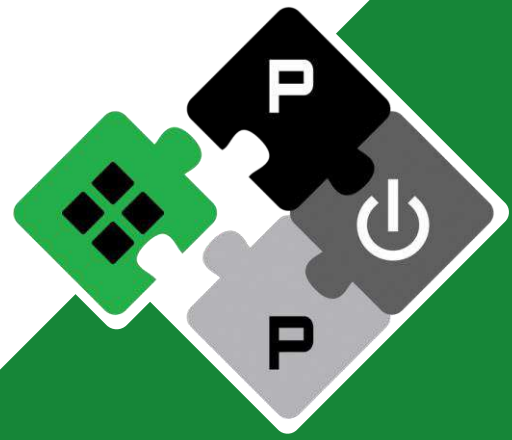
One Small Step for Man, One Giant Leap for PULP

PULP for Space Applications

10 Years of PULP

Yvan Tortorella yvan.tortorella@unibo.it

Michael Rogenmoser michaero@iis.ee.ethz.ch



@pulp_platform

pulp-platform.org

youtube.com/pulp_platform

PULP Platform

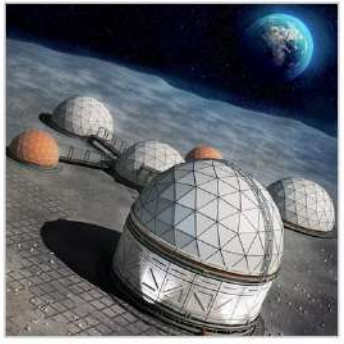
Open Source Hardware, the way it should be!

Why put PULP in Space?

Increasing demand for strong processing capabilities in space

- Image processing
- On-board orbit determination and reconfiguration
- Fault Detection, Isolation and Recovery (FDIR)

Increasing Demand for Processing in Space



January 30, 2023

NASA Recruits Microchip, SiFive, and RISC-V to Develop 12-Core Processor SoC for Autonomous Space Missions

by Steven Leibson

NASA's JPL (Jet Propulsion Lab) has selected Microchip to design and manufacture the multi-core High Performance Spaceflight Computer (HPSC) microprocessor SoC based on eight RISC-V X280 cores from SiFive with vector-processing instruction extensions organized into two clusters, with four additional RISC-V cores added for general-purpose computing. The project's operational goal is to develop "flight computing technology that will provide at least 100 times the computational capacity compared to current spaceflight computers." During a talk at the recent RISC-V Summit, Pete Fiacco, a member of the HPSC Leadership Team and JPL Consultant, explained the overall HPSC program goals.

RISC-V®

IN SPACE

14 DEC 2022
ESTEC
ERASMUS
AUDITORIUM

Why put PULP in Space?

Increasing demand for strong processing capabilities in space

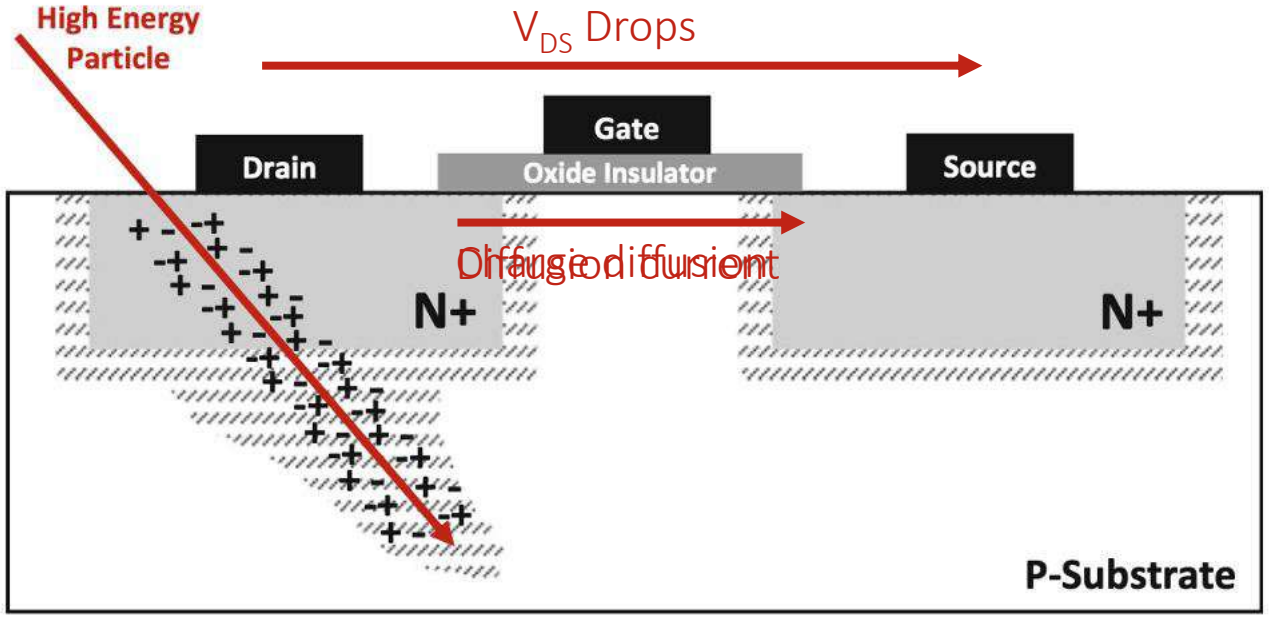
- Image processing
- On-board orbit determination and reconfiguration
- Fault Detection, Isolation and Recovery (FDIR)

Cosmic rays badly affect on-board computers

- Single Event Upset, Single Event Transient, Transient Faults (TF)
- Cannot be avoided, only attenuated!

Problems in Space?

- High Radiation Levels
- Cause Transient errors in transistors
- Can disrupt computation and crash the system.



Source: Bolchini et al.: "Dependability threats" in "Dependable Multicore Architectures at Nanoscale", Springer International Publishing, 2018, p. 53.

Why put PULP in Space?

Increasing demand for strong processing capabilities in space

- Image processing
- On-board orbit determination and reconfiguration
- Fault Detection, Isolation and Recovery (FDIR)

Cosmic rays badly affect on-board computers

- Single Event Upset, Single Event Transient, Transient Faults (TF)
- Cannot be avoided, only attenuated!

If we cannot get rid of transient faults, we must tolerate them!

How to deal with TFs?

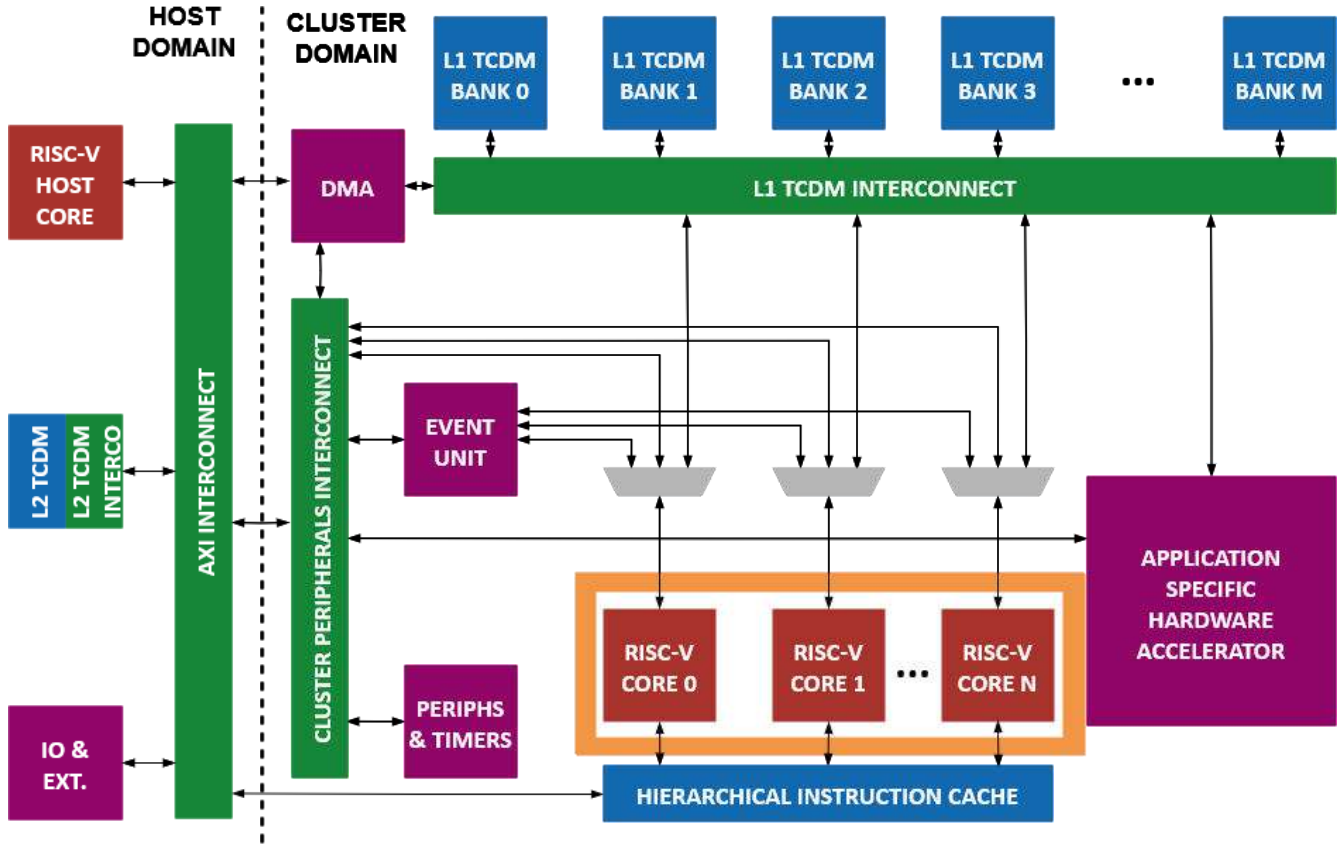
- Software approaches: easy, but time consuming (bad performance)
- Hardware approaches: good performance, require open platforms

PULP: Open-source, open HW/SW development!

