CEI UPM | Centro de Electrónica Industrial

Reconfigurable Architectures for On-Board Processing with Adaptive Fault Tolerance using COTS MPSoCs

Arturo Perez*, Alfonso Rodriguez, Andrés Otero*, **Eduardo De La Torre***, Yubal Barrios**, Antnio Sánchez**, Sebastián López**

* Centre for Industrial Electronics, Universidad Politécnica de Madrid

** Institue for Applied Microelectronics, Univ. las Palmas de Gran Canaria

cei@upm.es

POLITÉCNICA

Rad-hard versus COTS Devices

ECSEL JU Project ENABLE-S3

UC8 Space Use Case:
Reconfigurable Video Processor

150nm rad-hard bulk CMOS
Up to 200MHz
400 DMIPS at 200MHz

RAD750

A53 processor:
Up to 1,5 GHz
3450 DMIPS at 1,5GHz
Real-Time R5 processor
Up to 600 MHz
1470 DMIPS at 600MHz
FPGA HW

Zynq Ultrascale+

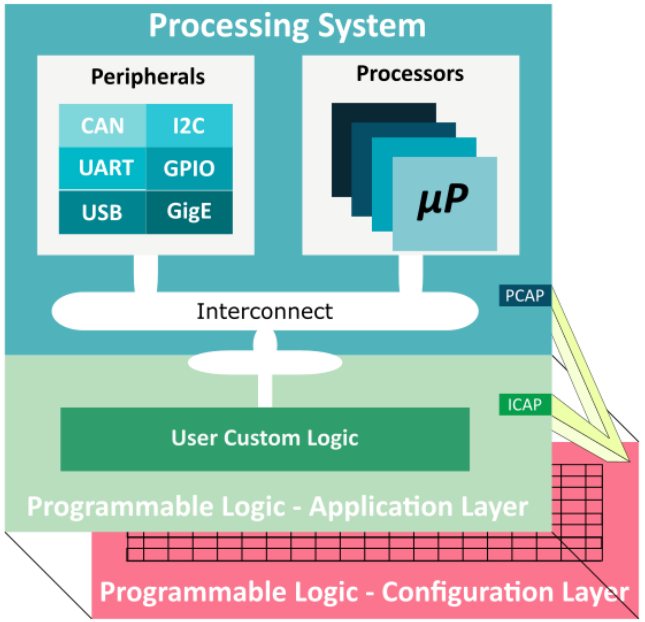
Reconfigurable MPSoC Devices

PROS

- Low cost
- High computing power
- Heterogeneity
- Diversity
- Flexibility (Reconfigurability)

CONS

- Configuration memory subject to:
  SEU, SEL, SEFI … => SEE in general

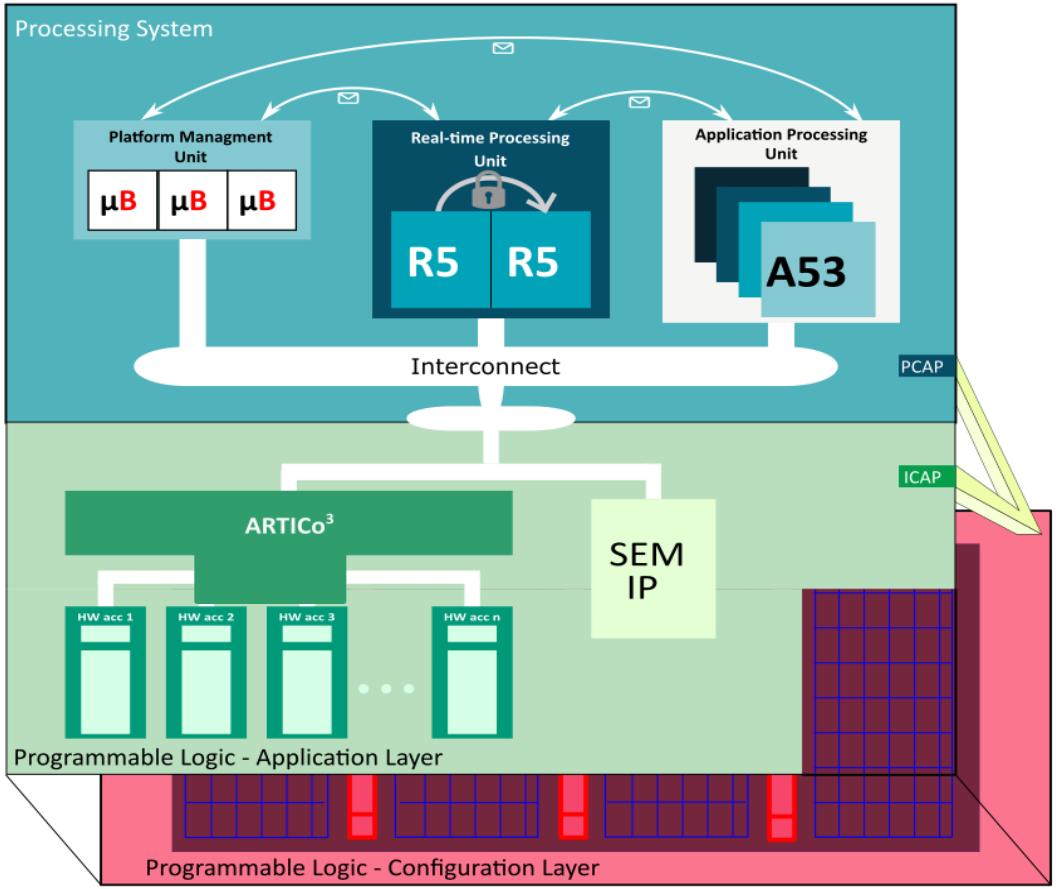


HARDENING TECHNIQUES

- At Logic Layer → Ex.: modular redundancy
- At Configuration Layer → Ex.: scrubbers

Zynq UltraScale+ Hardened Fabrics and Features

- **Reliable fabrics:**
  - RPU (R5):
    - 2 x Cortex-R5 Processors
    - Native Lockstep – 1 core
  - PMU:
    - TMR microblaze
- **Soft-error mitigation** (SEM-IP) embedded features:
  - Frame ECC
    - Error detection/correction
  - CRC
    - Error detection



Reconfigurable Video Processor

rtems
Real-Time Executive for Multiprocessor Systems

Mission:

