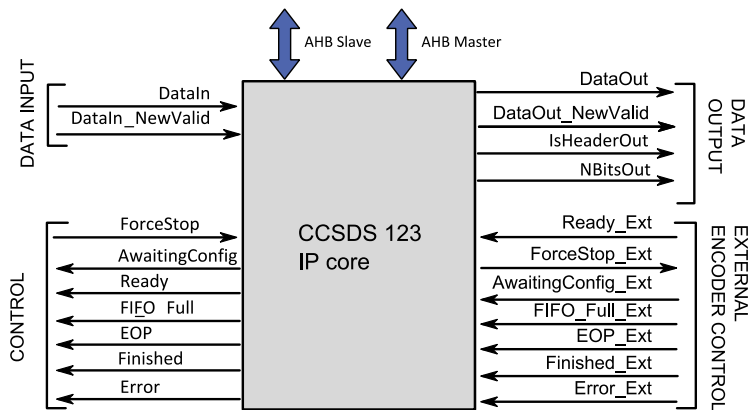
SHyLoC VHDL Description and Verification

Technology Mapping to FPGA and ASIC

Conclusions

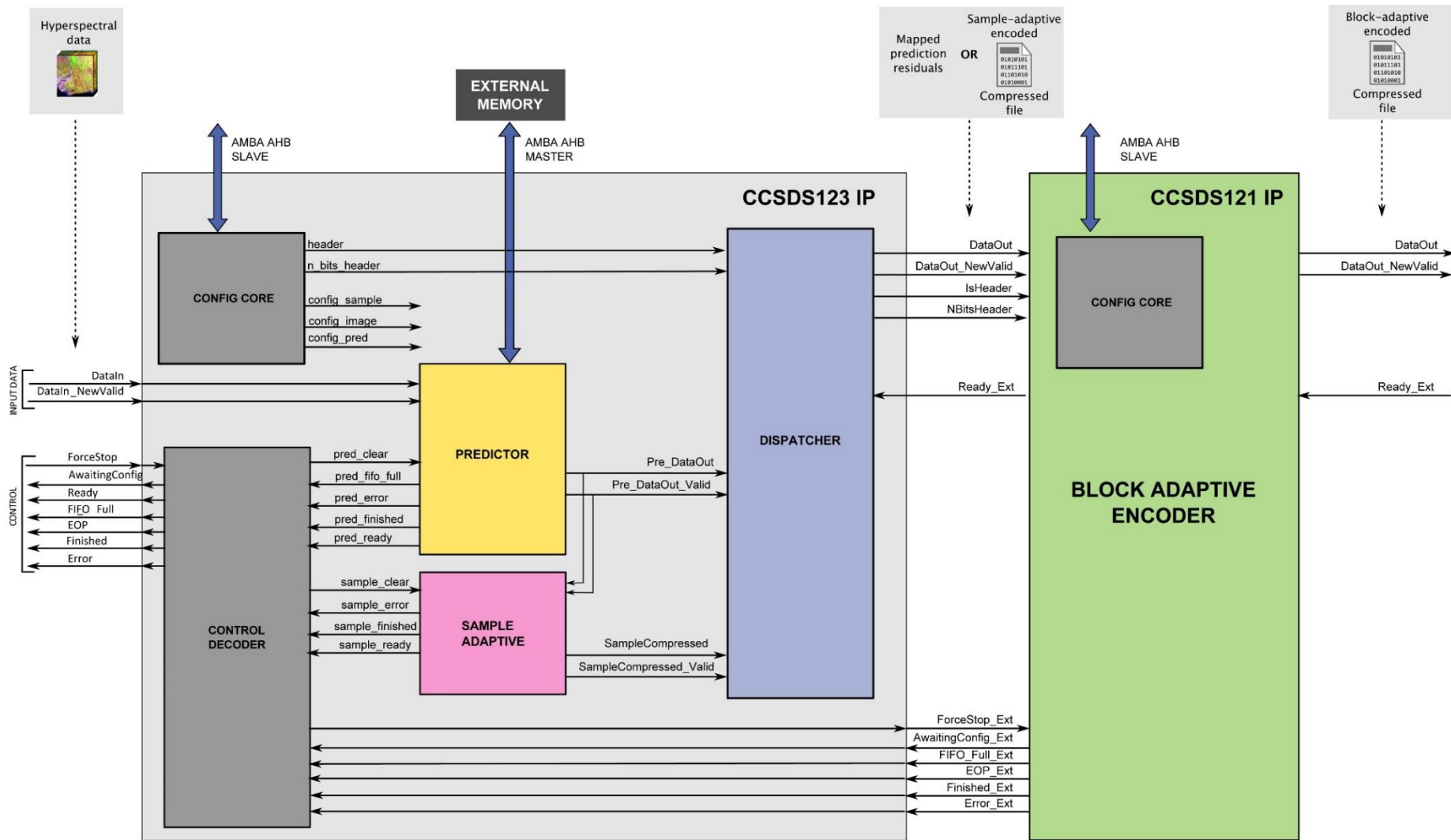
SHyLoC VHDL description

- ▶ Perform lossless on-board data compression according to the CCSDS 121 and CCSDS 123 standard algorithms.
- ▶ Separate VHDL IP cores that can work independently, or be connected together.



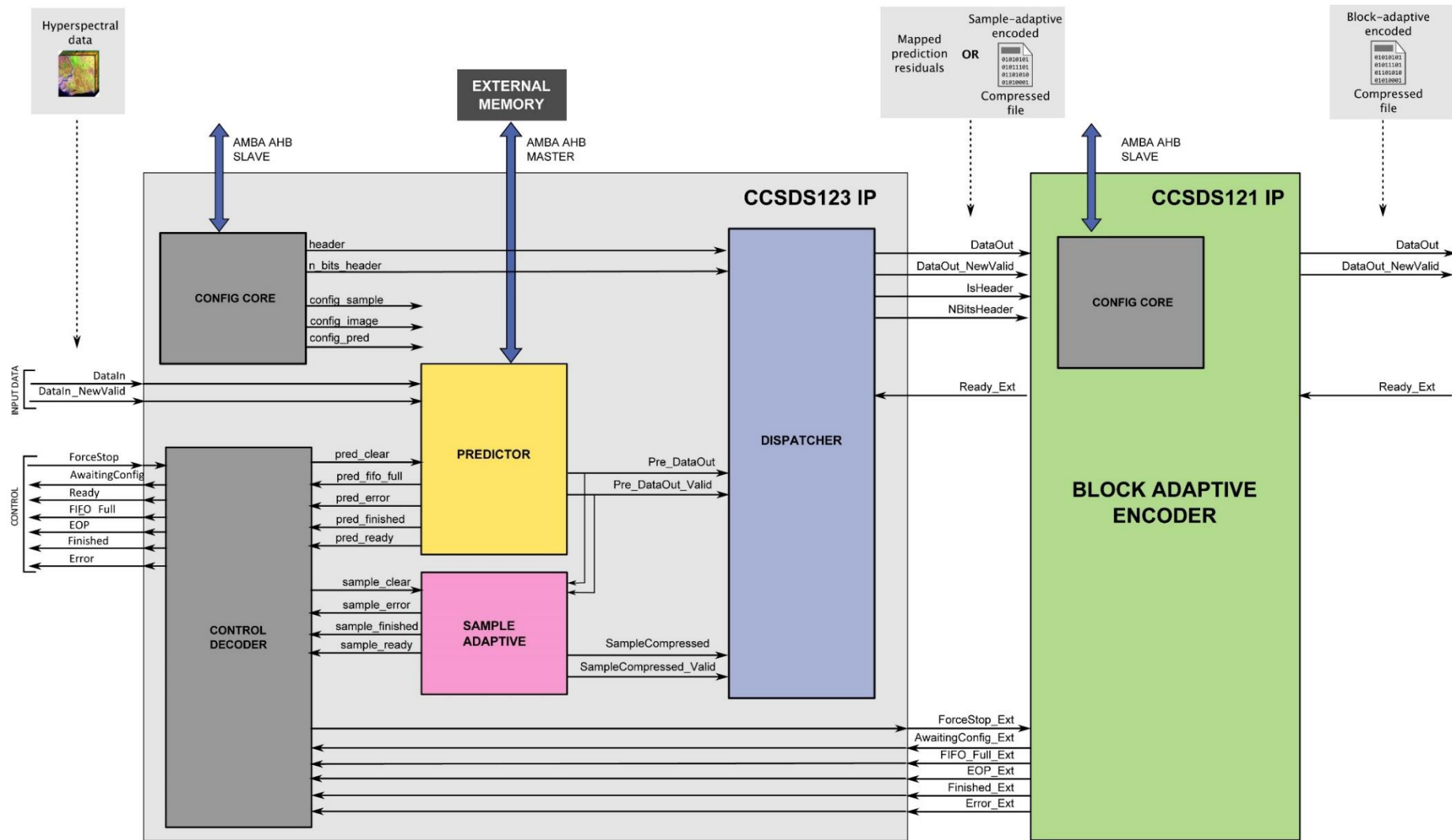
- ▶ CCSDS 123 IP core:
 - High-performance lossless compression of multispectral and hyperspectral data.
 - Supports BSQ, BIP and BIL sample order.
 - Can be used as external pre-processor (predictor) for the CCSDS 121 IP core.
- ▶ CCSDS 121 IP core:
 - Universal lossless compressor based on Rice's coding.
 - Can be used as external entropy coder for the CCSDS 123 IP.