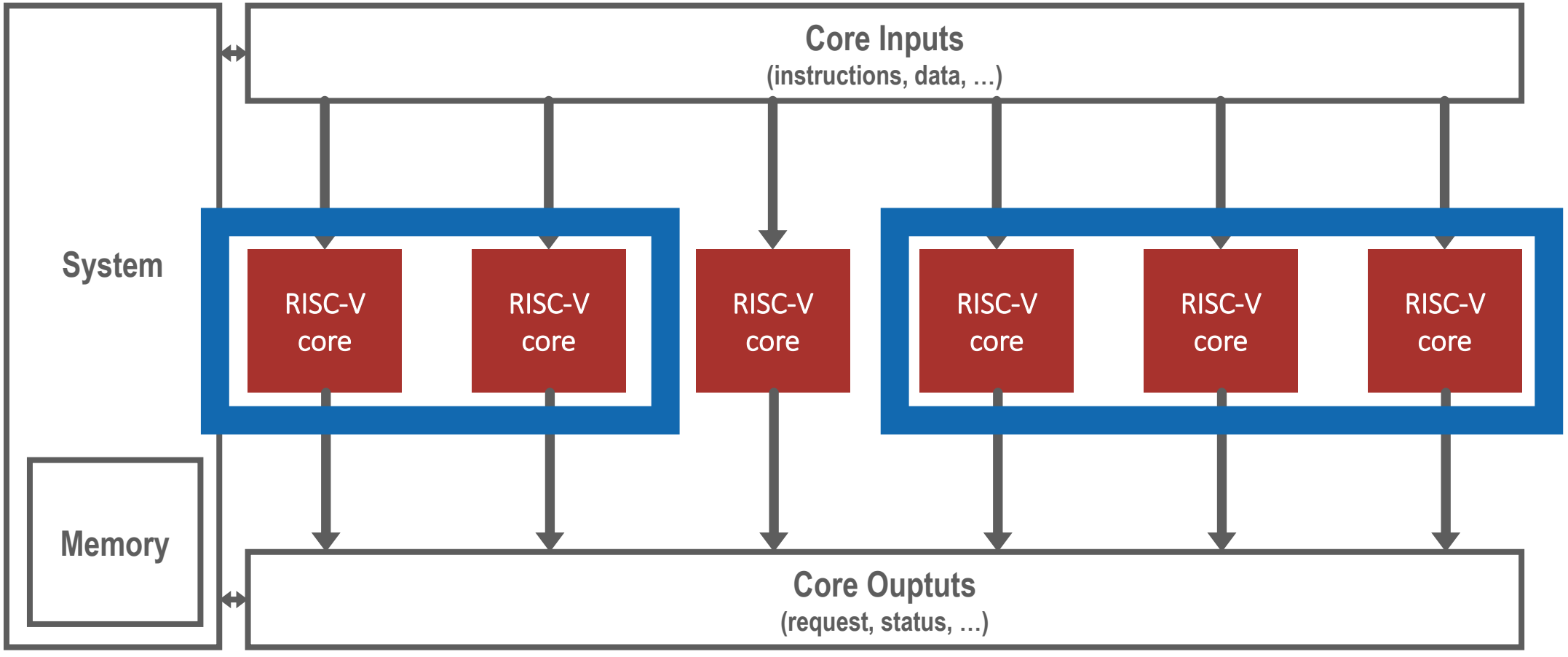
At this point we all know what PULP is…

- Parallel Ultra-Low-Power
- Cluster Domain
  - Efficient shared-mem cluster
  - From a few to thousand processing RISC-V cores
  - Support for tightly-coupled specialized accelerators
- Host Domain
  - Single-to-Multi core domain for decision making
  - Interface to external IOs and peripherals
  - Support for tightly-coupled specialized accelerators

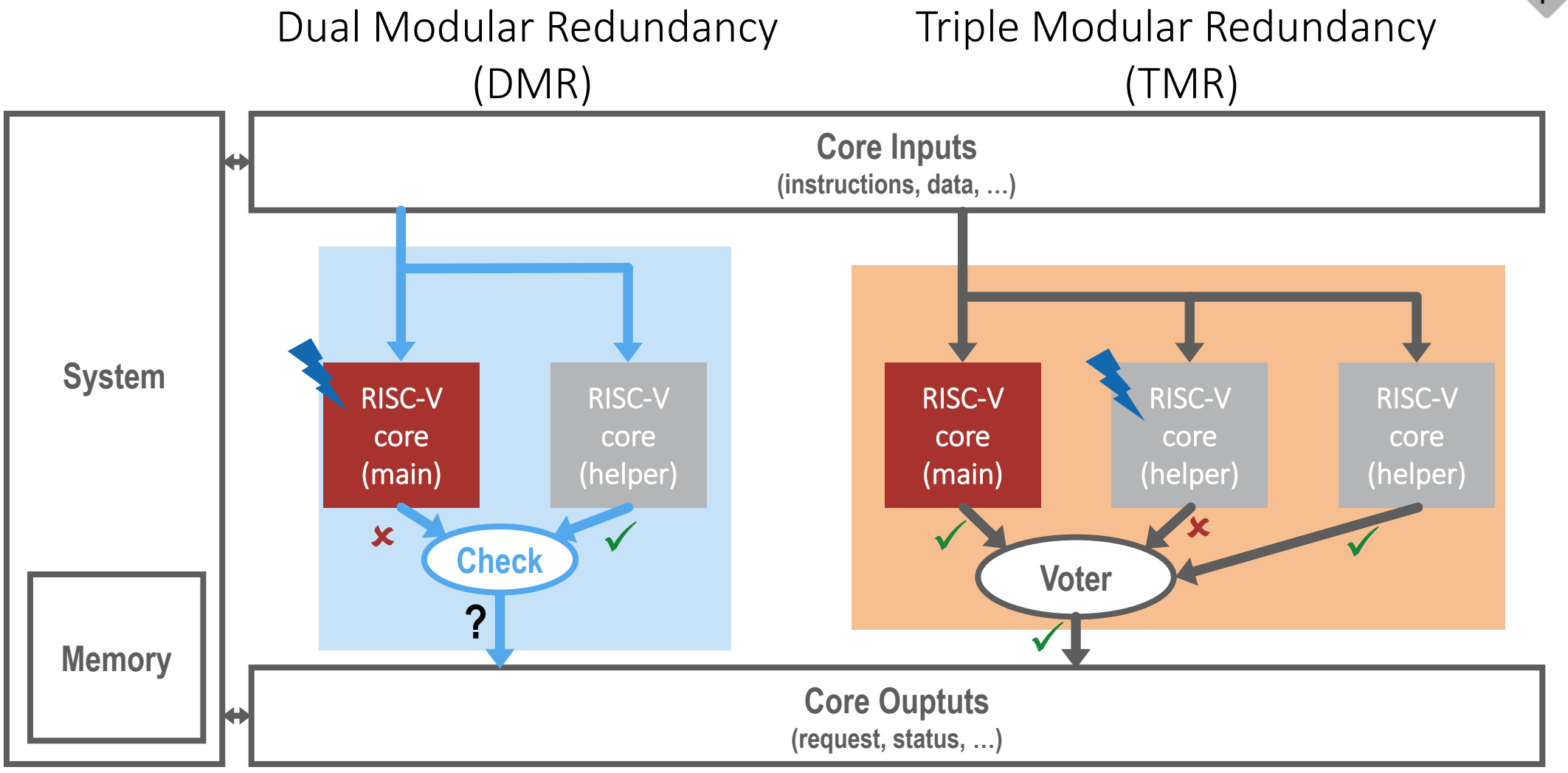
But how do we protect it?



Protecting the Cores: Redundant Grouping



Redundant Grouping: how to choose?