- Hyperspectral Image Compression
- Satellite Navigation

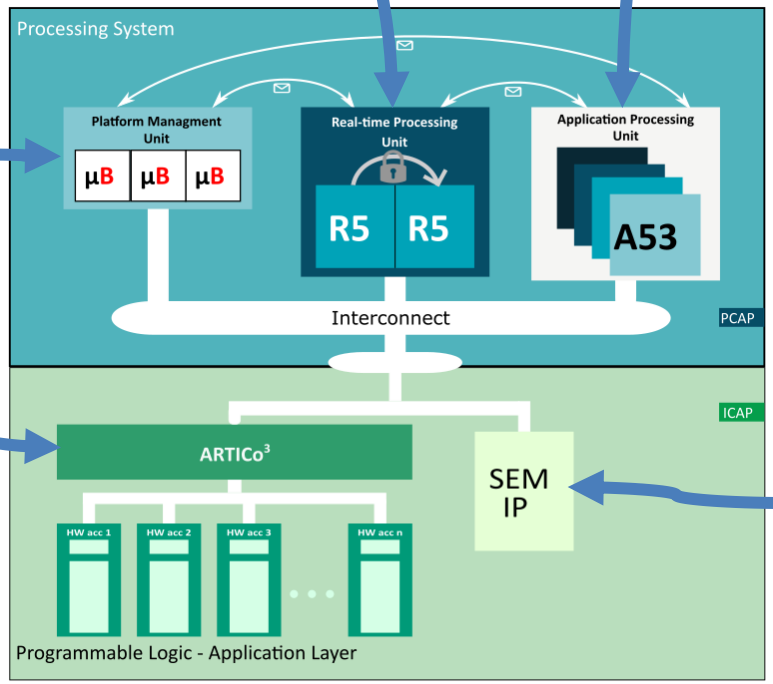
Support:

- Hybrid scrubber and repair
- Dynamic Partial Reconfiguration (DPR)

Linux

- Ethernet Interface
- Gather monitor data
- Receive Hyperspectral Images
- Receive TC/TM

- Error-management support
- Lock-step errors



ARTICo

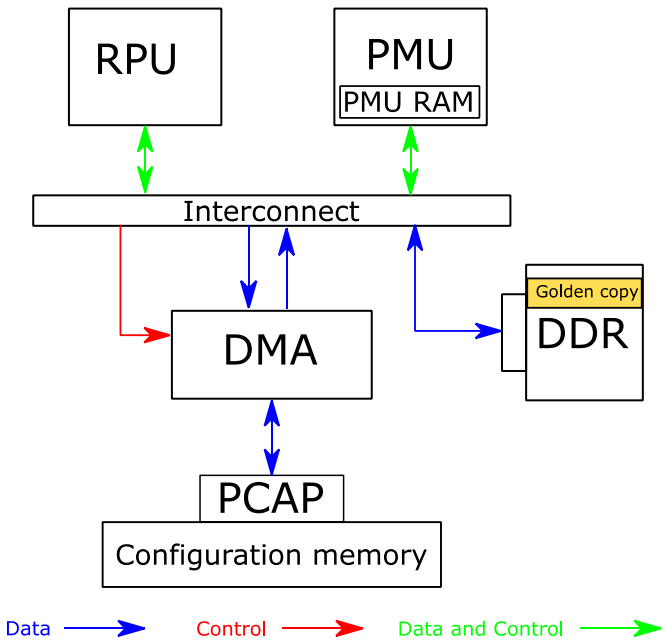
- HW accelerators (Hyperspectral Images Compression)

- ECC/CRC scrubber
- Fault Injection

CEI UPM

POLITÉCNICA

Readback scrubbers – Read Performance (AHS 2018)



Dest. memory / PCAP freq. [MHz]	DDR	PMU RAM
187.5	24	2
150	71260 – full mem	2
125	71260 – full mem	3
93.75	71260 – full mem	5
62.5	71260 – full mem	15
46.88	71260 – full mem	30

Table 0: Maximum number of frames that can be read depending on the destination memory and PCAP frequency

Mode / Frames	R5 no-cache	R5 cache	PMU DDR	PMU RAM
2	**65µs**	**18µs**	**99.86µs**	**26.9µs**
5	**101µs**	**21µs**	**225.95µs**	74.51µs
15	**225µs**	**46µs**	**646.64µs**	193.28µs
30	666µs	136µs	2.05ms	312.04
500	6.98ms	1.44ms	21.52ms	-[a]
5000	69.29ms	14.3ms	214.36ms	-[a]
50000	690ms	142.95ms	2.14s	-[a]
[a]. PMU RAM exceeded				

Table 1: Read time with PCAP frequency: 187.5MHz

Mode / Frames	R5 no-cache	R5 cache	PMU DDR	PMU RAM
2	**65µs**	**18µs**	**100.98µs**	**27.19µs**
5	**103µs**	**21µs**	**227.78µs**	41.02µs
15	**231µs**	**49µs**	**650.54µs**	144.89µs
30	**414µs**	**89µs**	**1.28ms**	294.72µs
500	**6.27ms**	**1.35ms**	**21.12ms**	-[a]
5000	**62.17ms**	**13.36ms**	**210.98ms**	-[a]
50000	**621.21ms**	**133.48ms**	**2.11s**	-[a]
[a]. PMU RAM exceeded				

Table 2: Read time with PCAP frequency: 125MHz

Mode / Frames	PMU DDR	PMU RAM
2	**105.08µs**	**28.06µs**
5	**235.66µs**	**42.66µs**
15	**670.87µs**	**90.93µs**
30	**1.32ms**	**164.3µs**
500	**21.74ms**	-[a]
5000	**217.21ms**	-[a]
50000	**2.17s**	-[a]
[a] PMU RAM exceeded		

Table 3: Read time with PCAP frequency: 46.88MHz

CEI UPM

POLITÉCNICA

Readback Scrubbers – Compare and Correct Performance (AHS 2018)

Comparison Time:

Mode / Frames	R5 no cache	R5 cache	PMU DDR	PMU RAM
2	109µs	12µs	223.4µs	166.7µs
5	270µs	23µs	561.4µs	416.23µs
15	808µs	78µs	1.67µs	1.25ms
30	1.57ms	178µs	3.35ms	2.5ms
500	24.64ms	3.46ms	55.7ms	[a]
5000	254.42ms	34.61ms	553.7ms	[a]
50000	2.45s	346.47ms	5.53s	[a]

[a] PMU RAM exceeded

Correction Time:

Mode / Frames	R5 cache	R5 no-cache	PMU
1	10µs	28µs	14.1µs
10	16µs	48µs	18.99µs
100	91µs	220µs	67.41µs
1000	832µs	1.92ms	544.95µs
10000	8.26ms	18.91ms	5.33ms
50000	41.25ms	94.42ms	26.55ms