CCSDS123 and CCSDS121 Interfaces

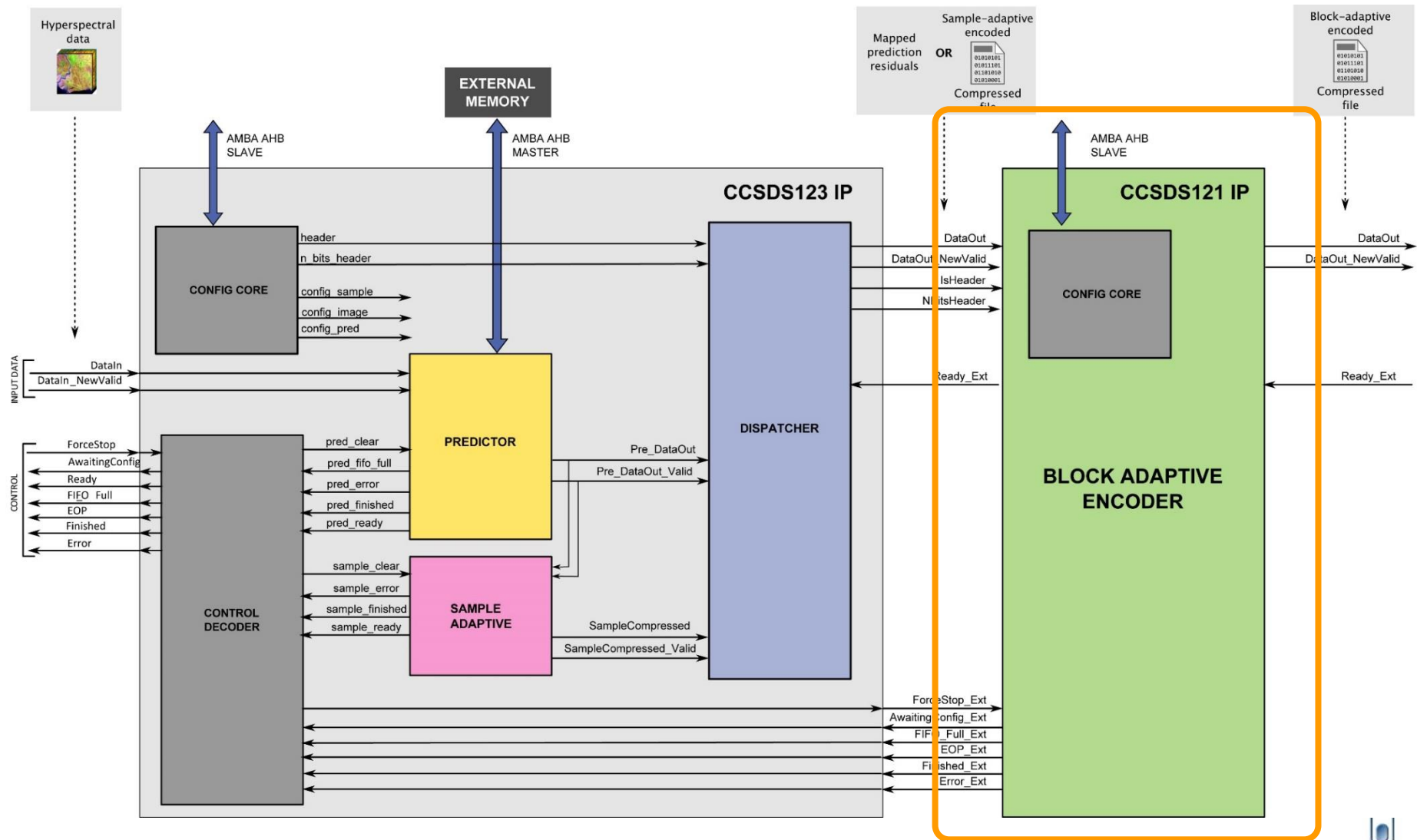


- ▶ Configuration at compile-time: VHDL generic constants.
- ▶ Configuration at runtime: memory-mapped registers.
- ▶ Runtime configuration might be enabled or disabled:
 - When disabled: compile-time generic constants determine the configuration.
 - When enabled: compile-time generic constants determine the boundaries of the runtime configuration values.
- ▶ Example: configuration of number of bands for prediction, P
 - Runtime configuration disabled: constant P_MAX used to set the parameter. AHB slave interface is not used.
 - Runtime configuration enabled: constant P_MAX determines the range of allowed runtime configuration values for P [0 to P_MAX].

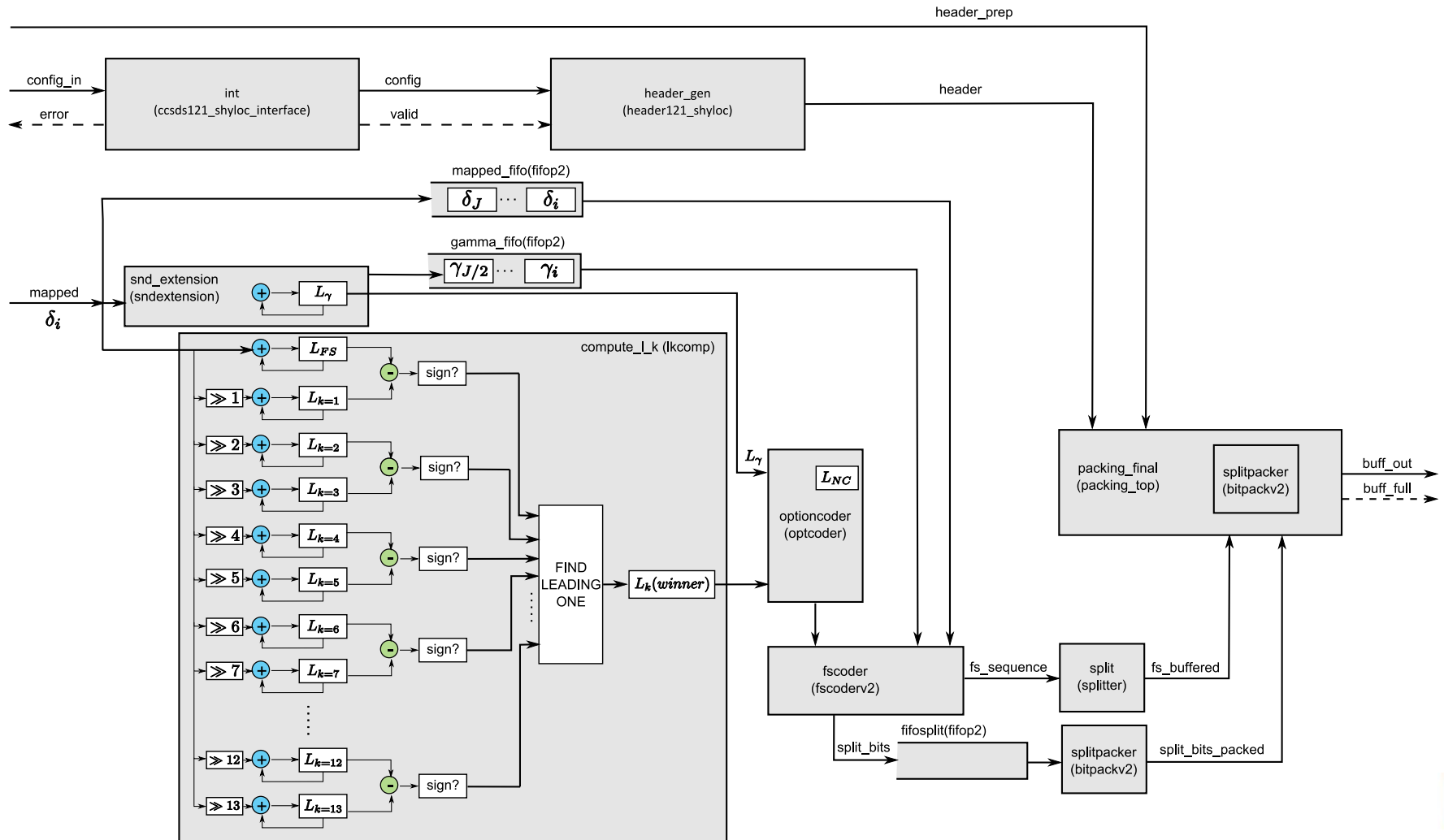
CCSDS123 and CCSDS121 Interfaces



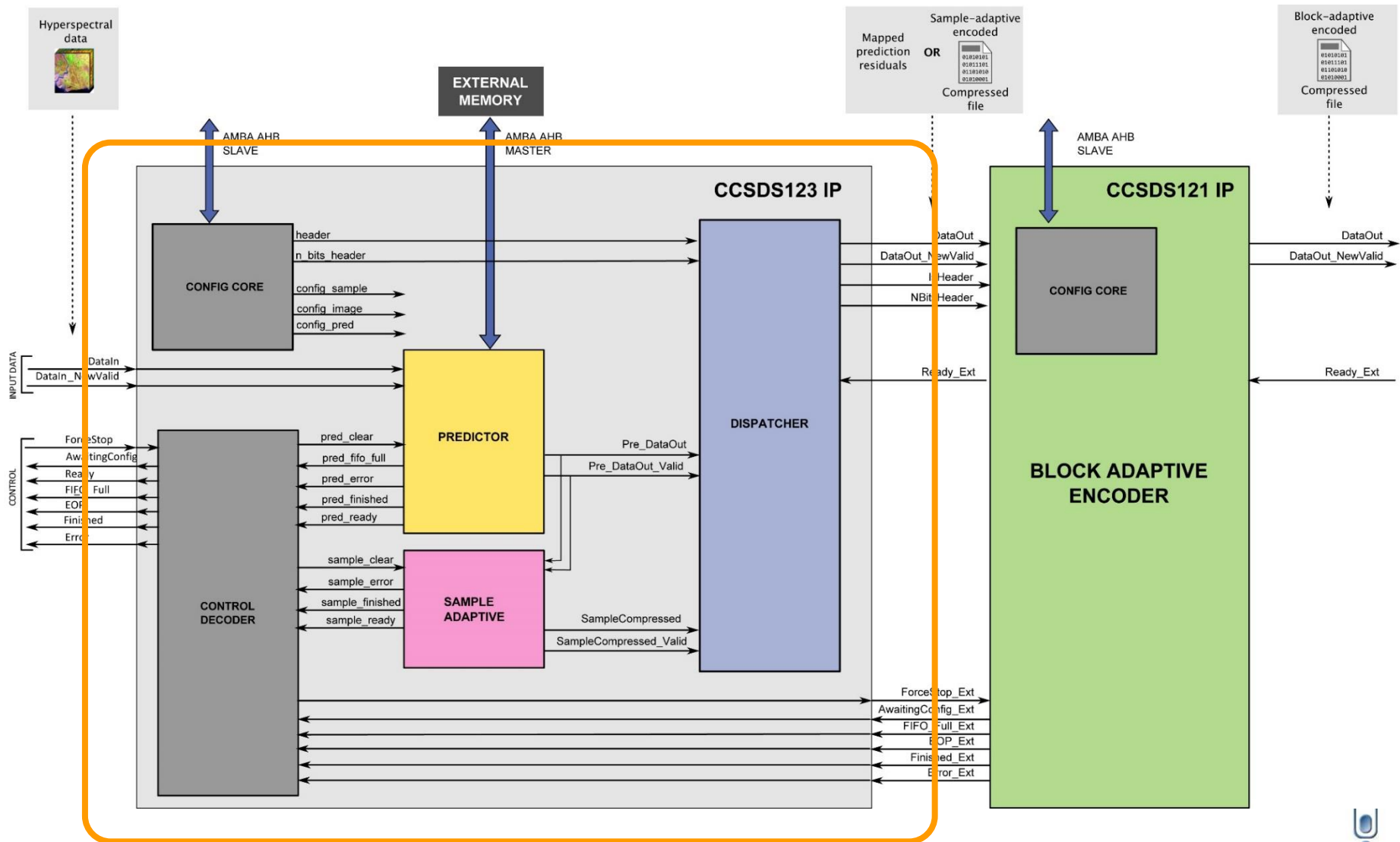
CCSDS123 and CCSDS121



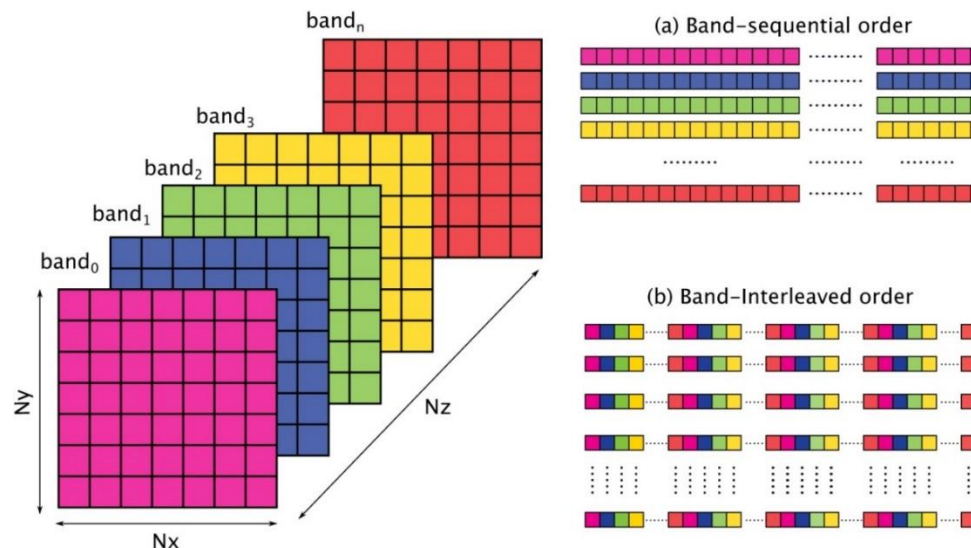
► Simplified schematic (ccsds121_shyloc_comp)