Sacrificing performance and area

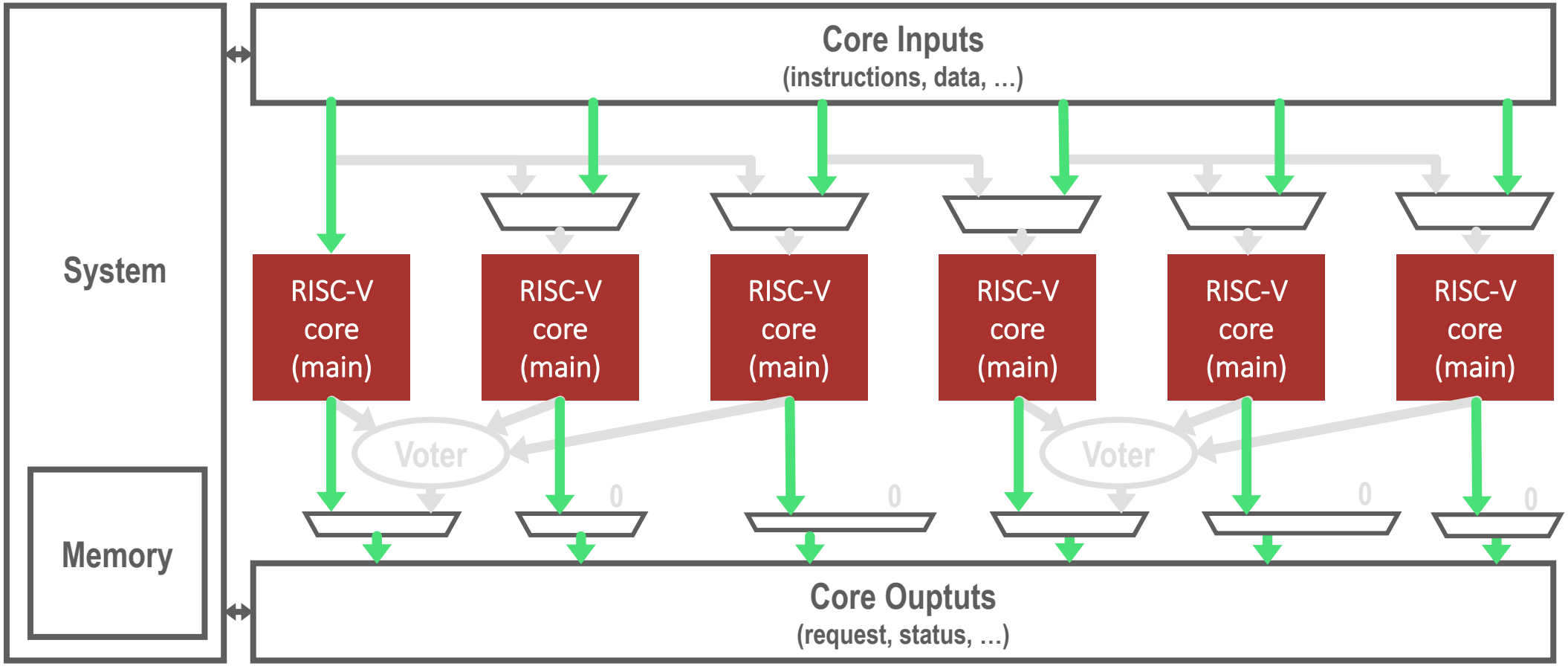
Maybe not all the computing tasks in space have the same level of criticality

Maybe we are not even in space when we start computing…

We strive for flexibility! → Reconfiguration: **Hybrid Modular Redundancy (HMR)**

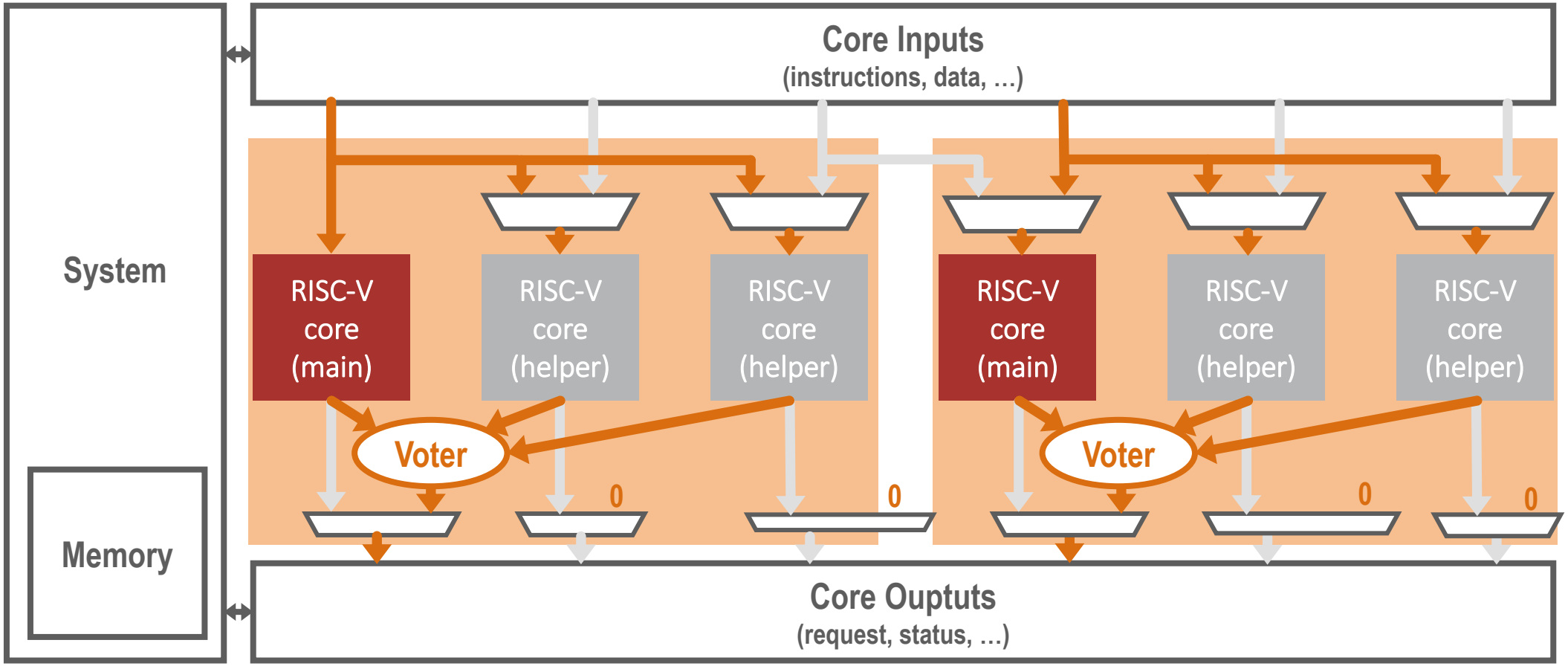
Hybrid Modular Redundancy (HMR): Reconfigurable

Independent Mode: high performance, no reliability



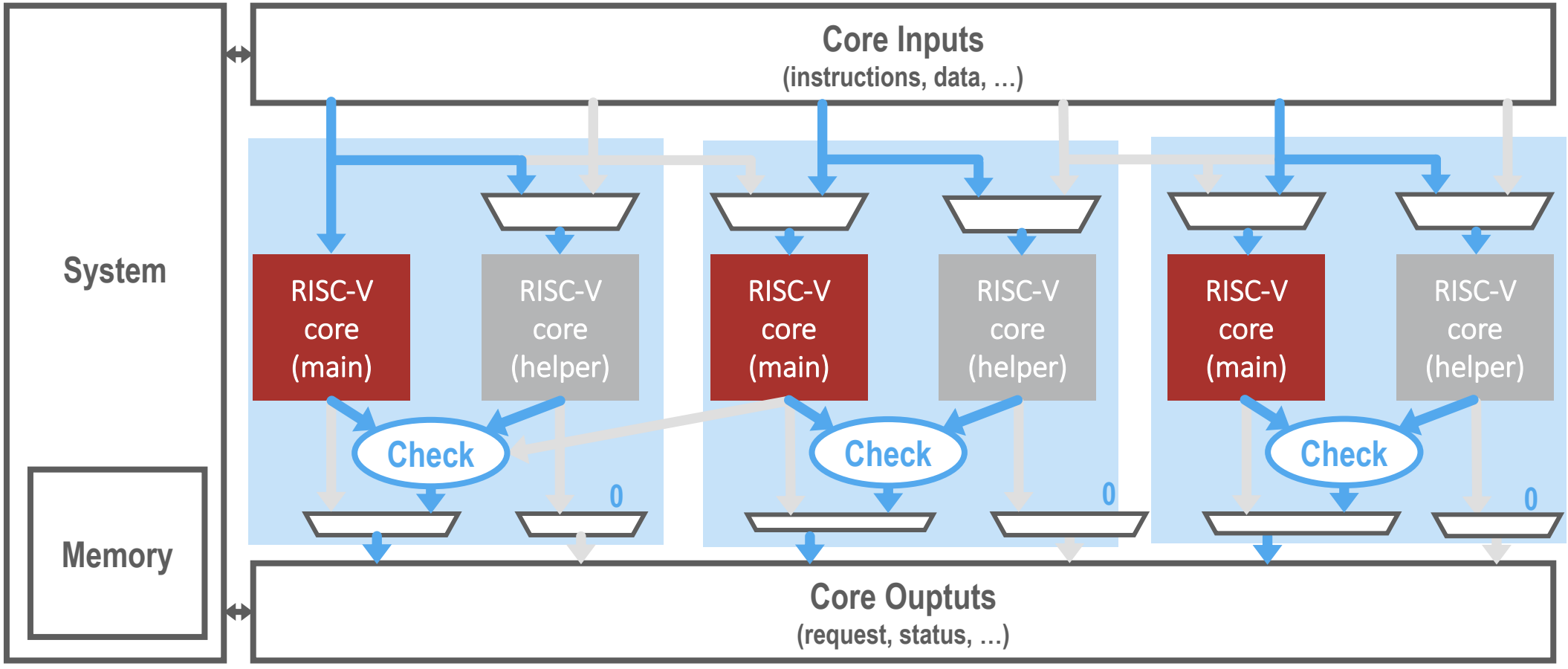
Hybrid Modular Redundancy (HMR): Reconfigurable