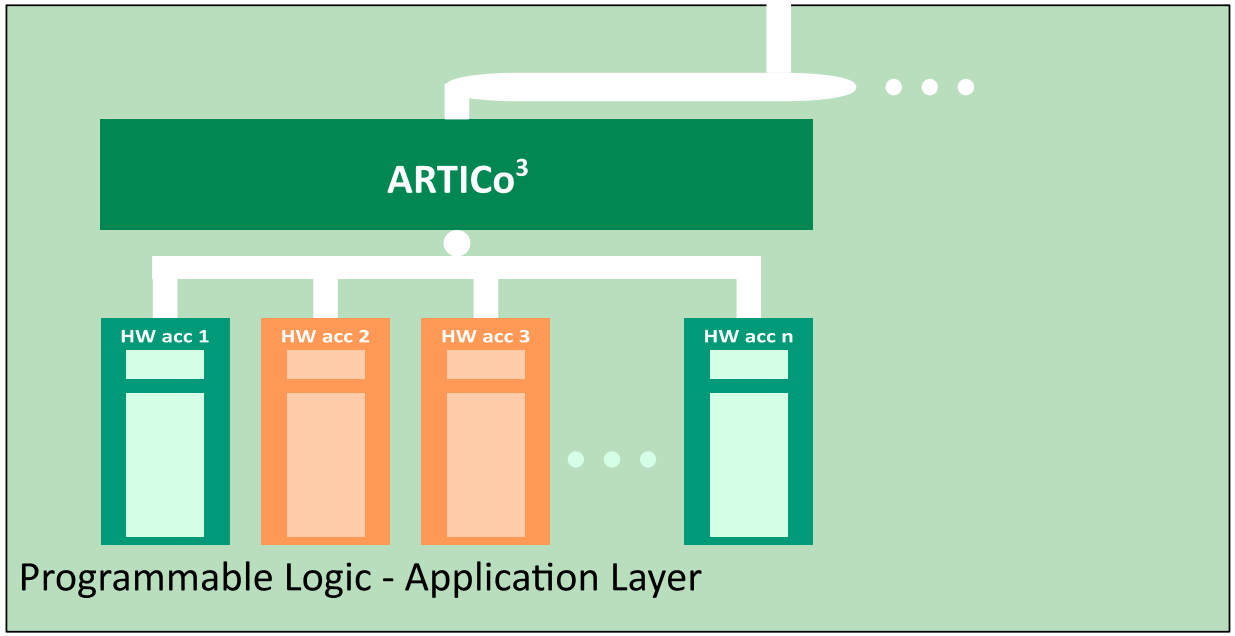
Reconfiguration time with PCAP frequency: 187.5MHz

Mode / Frames	R5 cache	R5 no-cache	PMU
1	15µs	35µs	16.91µs
10	33µs	77µs	34.67µs
100	238µs	489µs	213.28µs
1000	2.3ms	4.59ms	1.99ms
10000	22.88ms	45.61ms	19.85ms
50000	114.32ms	227.85ms	99.21ms

RECONFIGURATION TIME WITH PCAP FREQUENCY: 46.88MHZ

Configuration Aware Readback Scrubber

- When mixing **Reconfigurable** Architectures with **scrubbers**, there is not a single **golden copy** to compare with → bitstream composition or multiple file access?
  → Multiple file accesses preferable



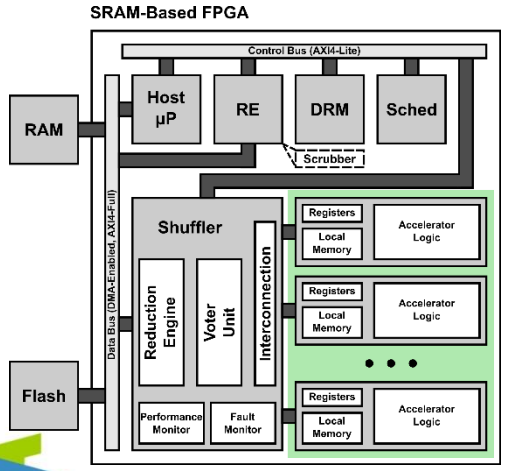
Configuration Memory

00000000	00000000	40000000
04000000	40000020	0000a010
00104000	01000400	10000010
4c000000	20111111	00010010
00000400	00F0A031	80010002
00020811	881010C0	00080002
00404000	06041240	0000a010
00104000	04501069	10000010
4c000000	00400101	00010010
00000400	80010002	80010002
00020811	00001000	00080002
00404000	00040040	0000a010
00104000	00001000	10000010
4c000000	40040000	80010010

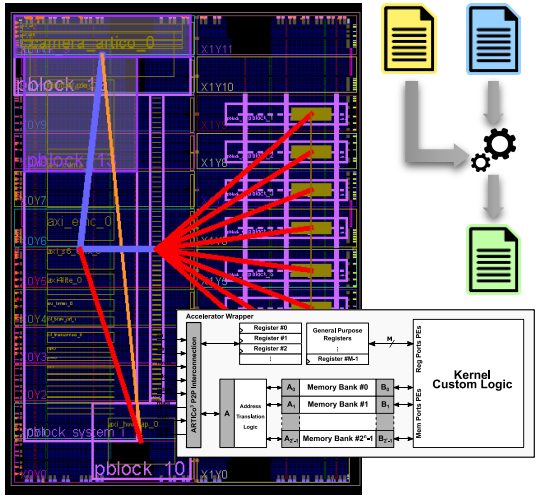
The ARTICo3 Framework

- ARTICo^3 is…
  - …a runtime reconfigurable architecture…
  - …for high-performance embedded computing…
  - …with adaptable fault tolerance and energy efficiency
- It has three components:
  - Processing architecture (hardware components)
  - Toolchain (design automation)
  - Runtime library (transparent use from host applications)