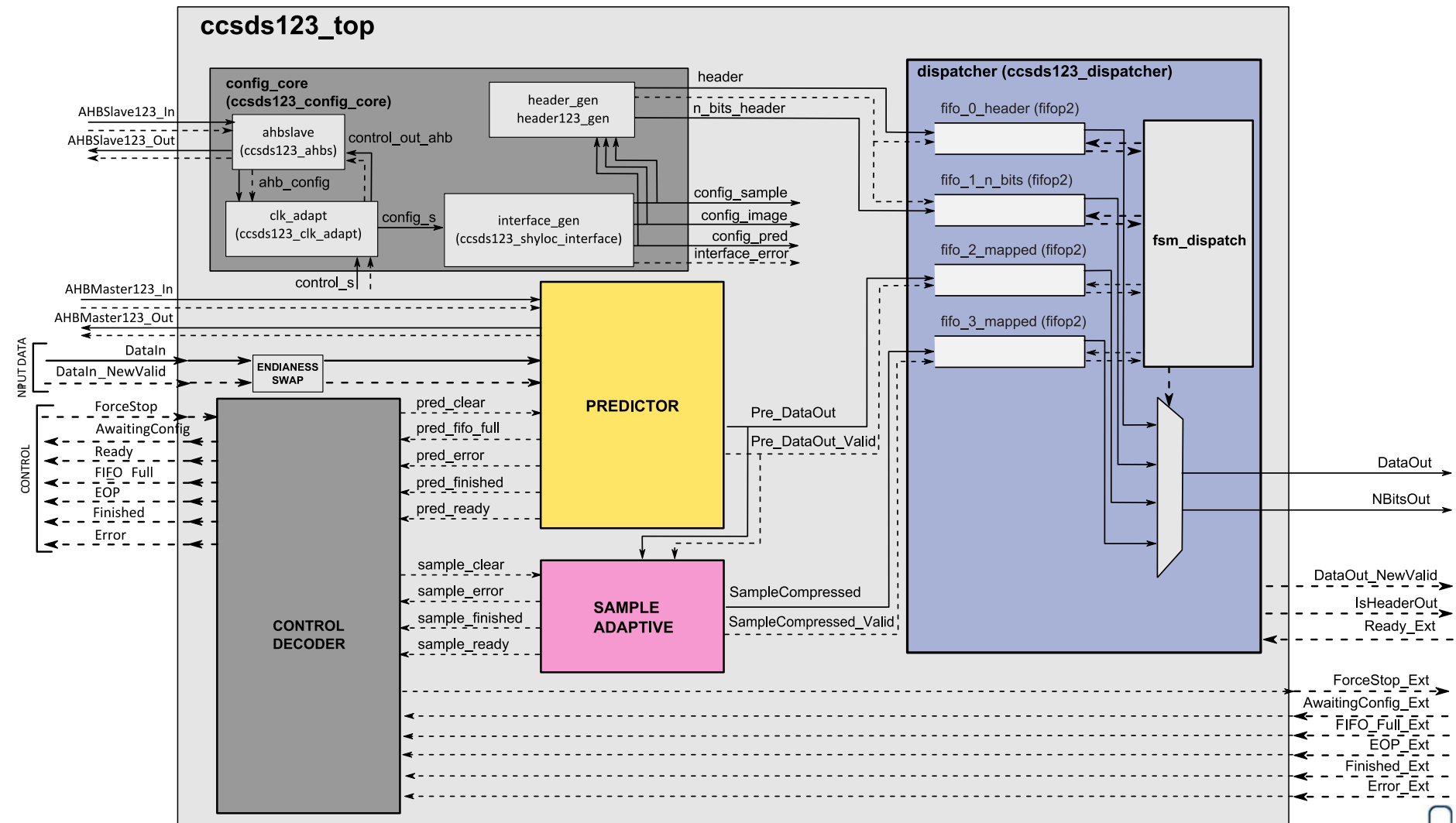
CCSDS123 and CCSDS121 Interfaces



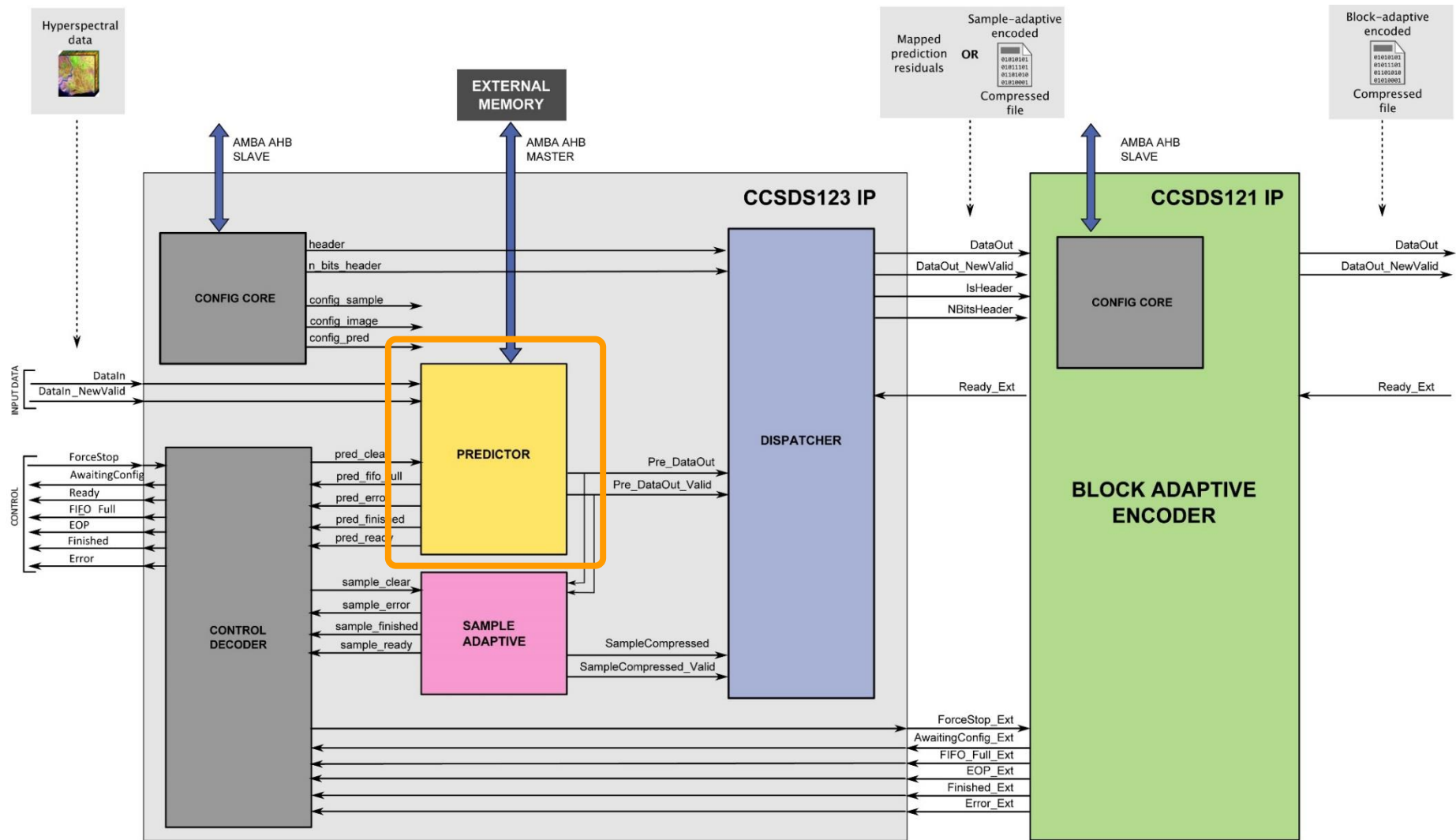
- ▶ Compression order and image dimensions:
 - Different achievable throughput:
 - BIP → allows for parallelization of prediction operations of a sample in all bands.
 - BSQ → prediction finished before starting the compression of samples in the same band.
 - BIL → mixed situation.
 - Different storage requirements depending on compression order, image size and P



CCSDS 123 – Architectural description (top)