TMR Mode: low performance, high reliability, quick recovery



Hybrid Modular Redundancy (HMR): Reconfigurable

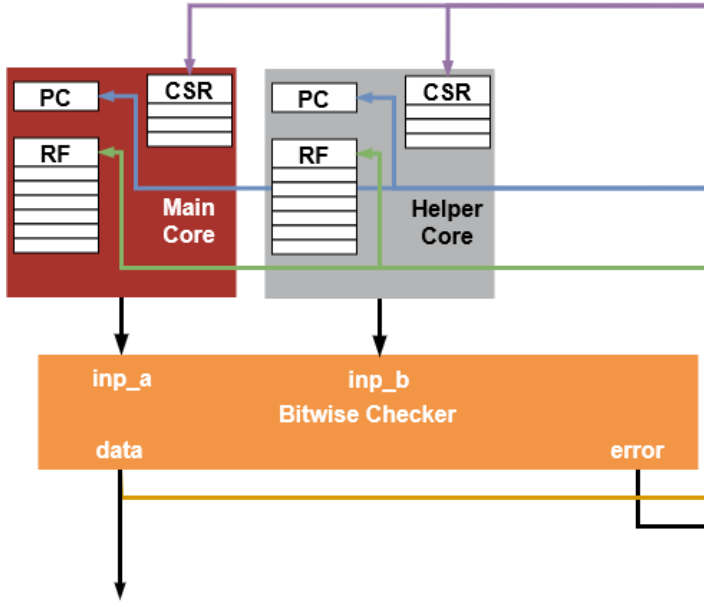
DMR Mode: good performance, good reliability, slow recovery



Rapid Recovery: shared hardware extension

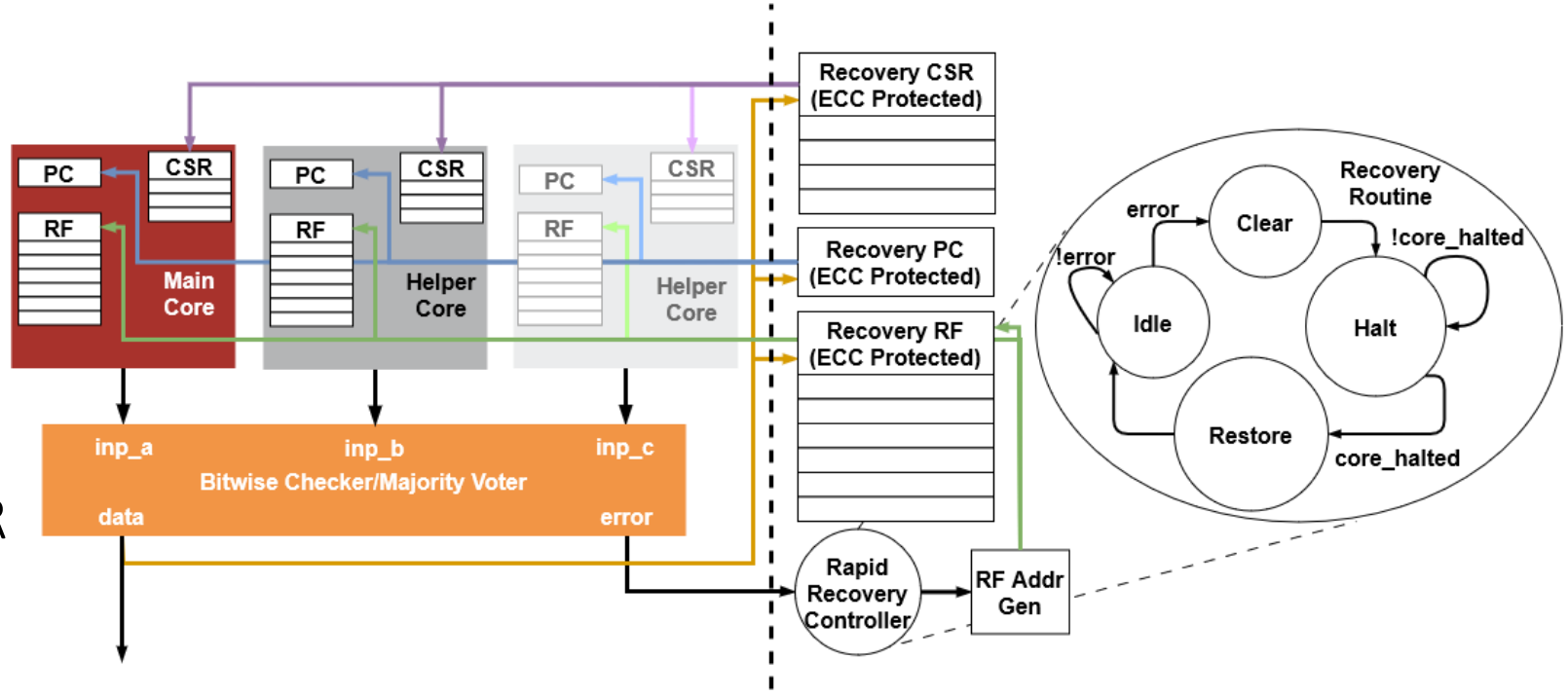
Rapid recovery mechanisms require **dedicated hardware** extensions

- The **state** of the core is defined by its internal registers (**CSRs, RF, PC**)
- Additional **backup copies** of the status registers, protected by **ECC**
- **Cycle-by-cycle backup** -> allow for recovery to the **most recent safe state**
- Quick recovery routine (**24 cycles**!)



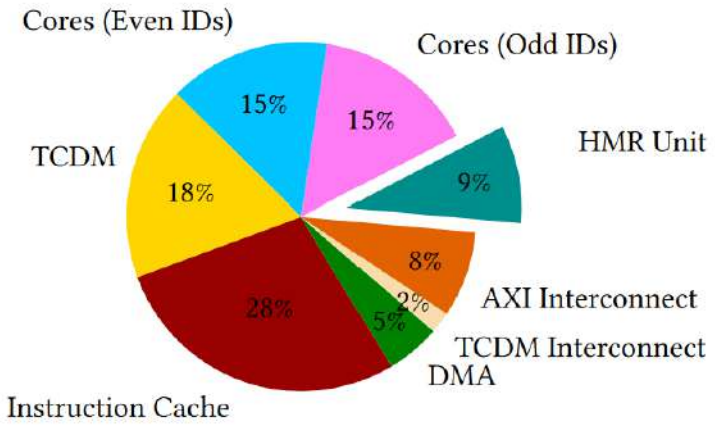
Rapid Recovery: shared hardware extension

- **Cycle-by-cycle backup** of the cores state in **ECC**-protected Status Registers
- Quick recovery procedure (**24 cycles**!)
- **Shared logic** between TMR and DMR modes

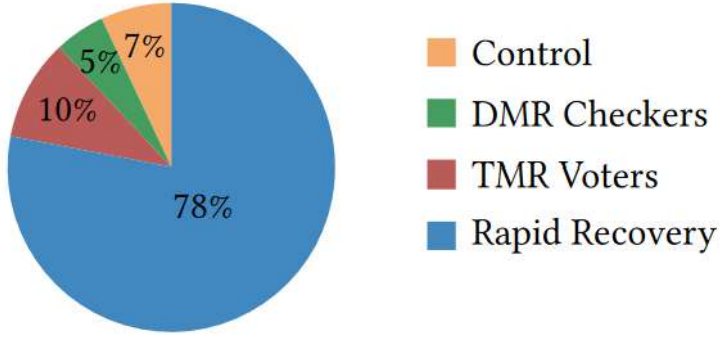


HMR, yes… but at what cost?

Cluster Area breakdown with HMR Unit



HMR Unit Area Breakdown



Area Overhead of HMR Configurations

PULP Cluster Area [mm^2]		Overhead
Baseline	0.604	-
DMR	0.605	0.3%
TMR	0.608	0.7%
HMR	0.612	1.3%
With Rapid Recovery		
DMR	0.654	8.4%
TMR	0.657	8.8%
HMR	0.660	9.4%

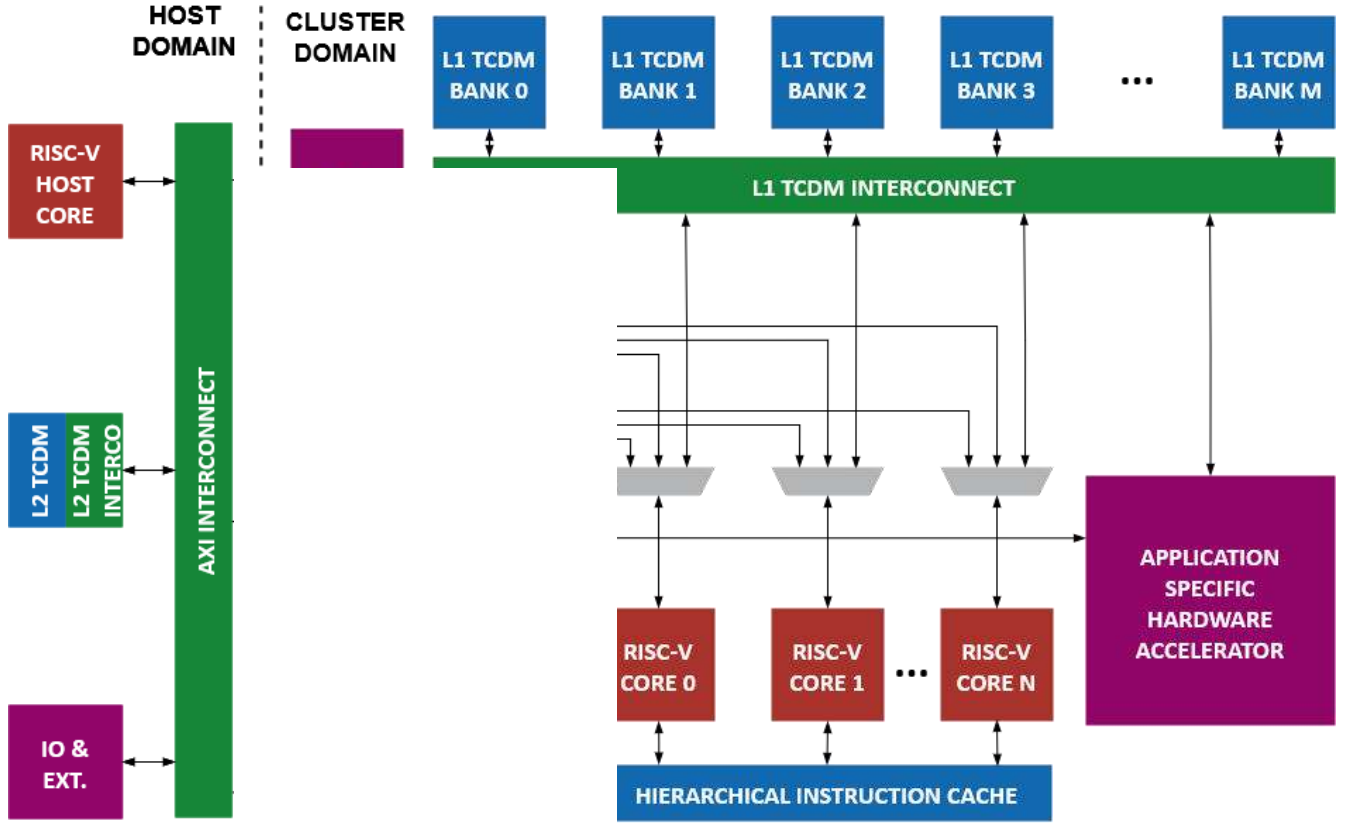
HMR Unit Recovery and Switching Mode Latency