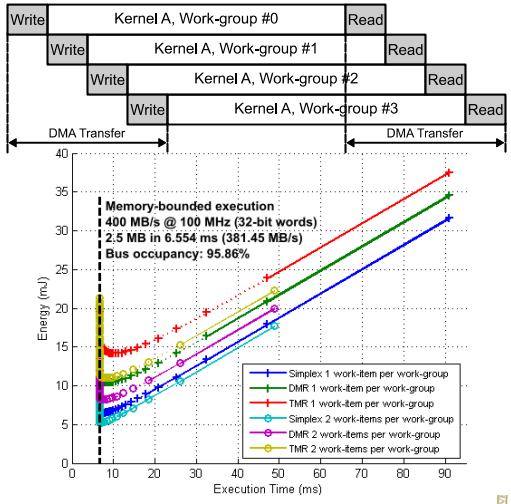
ARCHITECTURE



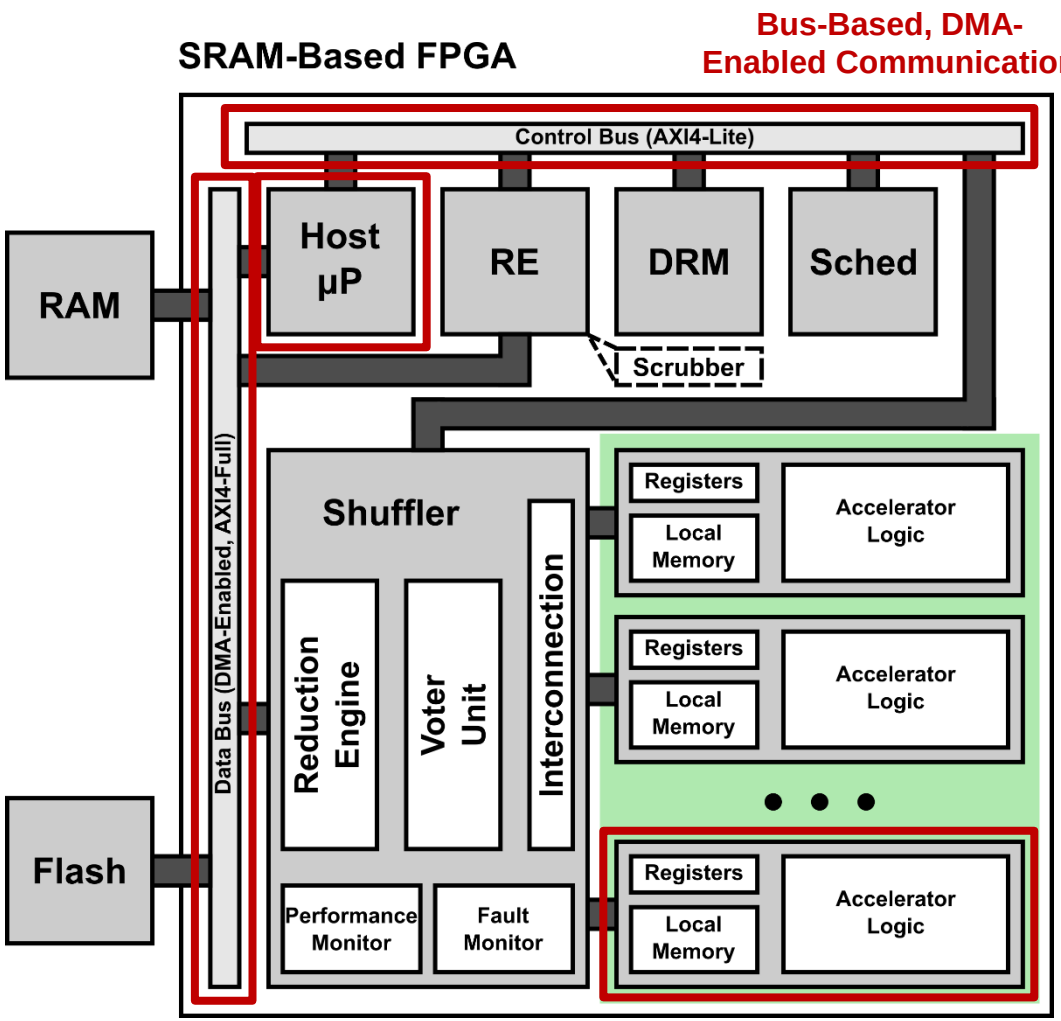
DESIGN FLOW



RUNTIME ENVIRONMENT

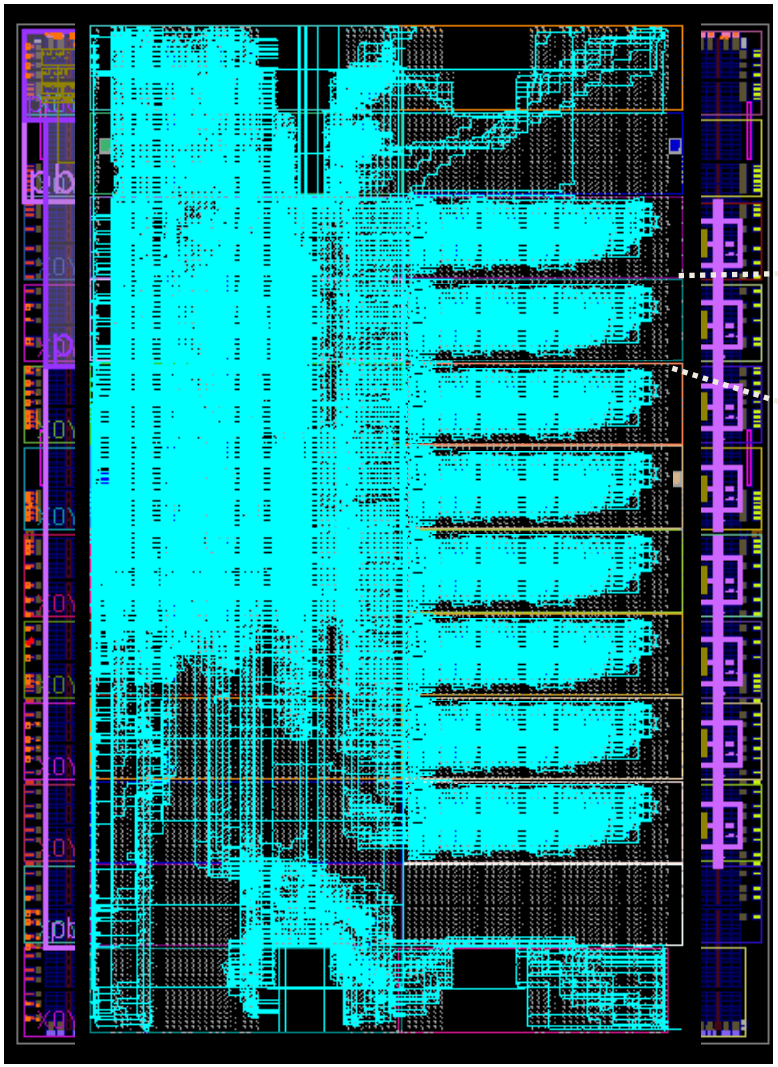


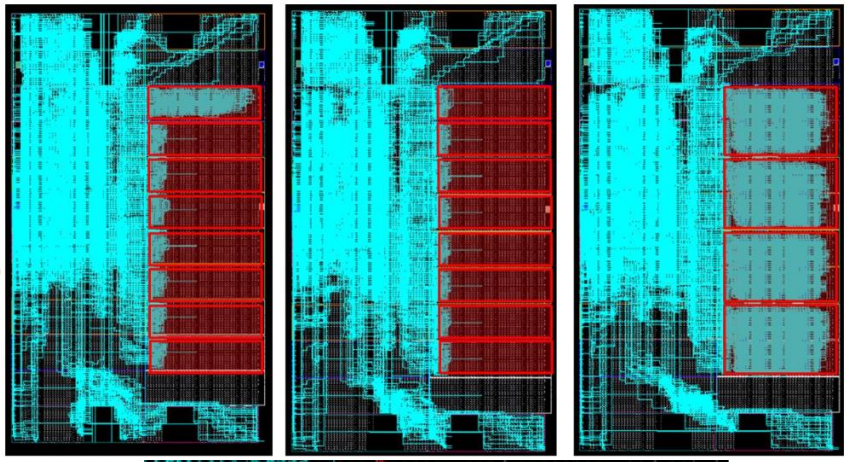
The ARTICo3 Architecture

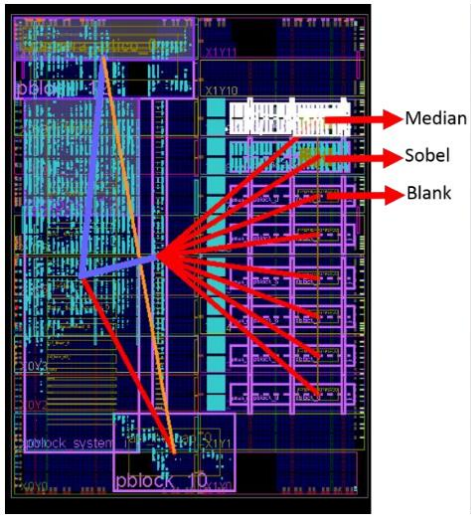