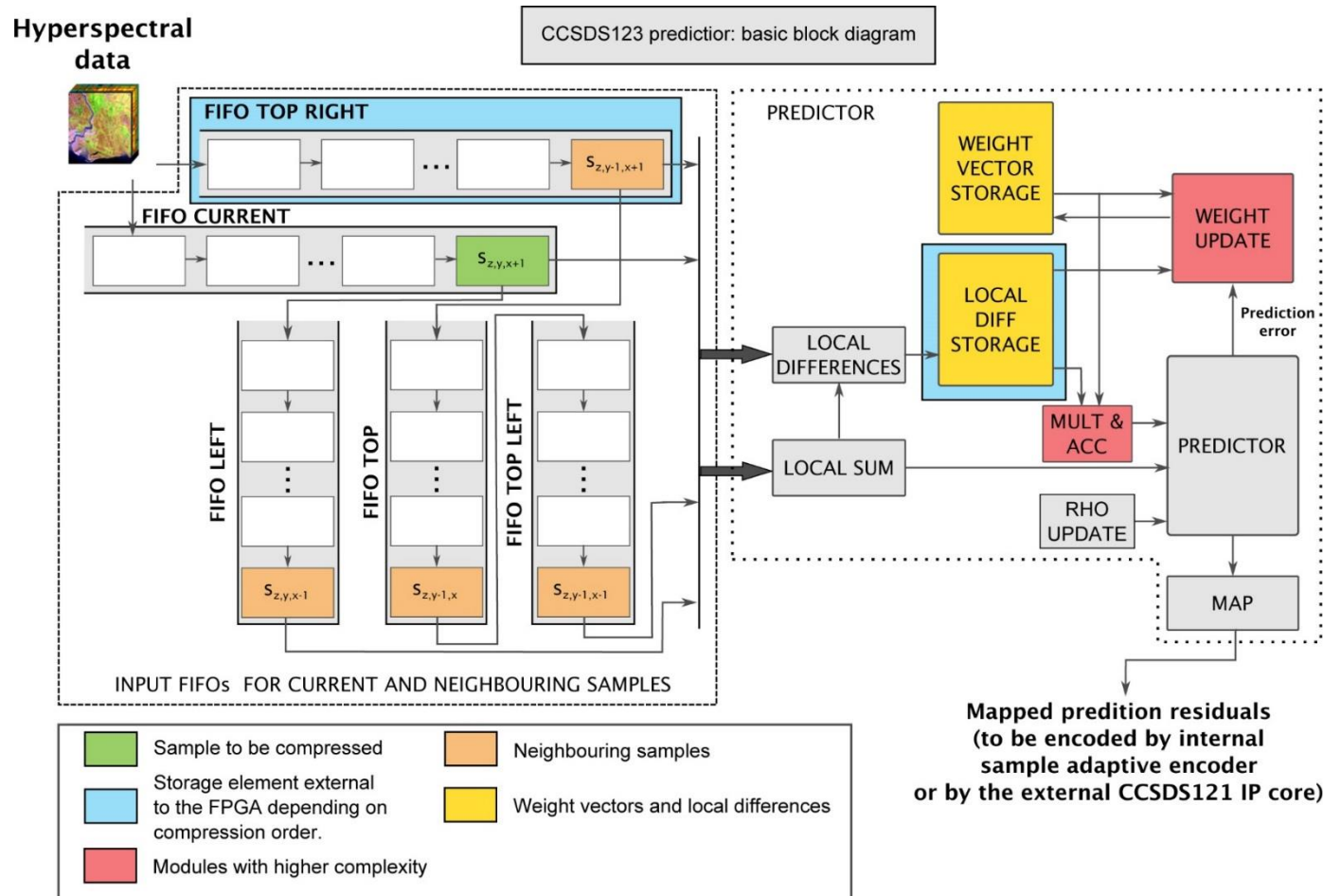


CCSDS123 and CCSDS121

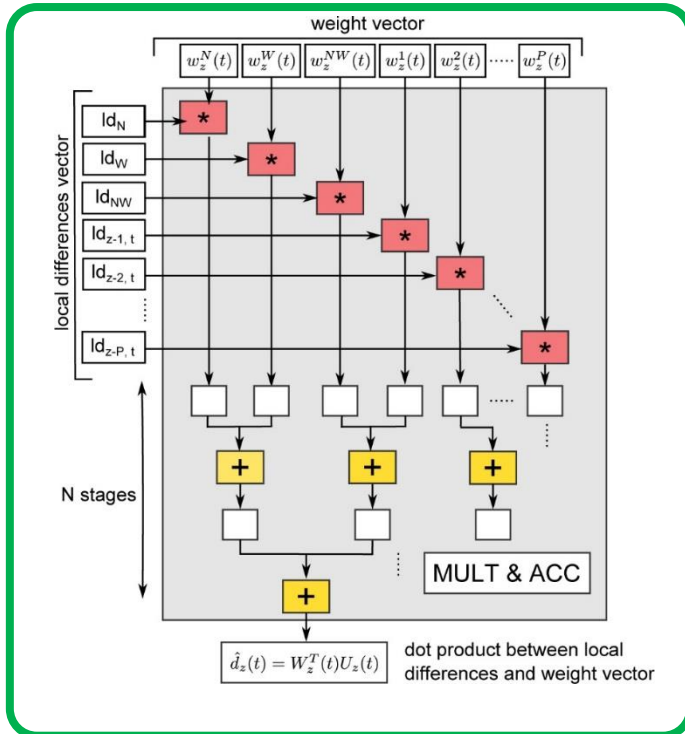


CCSDS123 VHDL description

- ▶ Different architectures, one for each prediction order: BIP, BSQ, BIL.
- ▶ Basic predictor block diagram:

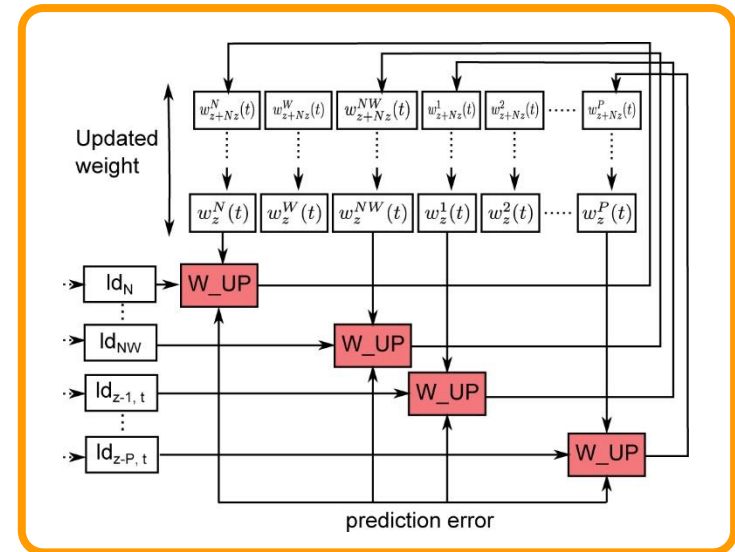
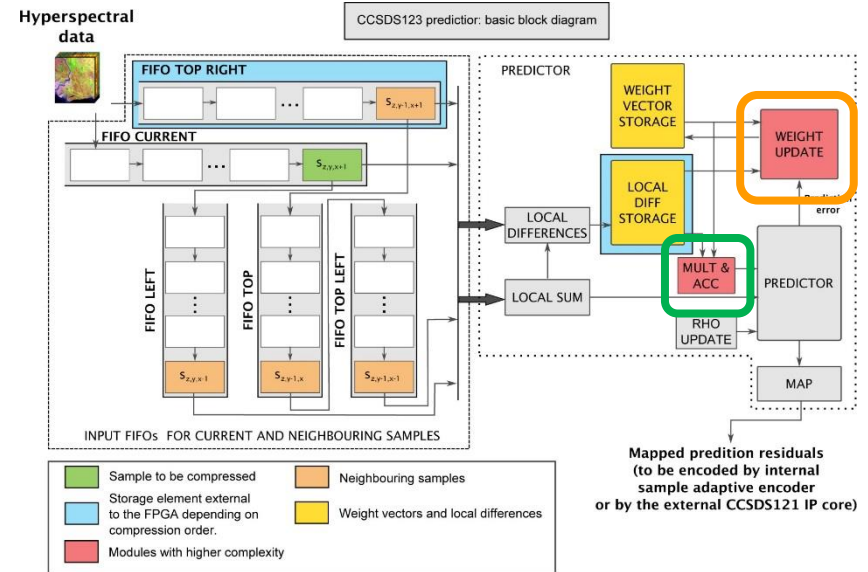


CCSDS123 IP BIP/ BIP-MEM architectures

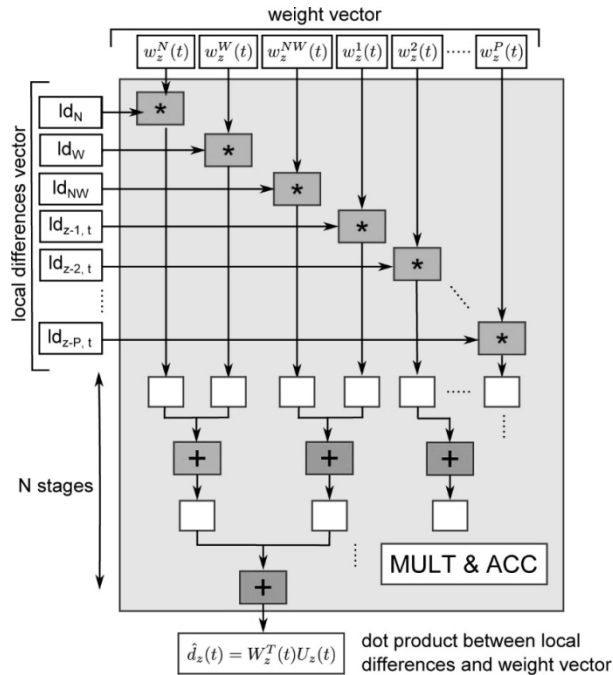


- Parallel structures for dot product computation and weight update.
- Weight vector internally stored. We store one vector per band.
- BIP-mem: top_right FIFO in external memory.

NOTE: $t = x + y * Ny$

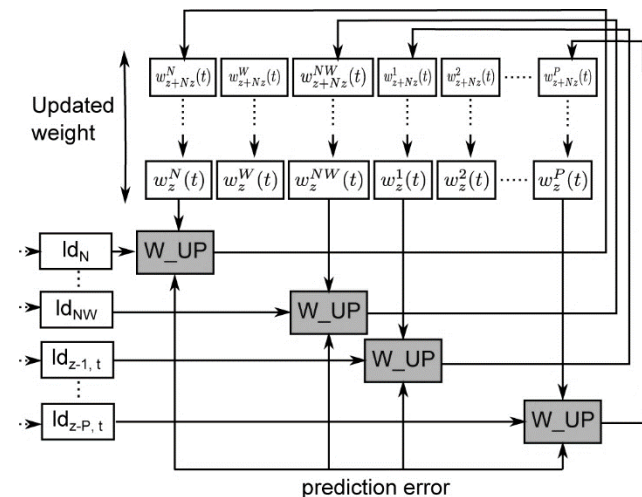
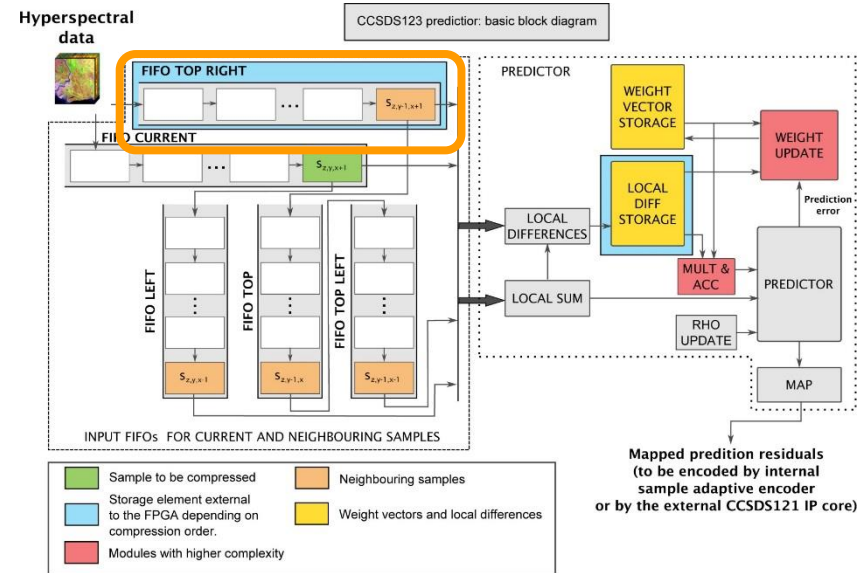


CCSDS123 IP BIP/ BIP-MEM architectures

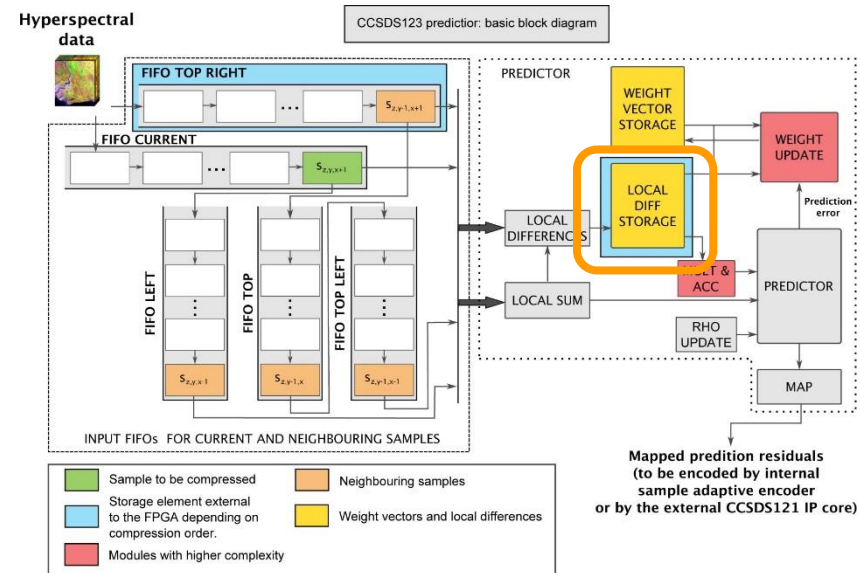
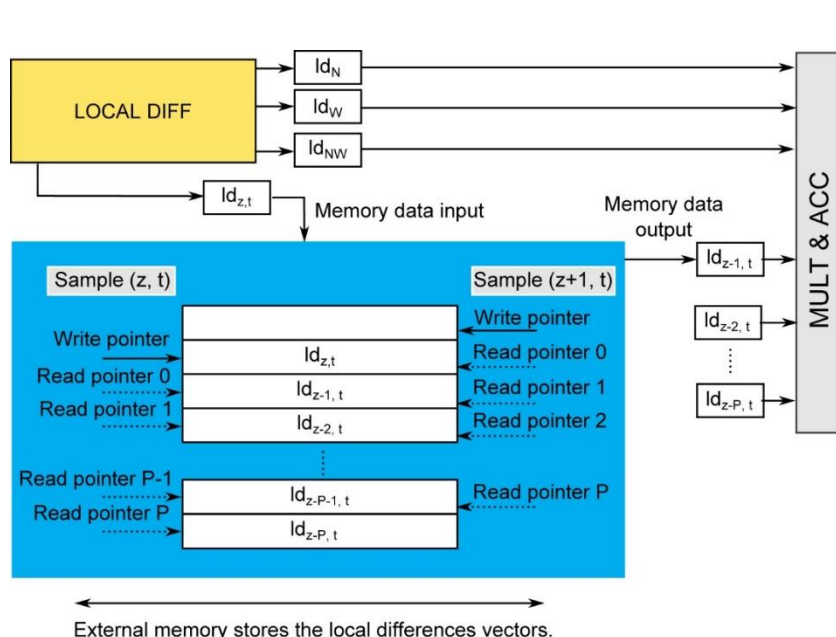


- Parallel structures for dot product computation and weight update.
- Weight vector internally stored. We store one vector per band.
- BIP-mem: top_right FIFO in external memory.