	DMR	TMR	DMR Rapid Recovery	TMR Rapid Recovery
Recovery Latency [cycles]	Application dependant	363	24	24
Mode Switching [cycles]	703	598	603	515

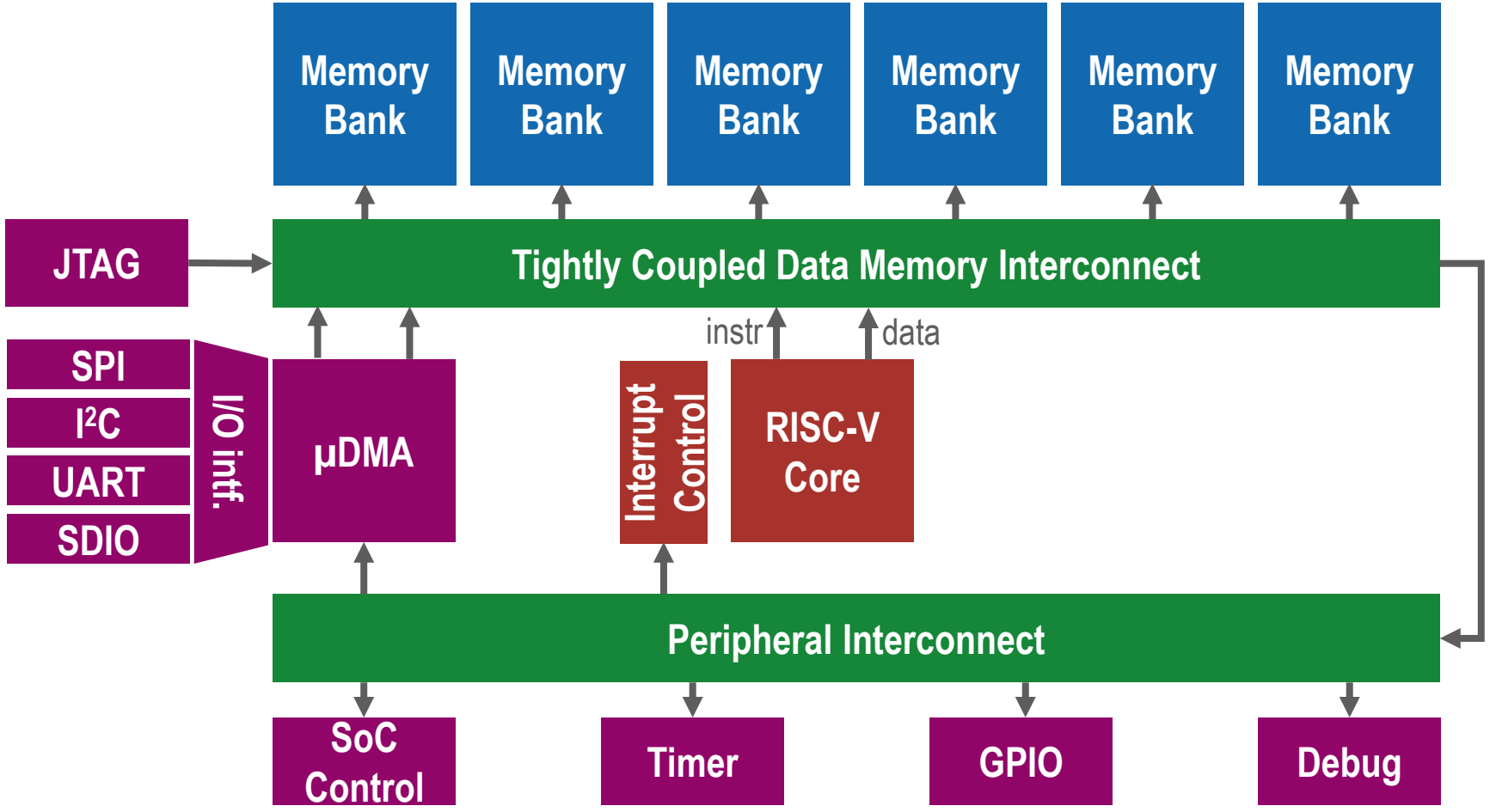
We do not take care only of multicore clusters

Many decisional tasks are extremely critical and must be executed safely

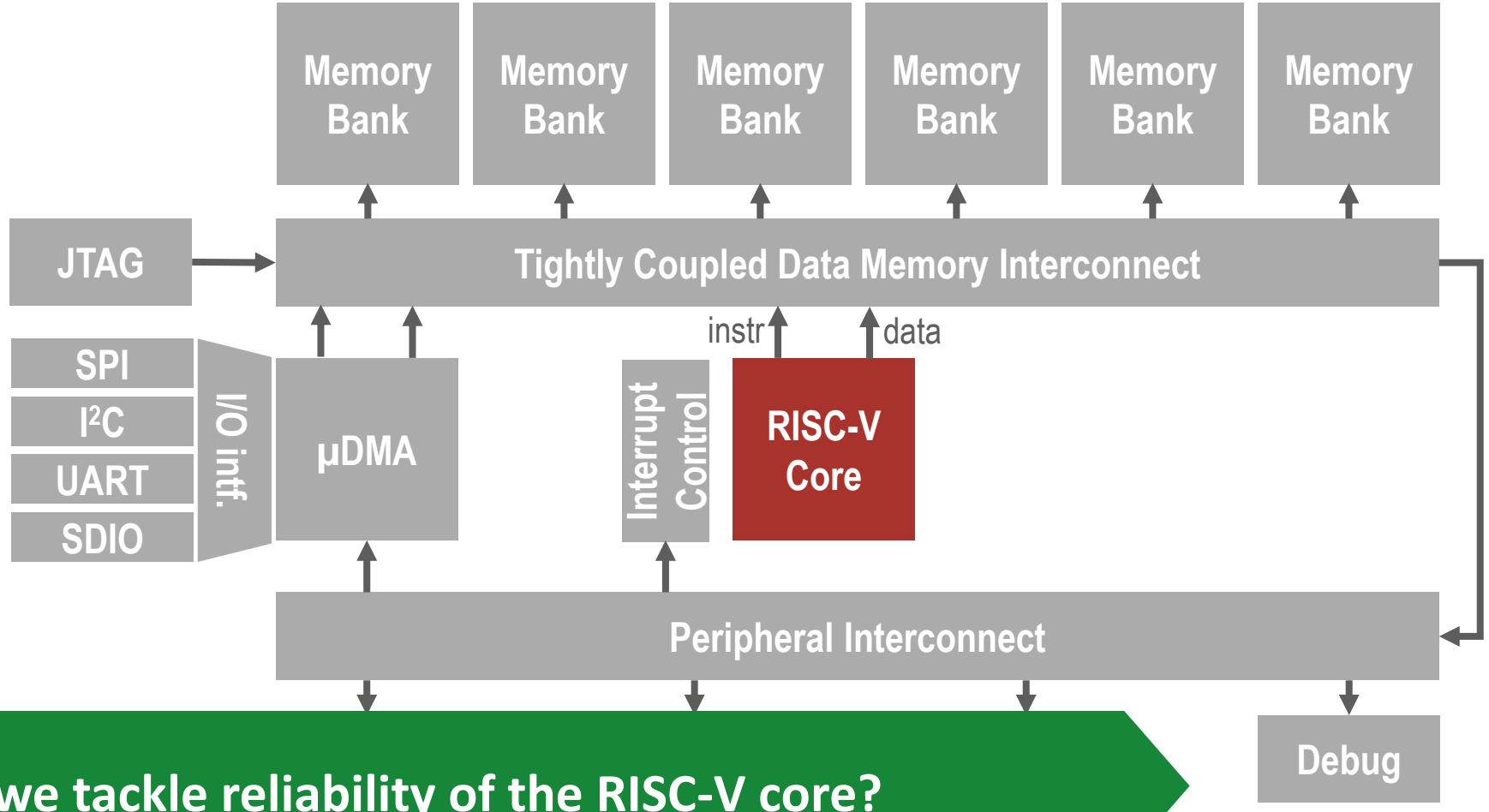
The Host domain also requires dedicated protection schemes