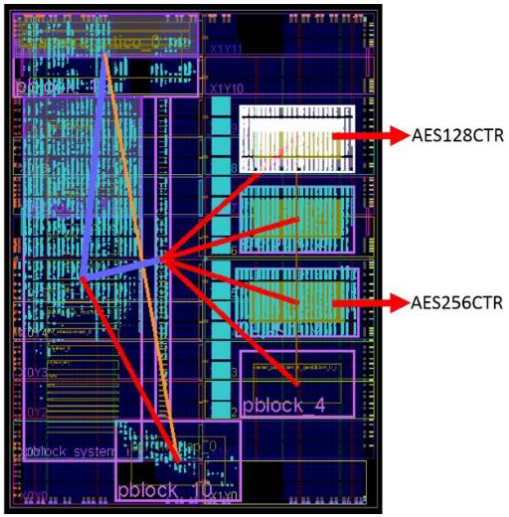
Reconfigurable
Architecture to enable
Smart
Management of
Performance
Energy Consumption
Dependability

ARTICo3: Some Implementations







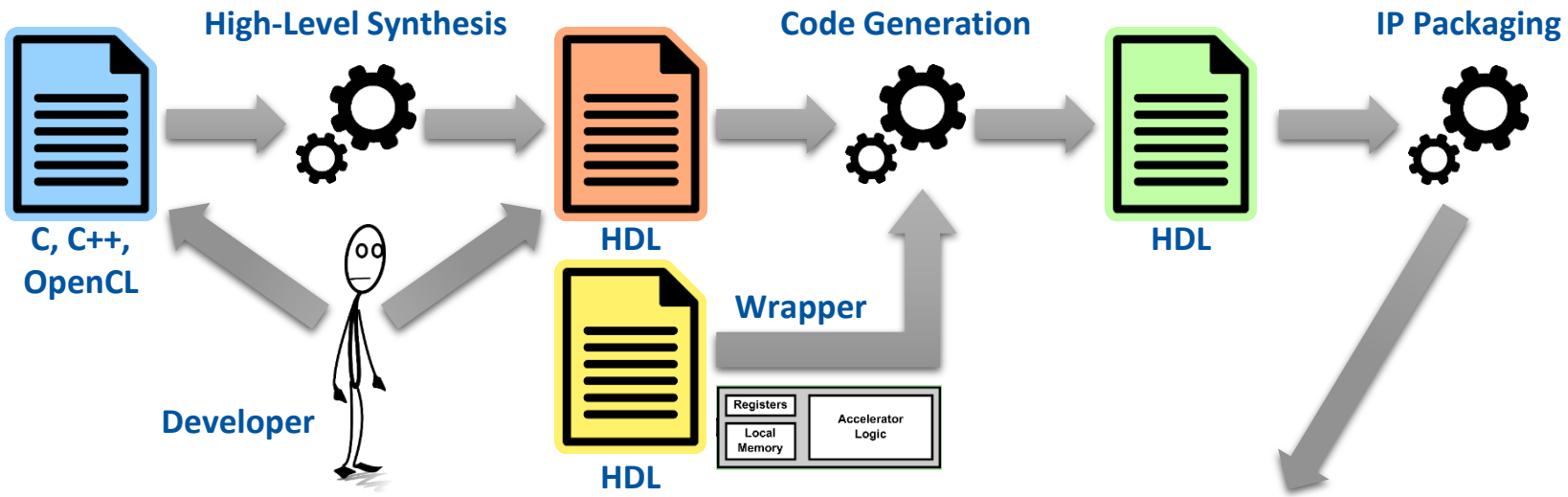


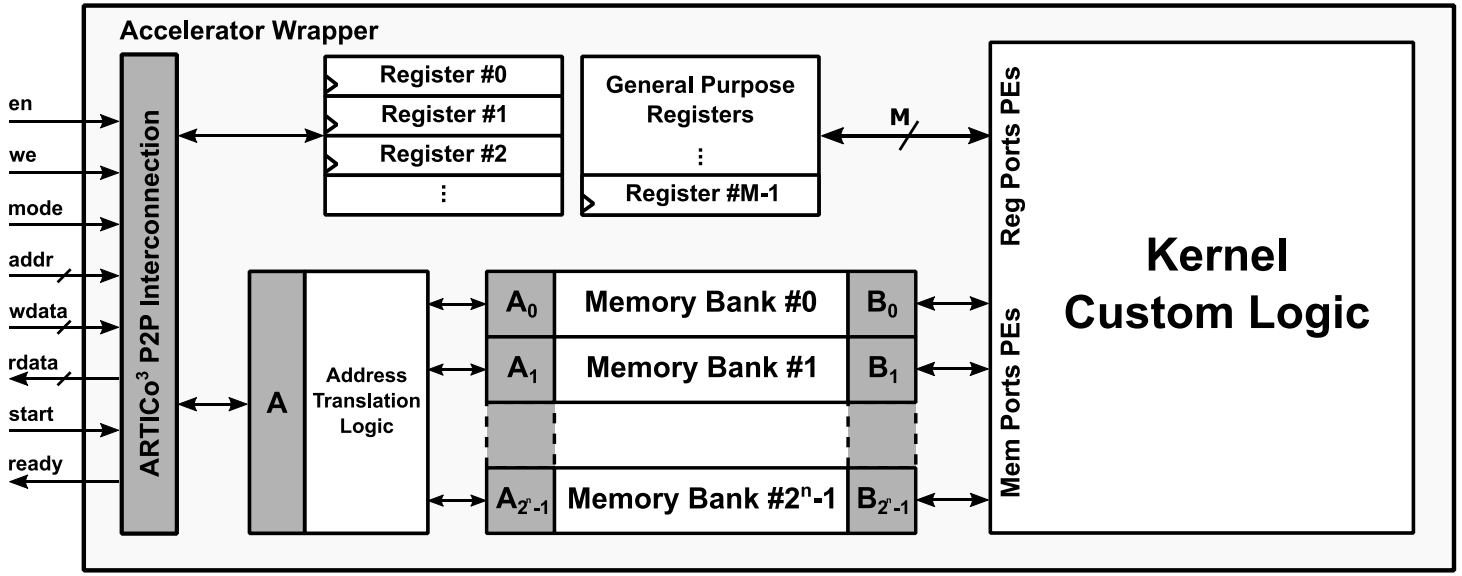
Spartan-6 LX150 (HiReCookie)
PlanAhead & FPGA Editor