NOTE: $t = x + y * Ny$



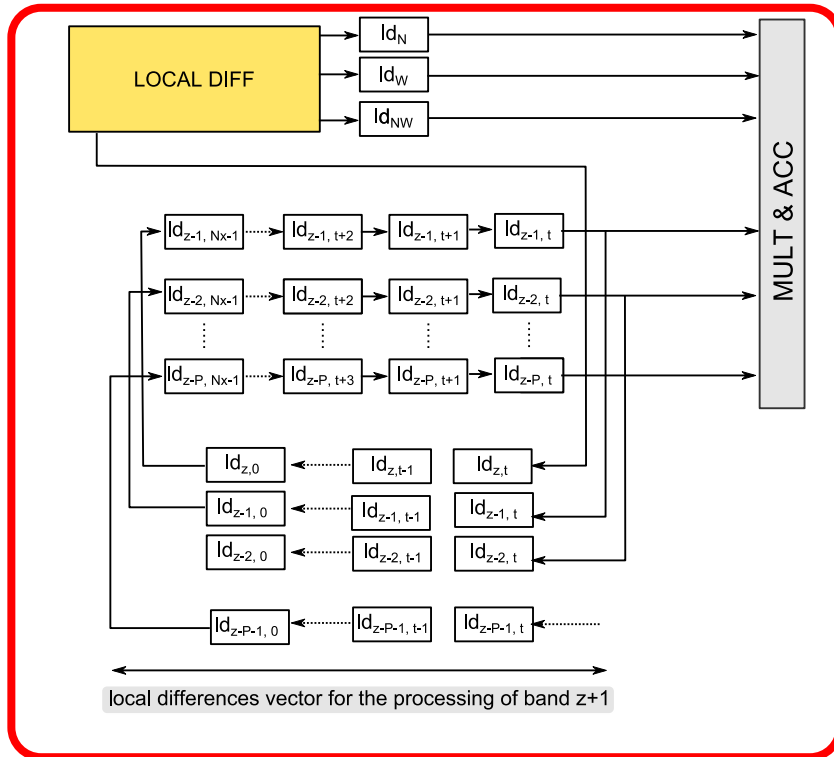
CCSDS123 IP BSQ architecture



- ▶ Local differences vector stored outside FPGA.
- ▶ Dot product and weight update performed serially in order to reduce complexity.

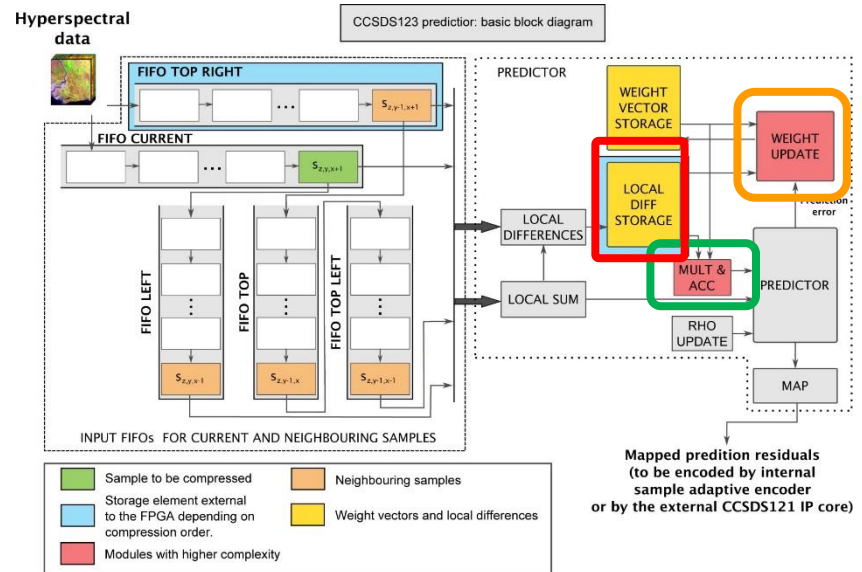
NOTE: $t = x + y * Ny$

CCSDS123 IP BIL architecture

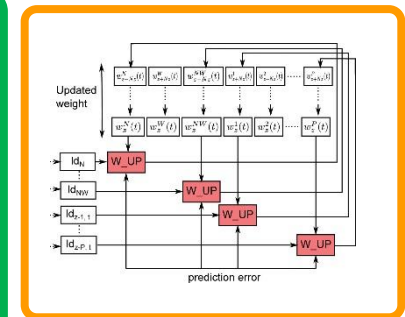
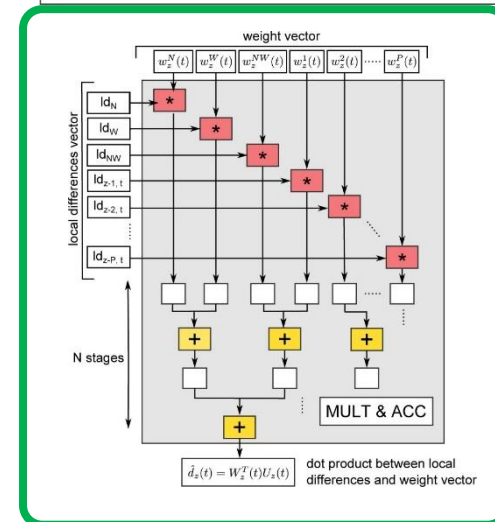


- ▶ Dot product and weight update as in BIP.
- ▶ Additional internal storage for local differences vector.
- ▶ Specific scheduling.

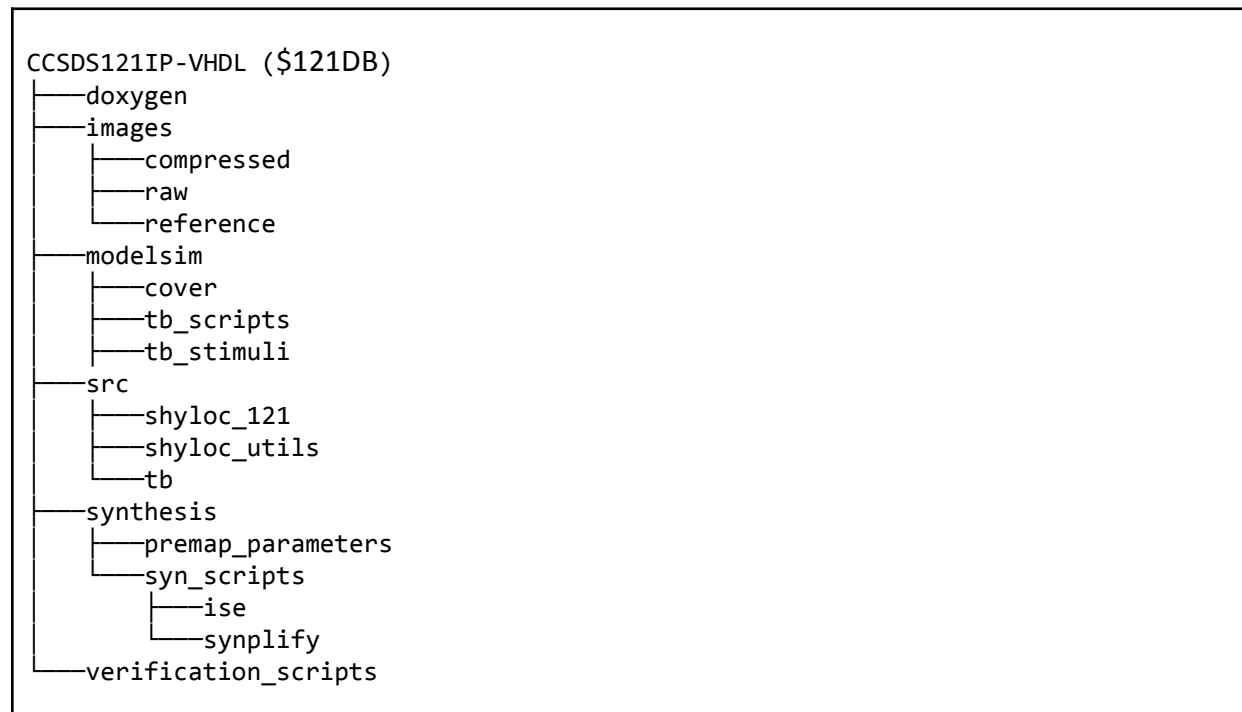
NOTE: $t = x + y * Ny$



Mapped prediction residuals
(to be encoded by internal sample adaptive encoder or by the external CCSDS121 IP core)