PULPissimo Architecture

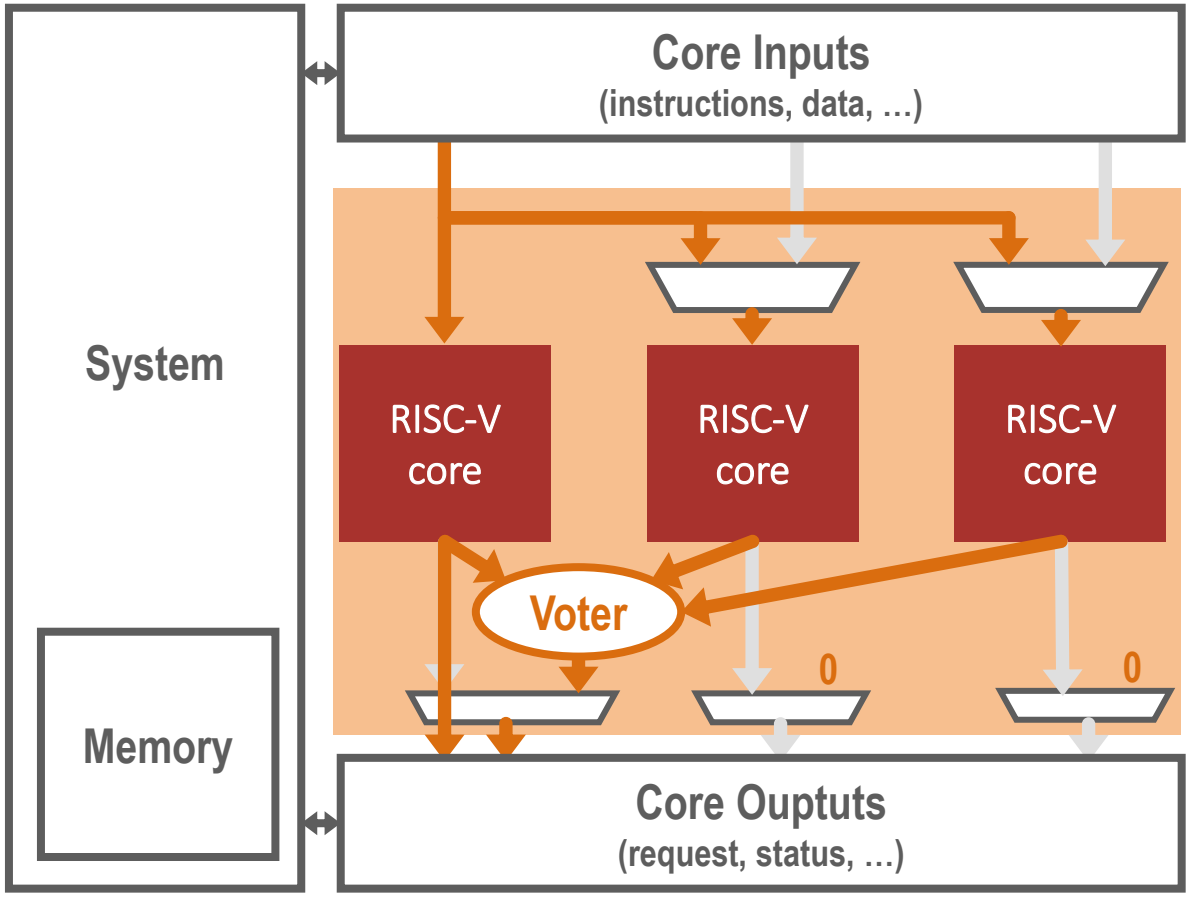


PULPissimo Architecture



How do we tackle reliability of the RISC-V core?

Reliable Processing Cores

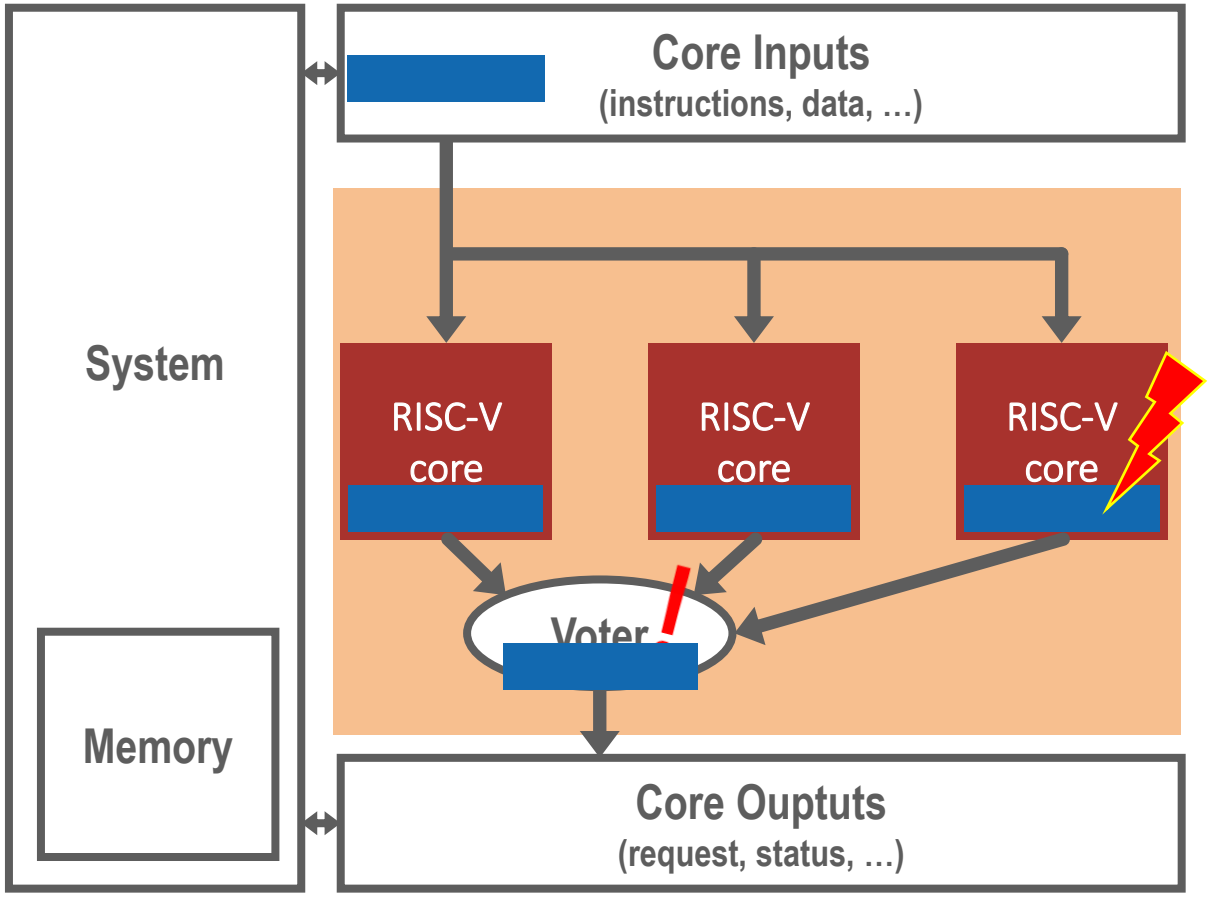


- Replicate Core

  →Triple-Core Lockstep

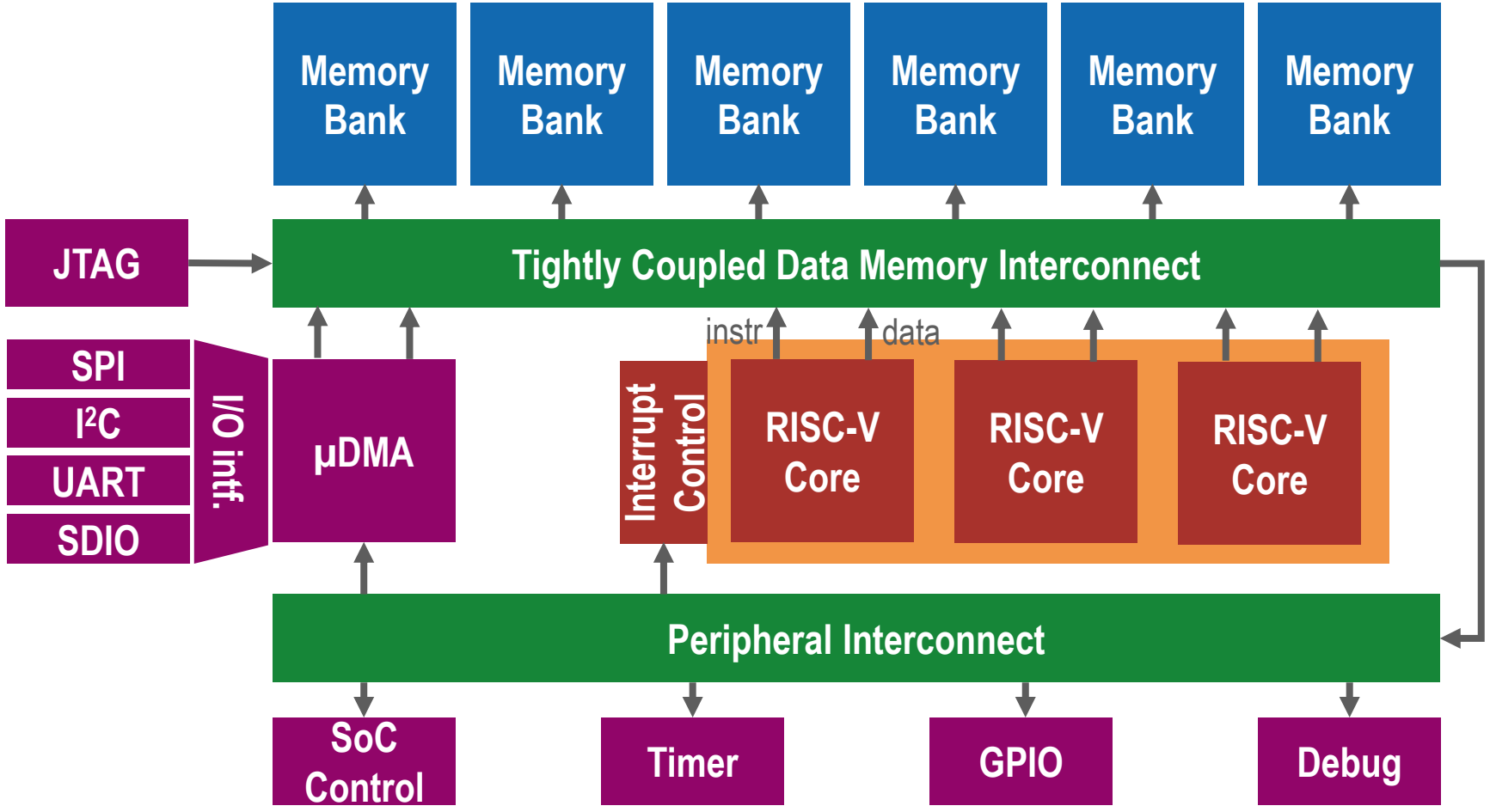
- Identical Inputs
- Voted Outputs
  - Directly connects any soft error
- Configurable for performance if reliability is not needed
  - **2.96x speedup**