CEI UPM

POLITÉCNICA

ARTICo³-Compliant Accelerator Design





Dynamic Solution Space Exploration

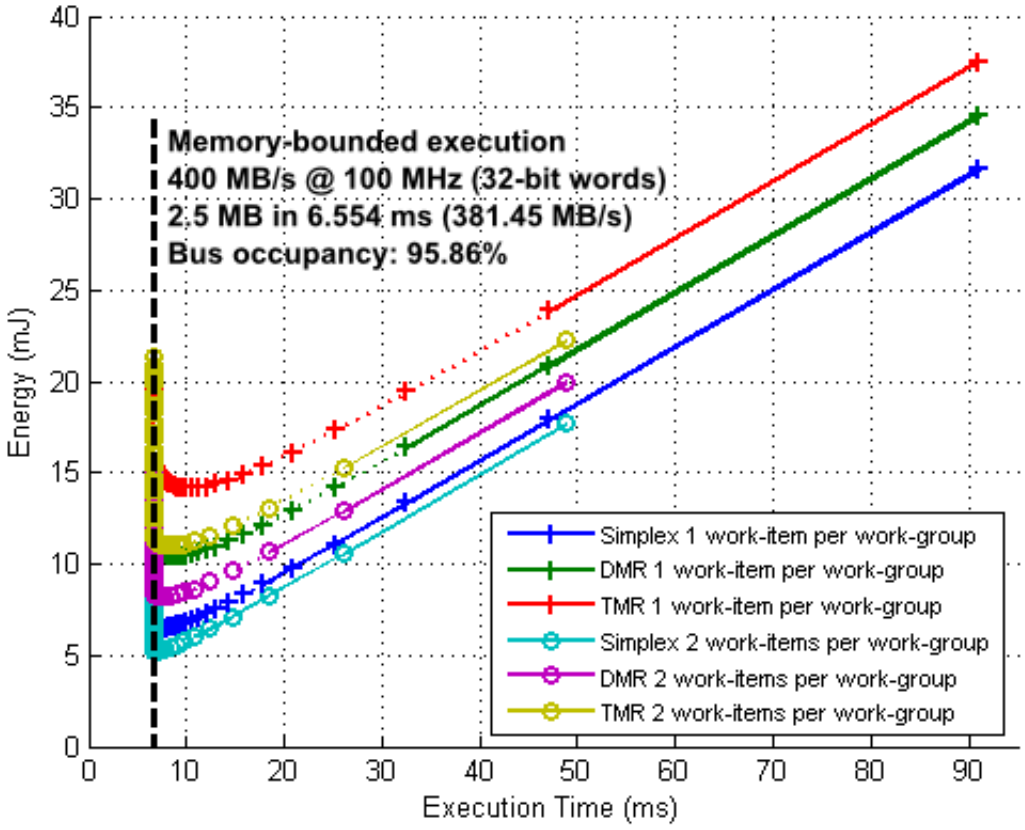
Embedded Models

+

Runtime Monitoring & Profiling

=

Self-Aware Online Resource Management

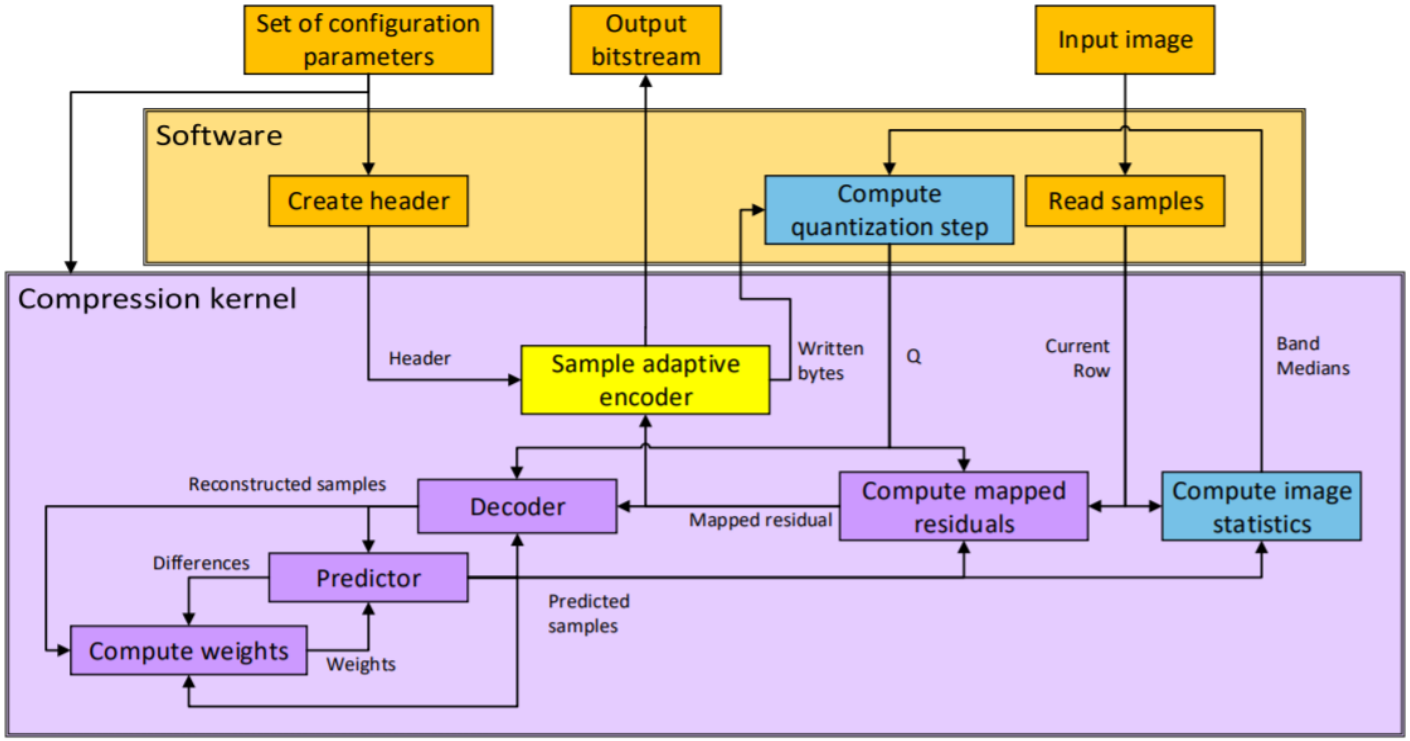


The ARTICo3 Projects echosystem

REBECCA National funding 2015-2017	Basic ARTICo3 architecture and modelling Extension to multi-FPGA context → Increased acceleration GPU-like model of computation Use case: Smart cities with resilient high-performance sensor nodes
Enable-S3 EU funding (ECSEL) 2016-2019	Hardening the basic architecture Combination with real-time operating systems (RTEMS) Use cases: Hyperspectral image compression (Thales Allenia Space + ULPGC) Camera-based satellite navigation system (GMV)
CERBERO EU (H2020-IC-1) 2017-2019	Toolflow integration Dataflow model of compuutation Combination of fine-grain HW composition, coarse-grain and ARTICo3 Use case: robotic arm controller for a martian rover
PLATINO National funding 2018-2020	Toolflow integration ARTICFo3 as a container for machine learning HW acceleration Exchangeable HW overlays for diverse ML algorithms Use case: farming application with drones + IoT + VR