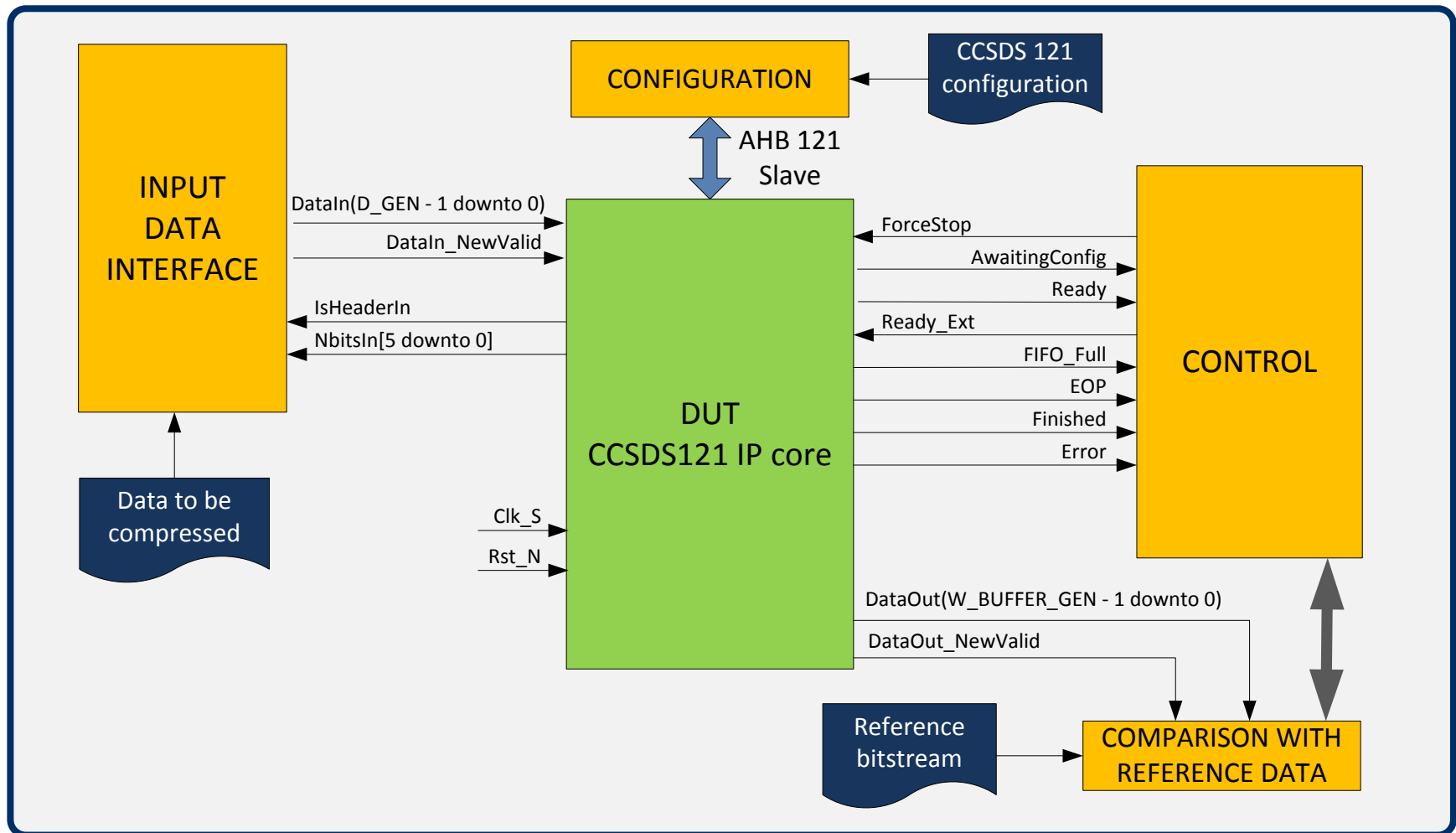


- ▶ The VHDL sources are in a Git repository.
- ▶ A makefile is provided to configure, simulate or synthesize the IP cores.
- ▶ Configurations options are set using *.csv files.
- ▶ A Python script generates the necessary VHDL files to configure the IP core and testbench.



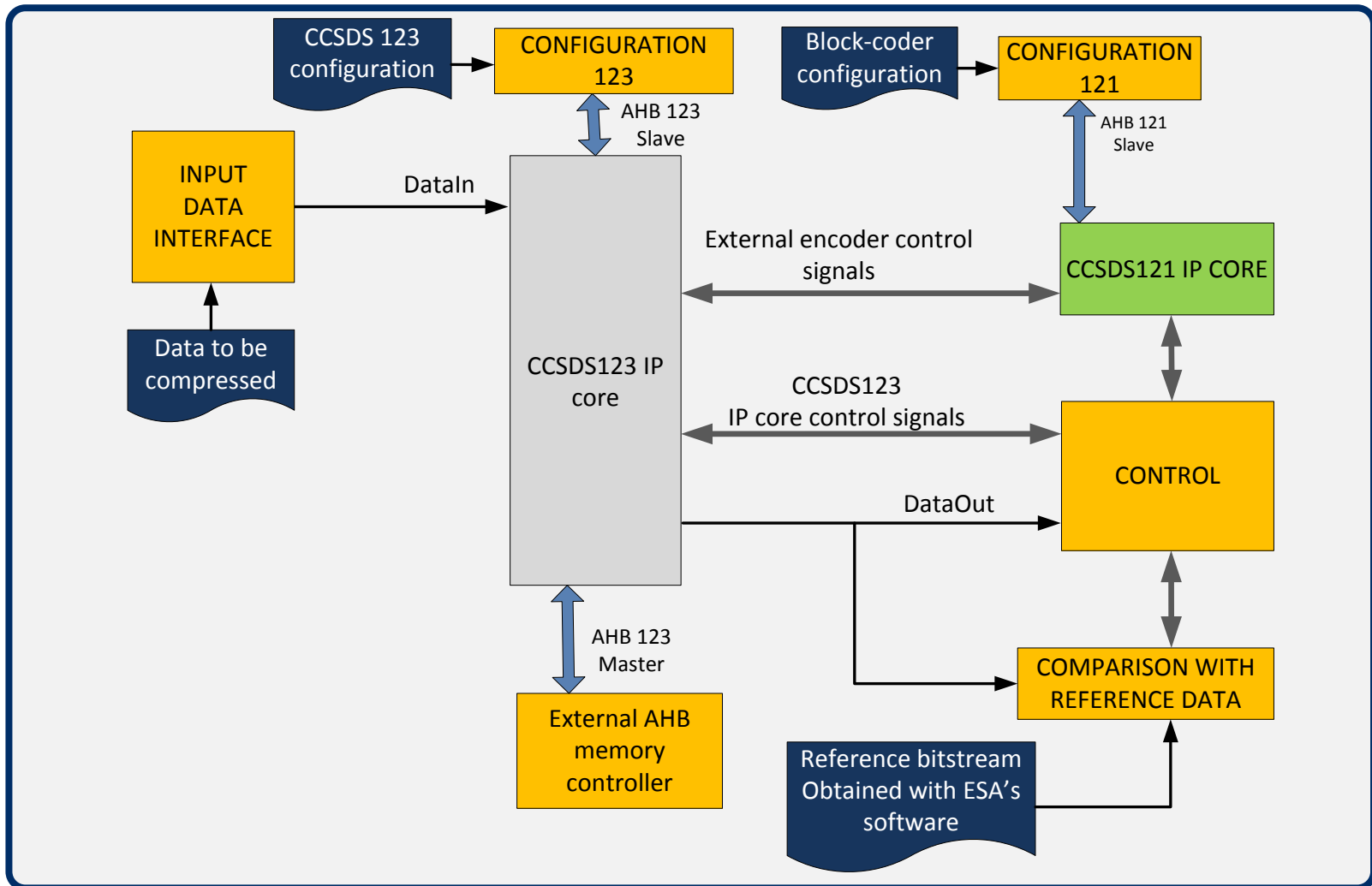
CCSDS 121 - Testbench

- Reference software implementations: [CCSDS 123.0-B-1 software](#) developed by ESA. Emporda software from UAB.



CCSDS 123 - Testbench

- Reference software implementations: [CCSDS 123.0-B-1 software developed by ESA](#). [Emporda software from UAB](#).

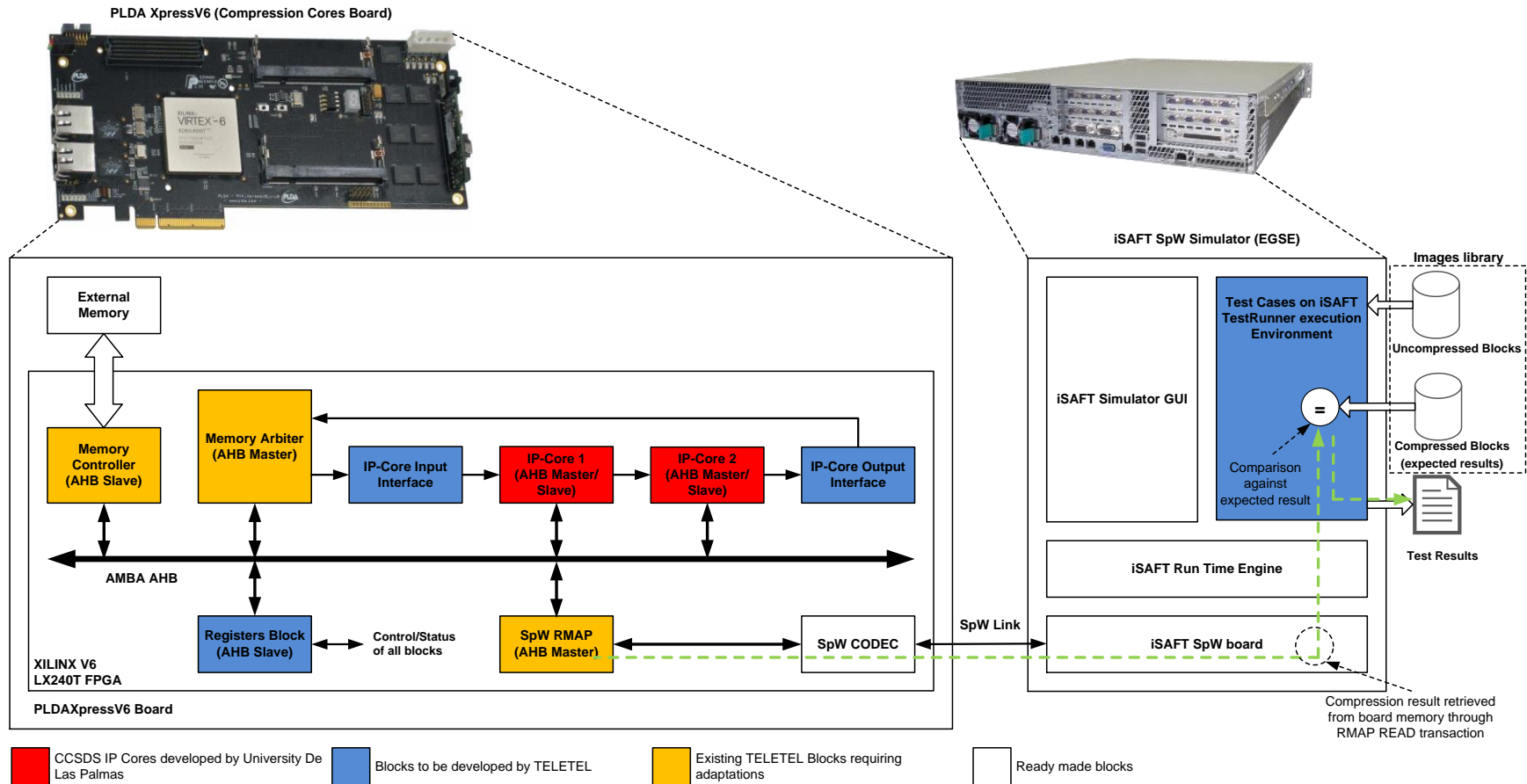


- ▶ Verification dataset combines:
 - 35 images with different number of samples, dynamic range, and sample distribution.
 - 10 sets of compile-time configuration parameters
 - 10 sets of run-time configuration parameters
- ▶ Executed tests:
 - Basic Sanity (BS) → 36 tests
 - Intentional → 15 tests
 - Stress → 46 tests
 - Additional tests to improve coverage → 22
- ▶ Total: 121 tests.
- ▶ Simulations are performed with QuestaSim, and automated with scripts.
- ▶ A pass/fail simulation report is generated.

- ▶ Demonstrator with CCSDS 123 + CCSDS 121 IPs.
- ▶ **Compression Core Board:** An FPGA based board with SpW interfaces. Based on PLDA's XpressV6:
 - Xilinx Virtex-6 LX240T.
 - Up to 2x4GB DDR2 SDRAM
 - Extension connector with up to 168 signals
 - PCIe core with multi-channel DMA host interface
- ▶ **EGSE:** An EGSE based on the iSAFT Simulator connected to the Compression Core board through a SpW link which will validate the core's functionality, by stimulating the board and retrieving the results through RMAP transactions over a SpW link.

SHyLoC– Validation

- ▶ Demonstrator with CCSDS 123 + CCSDS 121 IPs.