Software Recovery of Triple Modular Redundant Cores

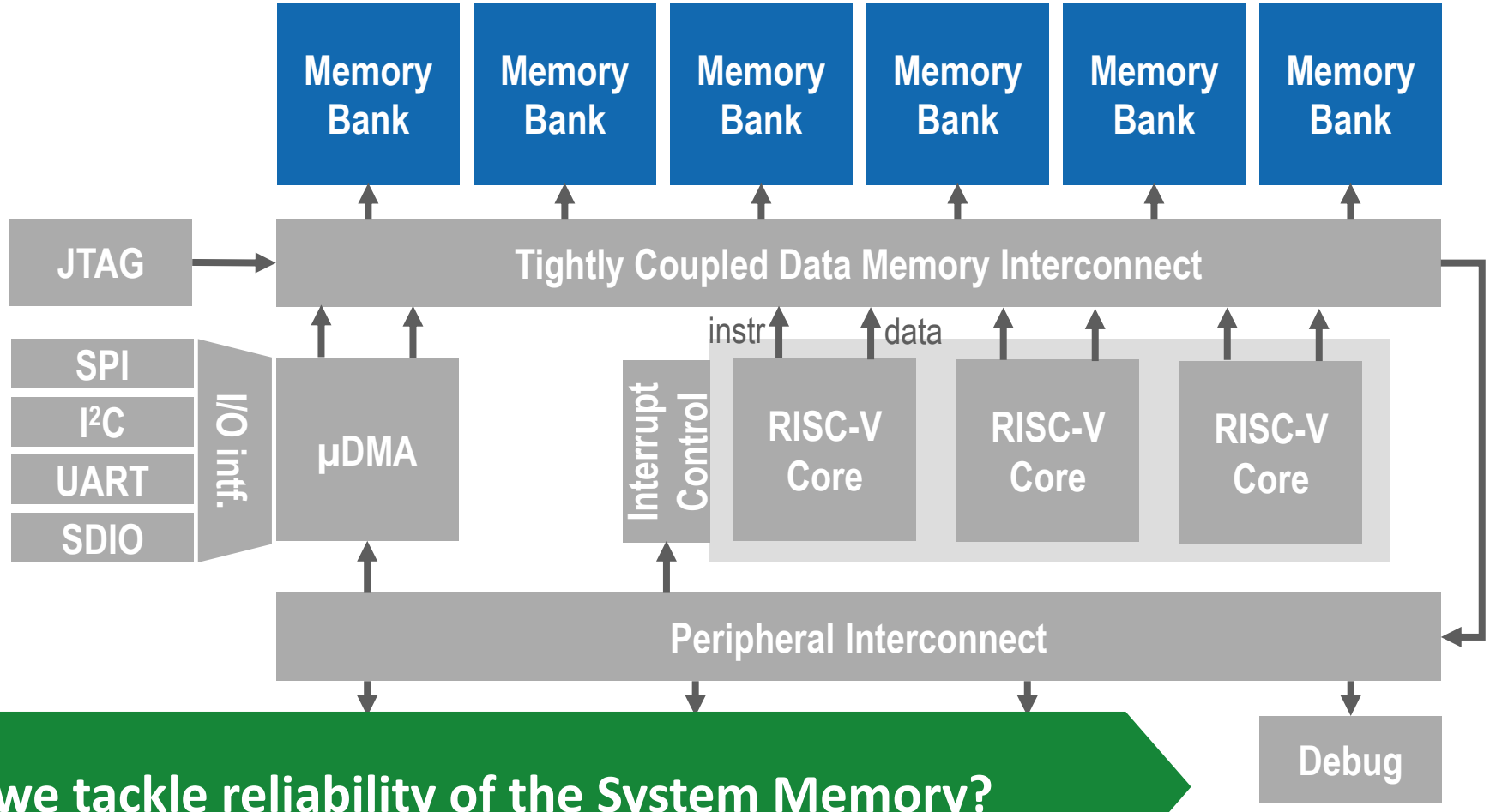


- Radiation causes soft-error
- Error detected by voter
- Core state (RF, PC, CSRs) stored to memory
  - Corrected by voter
- Core state loaded back into cores
- Total procedure in ~600 cycles