Application: CCSDS123 Lossy Extension

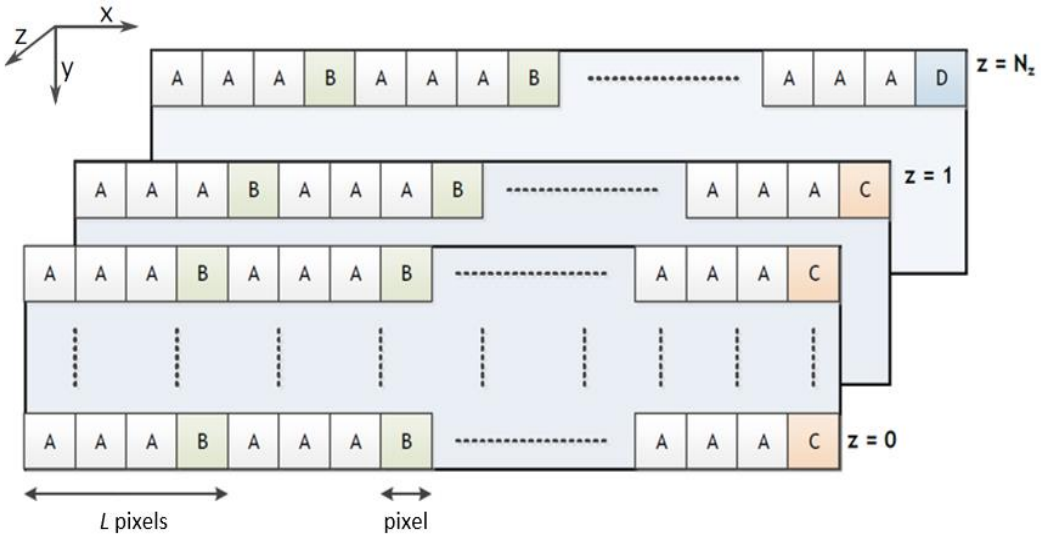
- Extends the CCSDS 123.0-B-1 lossless compression algorithm, working in a **near-lossless to lossy** range*.
- Able to adapt losses according to the user-selected bit rate (rate control).
- The **quantizer** is able to calculate the suitable quantization step for the next spectral line taking into account the desirable compression ratio specified by the user.
- A HW/SW partition has been performed according to the tasks complexity, taking advantage of an MPSoC implementation.



*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.

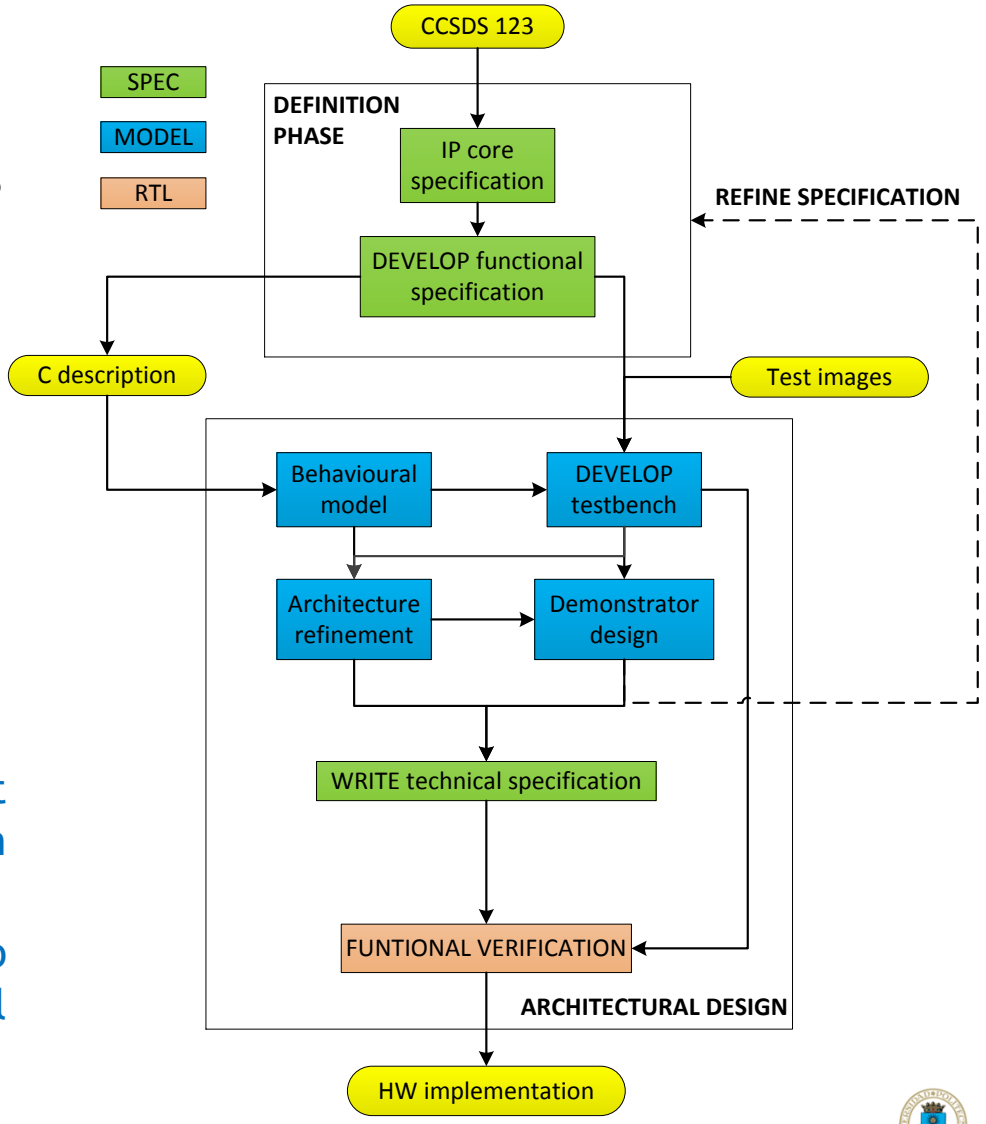
Bit rate control

- Hardware-friendly description, simplifying the algorithm complexity and reducing the latency.
- A unique quantization step is applied to each spectral line.
- The calculation is done considering that the variance of the prediction residuals between two adjacent lines are highly correlated.
- A **median** is computed for each band, and after all the medians have been obtained, the quantization step for the next line is computed.