- Testcases → 4 sets of compile-time parameters
 - BIP-MEM (01_BS_Val) → (Up to Nx = 8192; Ny = 8192; Nz = 2048)
 - BIP (02_BS_Val) → (Up to Nx = 1024; Ny = 512; Nz = 256)
 - BSQ (03_BS_Val) → (Up to Nx = 1024; Ny = 1024; Nz = 1024)
 - BIL (04_BS_Val) → (Up to Nx = 512; Ny = 512; Nz = 256)
- In all tests, the CCSDS121 is configured as external entropy coder
- In total: 17 validation tests.
- Results of performance tests:
 - Running 02_Val test:
 - BIP architecture
 - AVIRIS image (Nx = 677; Ny = 512; Nz = 224; 16 bits per sample)
 - Run-time configuration enabled
 - AHB frequency: 125 MHz; Core frequency: 62.5 MHz
 - Total from reception of first valid input sample until end of compression: 77987924 cycles
 - Throughput ~ 1 Gbps

Introduction and motivation

CCSDS lossless compressors overview

SHyLoC SystemC Model

SHyLoC VHDL Description and Verification

Technology Mapping to FPGA and ASIC

Conclusions

- ▶ Different configurations representative of use cases have been synthesized.
 - CCSDS121: 3 sets with compile-time configuration only; 1 set with run-time configuration.
 - CCSDS123: 8 sets with compile-time configuration only; 4 set with run-time configuration.
- ▶ Synthesis has been performed for the following technologies:
 - FPGA:
 - Xilinx Virtex 5 & 5QR
 - Microsemi ProASIC3E, ProASIC3L, RTAX2000, RTAX4000 and RTG4
 - ASIC DARE 180 nm

SHyLoC – CCSDS121 mapping

- ▶ Synthesis results for Virtex5 FX130; RTG4 150 and DARE 180nm
- ▶ Baseline encoder configuration with:
 - Block size $J = 16$
 - Dynamic range of input samples, $D = 16$
 - Bit width of output buffer $W_BUFFER = 32$.

Virtex5 FX130	Usage	
BRAM	0	0%
DSP48	7	2%
LUT	3495	4%
Est. Freq (MHz)	118	
Msamples/second	118	

RTG4 150	Usage	
MACC	8	2%
RAM64x18_RT	11	5%
LUTs*	6418	4%
Est. Freq (MHz)	42	
Msamples/second	42	

DARE 180nm	Usage
Area (mm ²)	2.985
Cells (kilo)	55.97

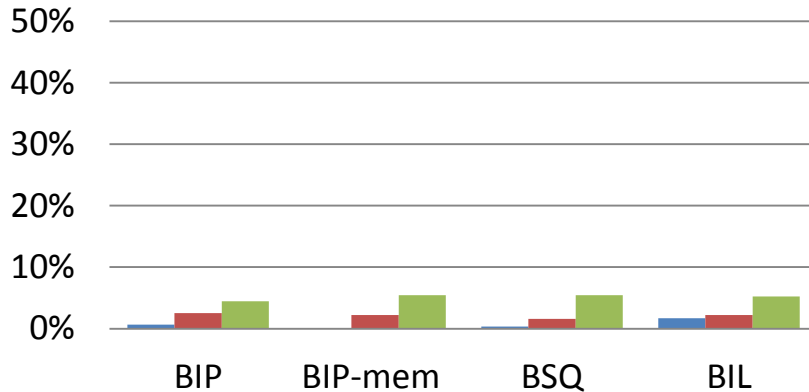
- ▶ Compile-time configurations for specific images and runtime configurable.
- ▶ Synthesis performed for Virtex5 FX130 and RTG4 150.
- ▶ Baseline predictor configuration with:
 - Number of bands for prediction, $P = 3$
 - Weight component resolution $W = 13$
 - Neighbor oriented and full prediction.
 - Sample-adaptive encoder is always implemented.
 - Bit width of output buffer $W_BUFFER = 32$.

IMAGE	Nx	Ny	Nz	bpp
LANDSAT	1024	1024	6	8
AVIRIS	512	680	224	16
AIRS	90	135	1501	14
RUNTIME CONFIG	512	1024	256	16

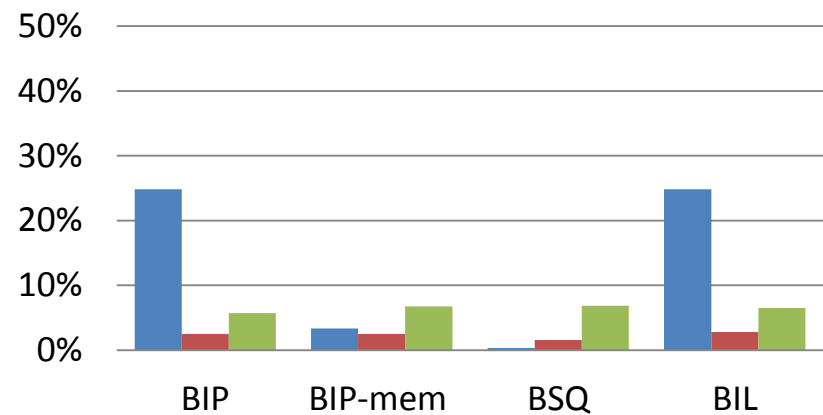
Synthesis on Virtex5 FX130

Resource usage VS Image size VS Predictor Architecture

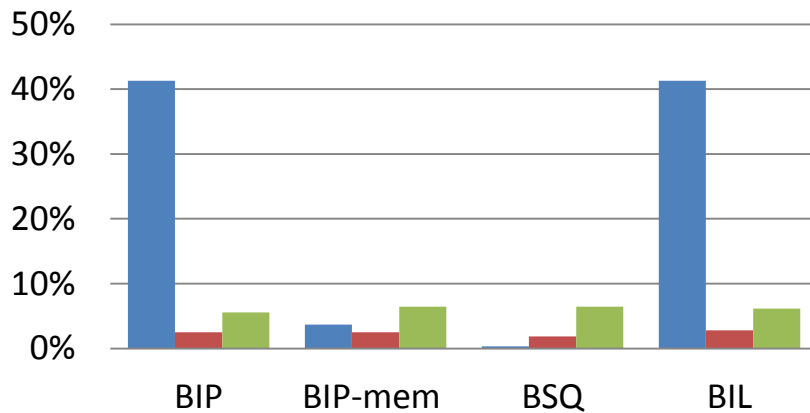
LANDSAT



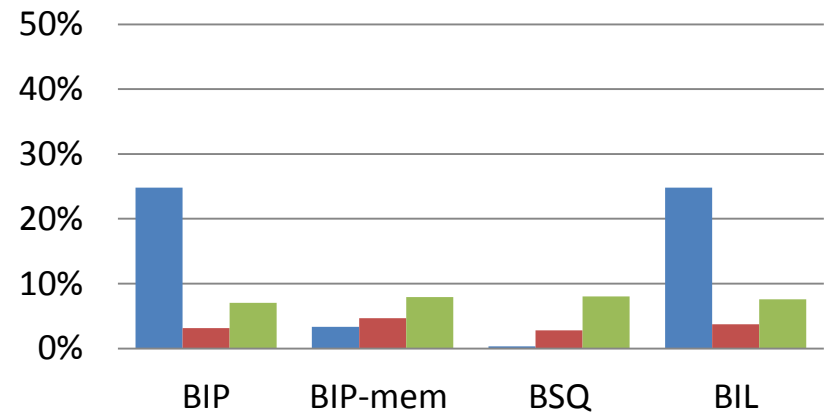
AVIRIS



AIRS



RUNTIME CONFIGURABLE

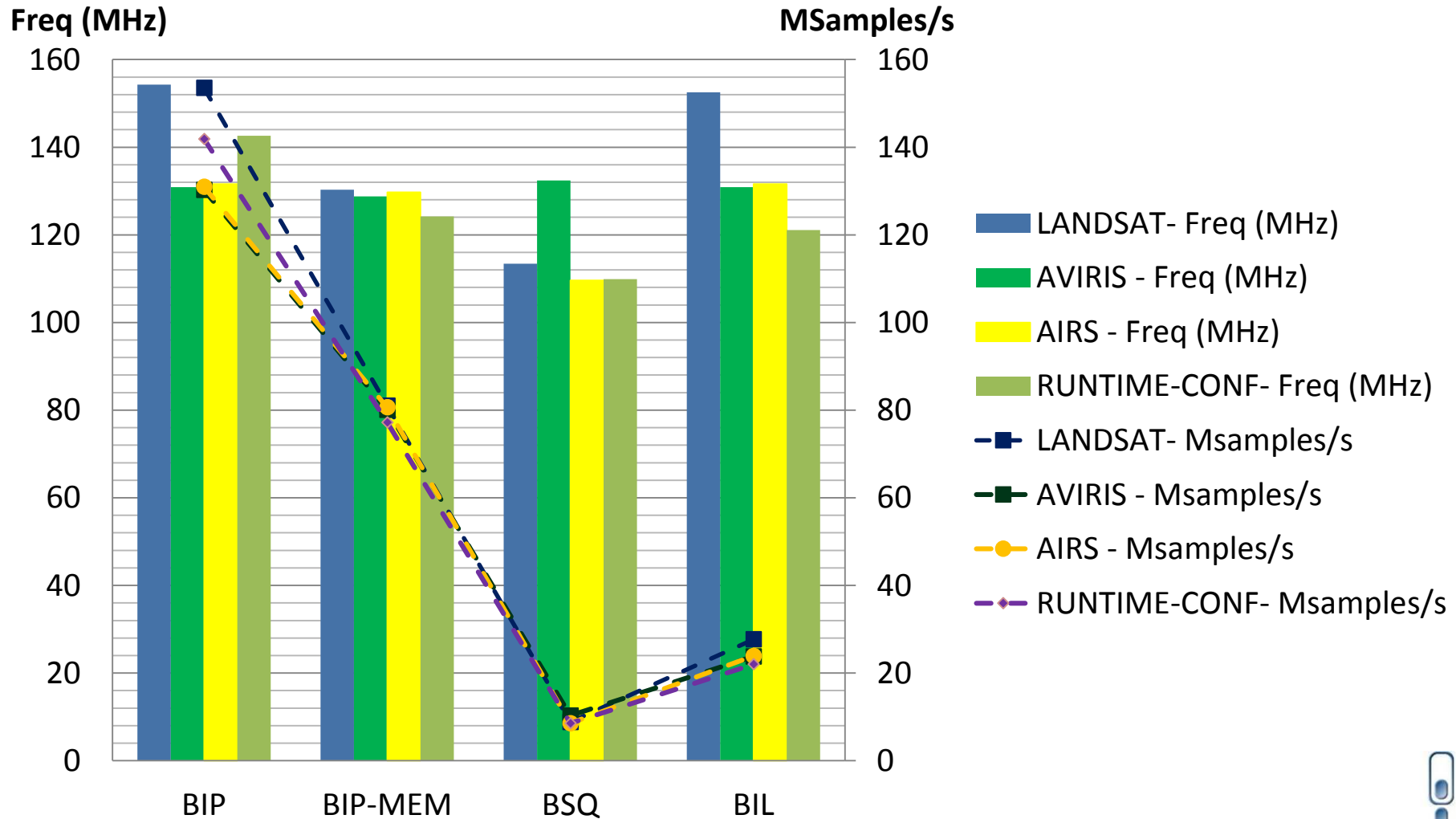


■ % of BlockRAMs ■ % of DSP48 ■ % of LUTs

■ % of BlockRAMs ■ % of DSP48 ■ % of LUTs

Synthesis on Virtex5 FX130

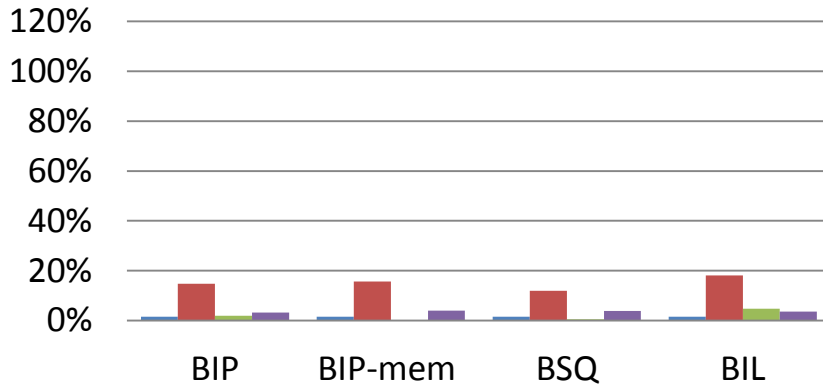
Frequency and throughput VS Image size VS Predictor Architecture