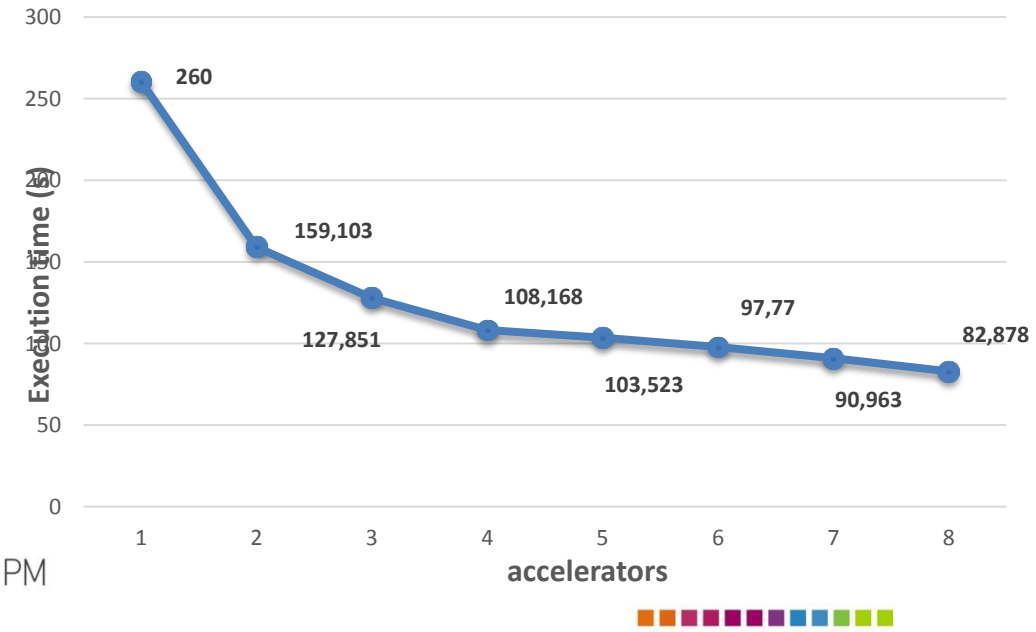
HLS design flow

- CCSDS-123 lossy extension modelled in C and directly transformed into RTL using HLS tools.
- Implementations by automated tools (Xilinx Vivado HLS).
- C reference code from ESA has been adapted for an efficient hardware implementation.
- Advantages of HLS design:
  - Minimal design at RTL level.
  - Untimed simulation for hardware functional verification.
  - Reduced Time-to-Market.
  - Fast exploration of different architectures and parallelization approaches.
  - Reduced design time, returning to previous steps without additional costs.



Execution results within ARTICo3

- Totally dependent on the number of hardware accelerators running over the ARTICo^3 architecture.
- The use of multiple accelerators is intended to split the hyperspectral images into portions, distributing them among the different accelerators → exploit parallelism.
- Software latency running on an ARM Cortex-A53 → around 560 s.
- Maximum speed up **x7** when 8 accelerators are instantiated.



NEW! Best results:

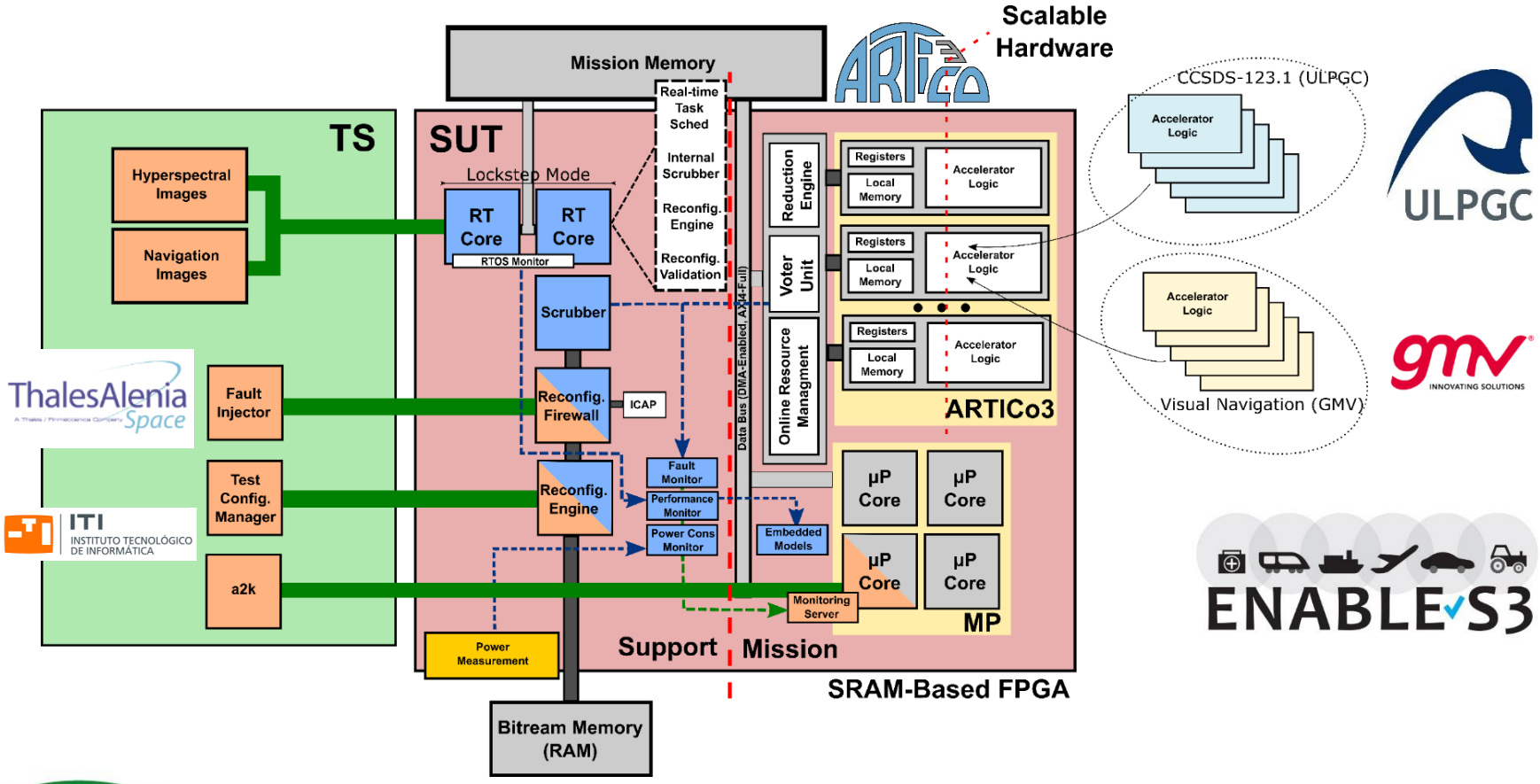
35 s for 512x512x256
7 s for 64x256x256
7,5 s for 6 4x512x256
6,5 s for 32x256x256
9 s for 256x256x256
4 s for 128x128x256

CEI UPM

POLITÉCNICA

Conclusions: Hardening MPSoCs for Space Applications?

Implementing **On-board Processors** for Space applications on **reconfigurable**,
non rad-hard, SRAM-based COTS FPGA (Zynq Ultrascale+).



RTEMS ported to the Zynq Ultrascale+ Cortex-R5 processors, running in lockstep mode.

CEI UPM | Centro de Electrónica Industrial

Thank you very much!

Questions?

Contact: eduardo.delatorre@upm.es

cei@upm.es

POLITÉCNICA