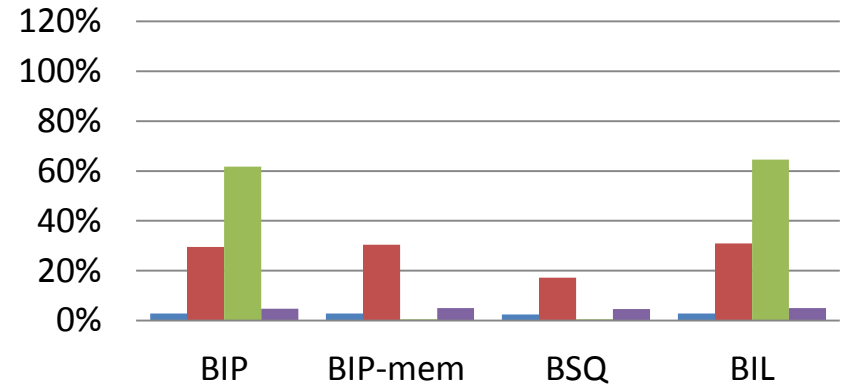
Synthesis on RTG4 150

Resource usage VS Image size VS Predictor Architecture

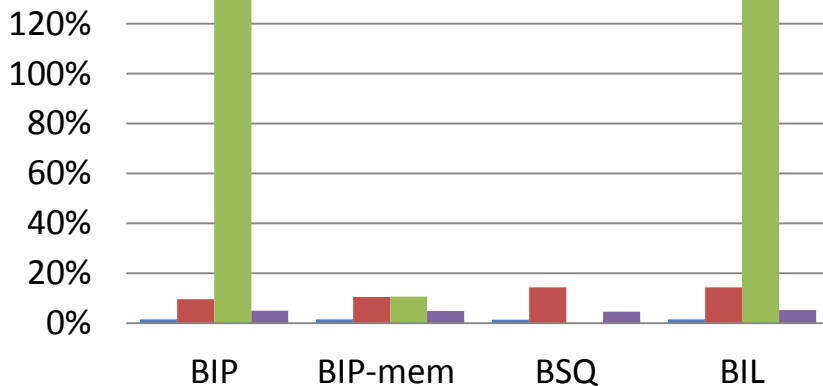
LANDSAT



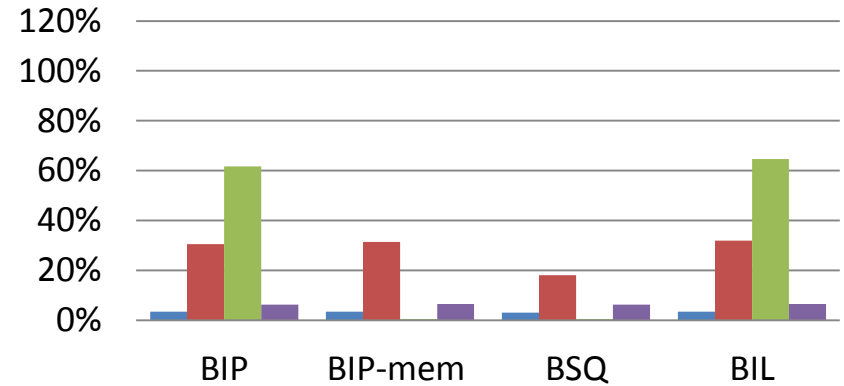
AVIRIS



AIRS



RUNTIME CONFIGURABLE

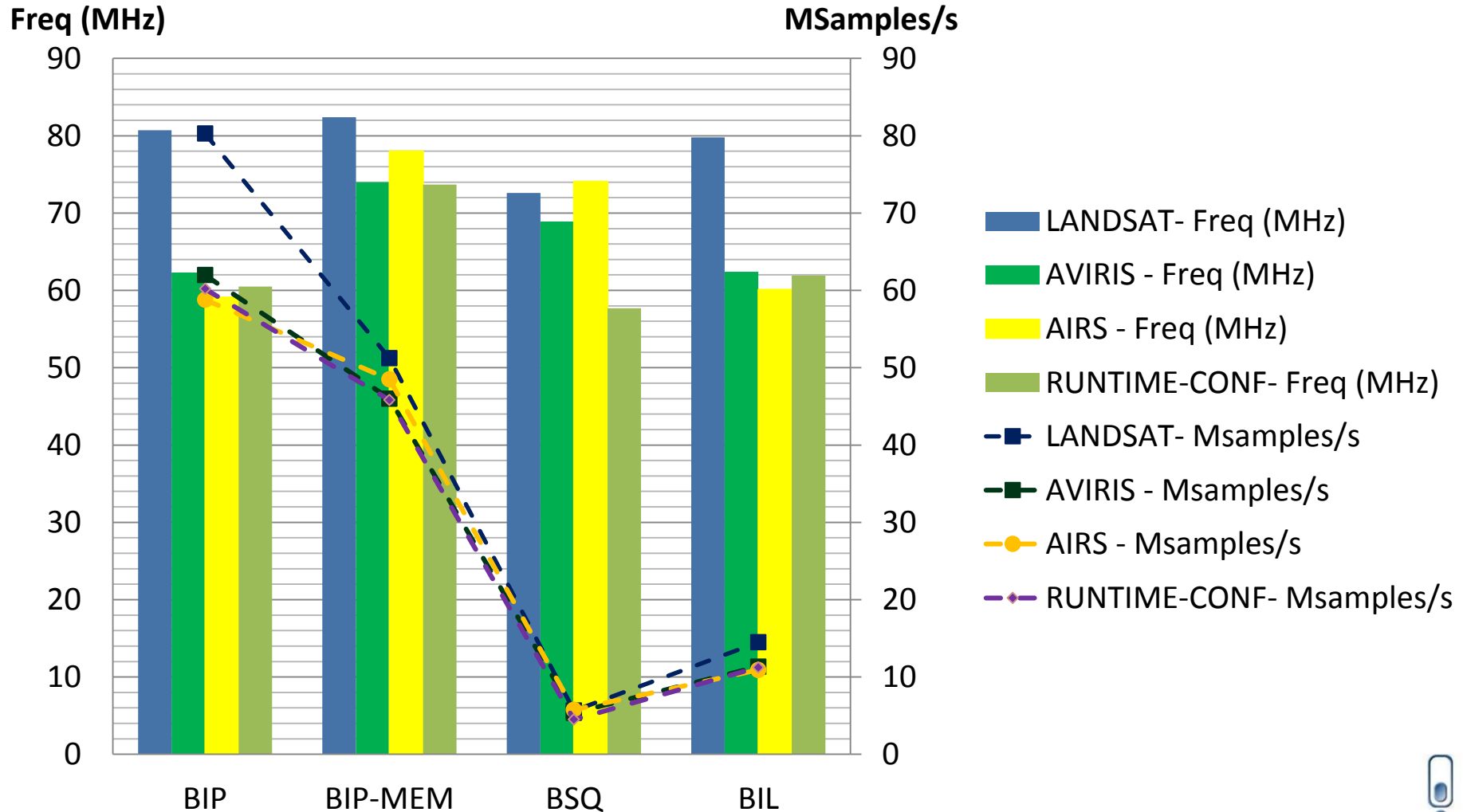


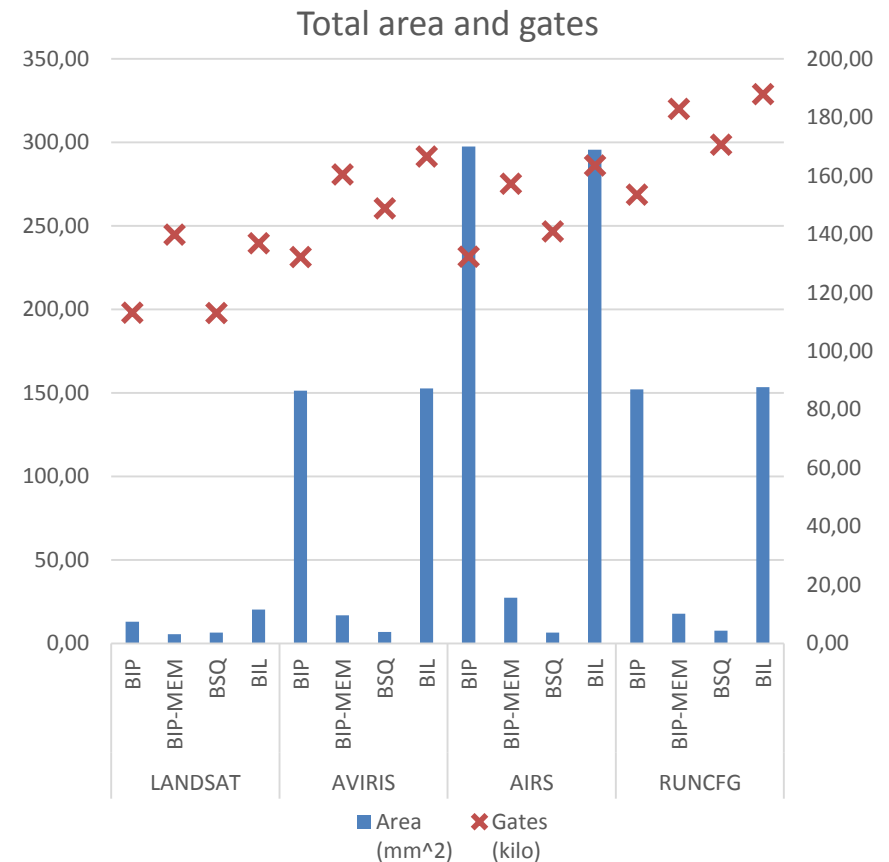
■ MACC ■ RAM64x18_RT ■ RAM1K18_RT ■ LUTs*

■ MACC ■ RAM64x18_RT ■ RAM1K18_RT ■ LUTs*

Synthesis on RTG4 150

Frequency and throughput VS Image size VS Predictor Architecture





Notes:

- Memories smaller than 64 words are mapped into FF.
- There's an overhead in memory usage due to port bit width alignment to match the memories view available for this project.

- ▶ We have presented the hardware architecture and VHDL description IP cores, which perform lossless compression as specified by the the CCSDS-121 and CCSDS-123 standards.
- ▶ The cores might work independently as well as jointly, offering simple plug-and-play compatible interfaces.
- ▶ Technology independent. Configurable at compile-time and runtime.
- ▶ Resource usage and throughput depend on the selected compile-time configuration.
- ▶ Mapped to 7 different FPGA devices: Xilinx Virtex 5 & 5QR; Microsemi ProASIC3E, ProASIC3L, RTAX2000, RTAX4000 and RTG4
- ▶ Feasibility of implementation on Virtex5 FX130. Maximum throughput 153 Msamples/s.
- ▶ Low complexity: maximum 7% of LUTs Virtex5 and 13% RTG4.
- ▶ Demonstrator validates design and shows throughput of up to 1 Gbps.



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



EXPRO+ ESA AO/1-8032/14/NL/AK

CCSDS Lossless Compression IP-CORE Space Applications

Final Presentation Days

8th May 2017