Trikarenos – PULPissimo with Reliability

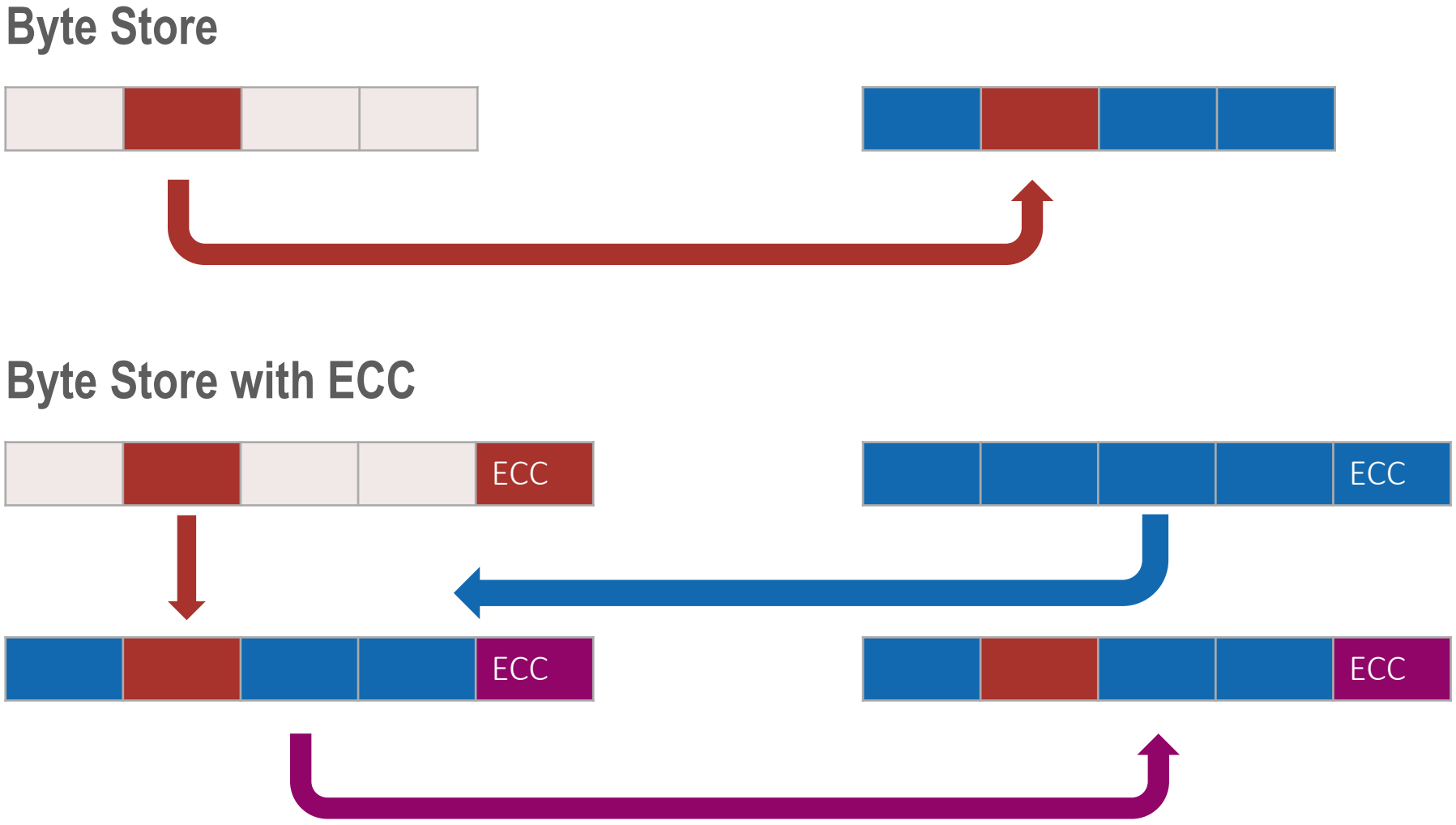


Trikarenos – PULPissimo with Reliability

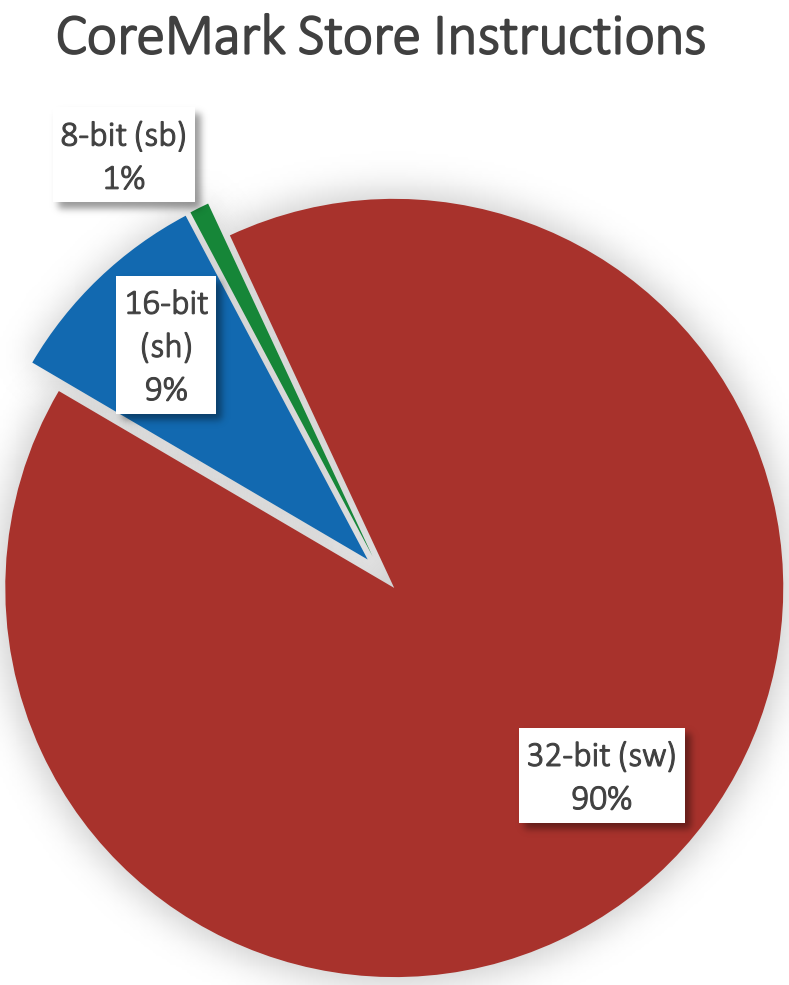


How do we tackle reliability of the System Memory?

Byte-addressable Memory – ECC Load and Store

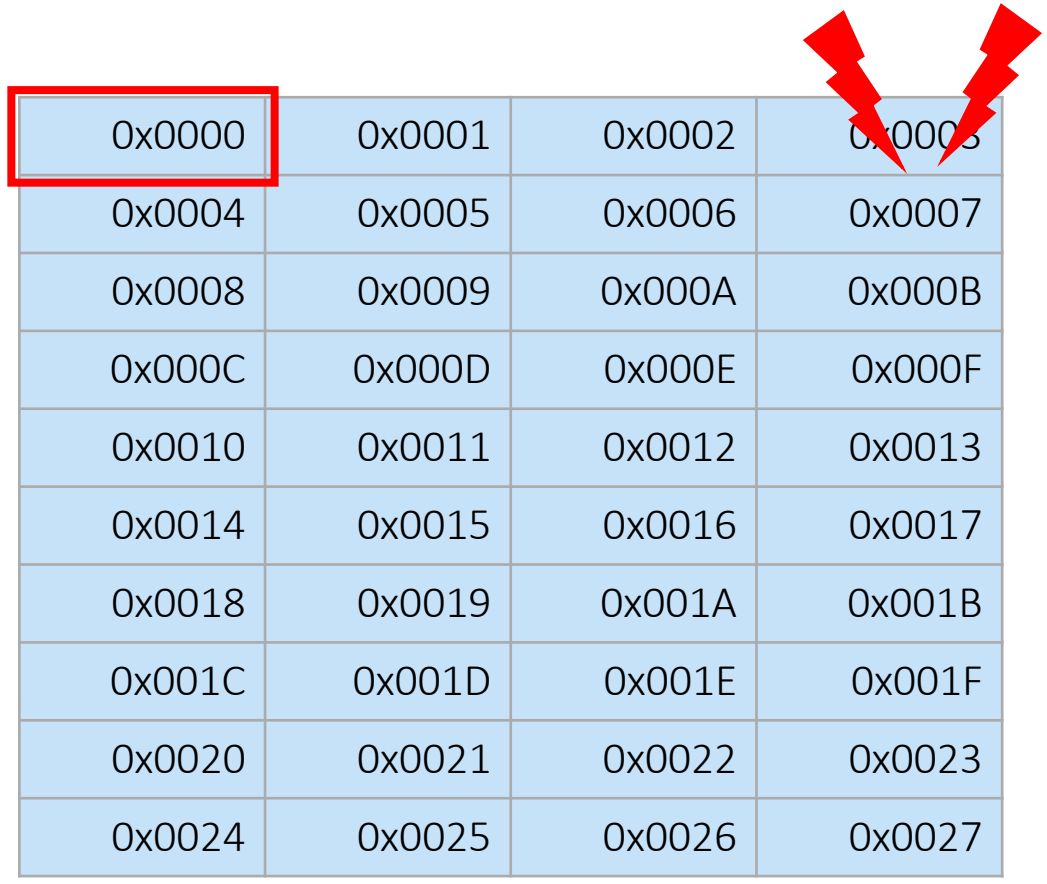


ECC Load-and-Store – Performance Impact



- 32-bit Hsiao ECC word protection
- Directly grant store
  - Delay following transaction, not current transaction, to shift & reduce impact
- Results: **<1% cycle increase**
  - Various tests, such as 8-bit Matrix-Matrix Multiplication

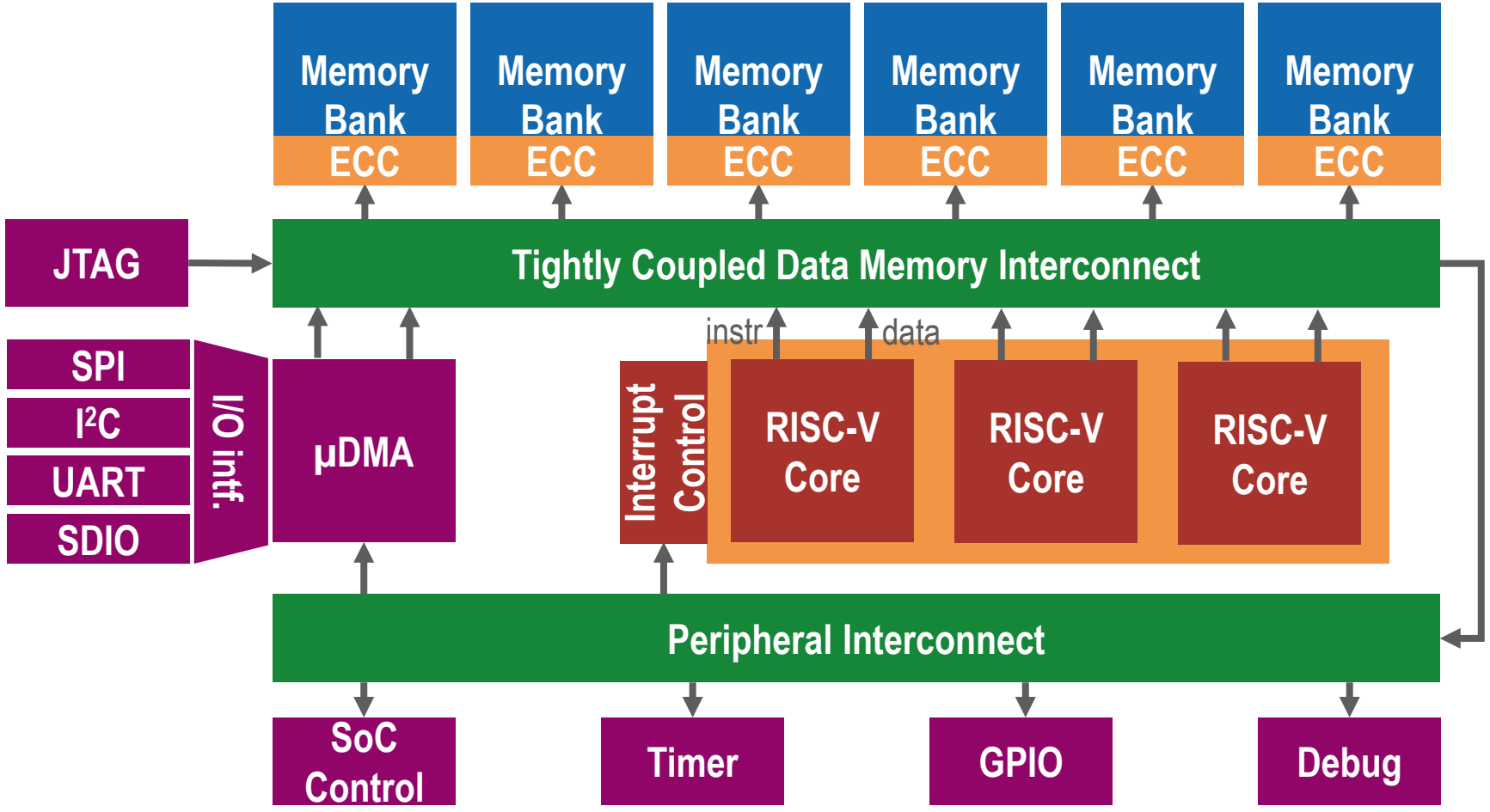
ECC Scrubber



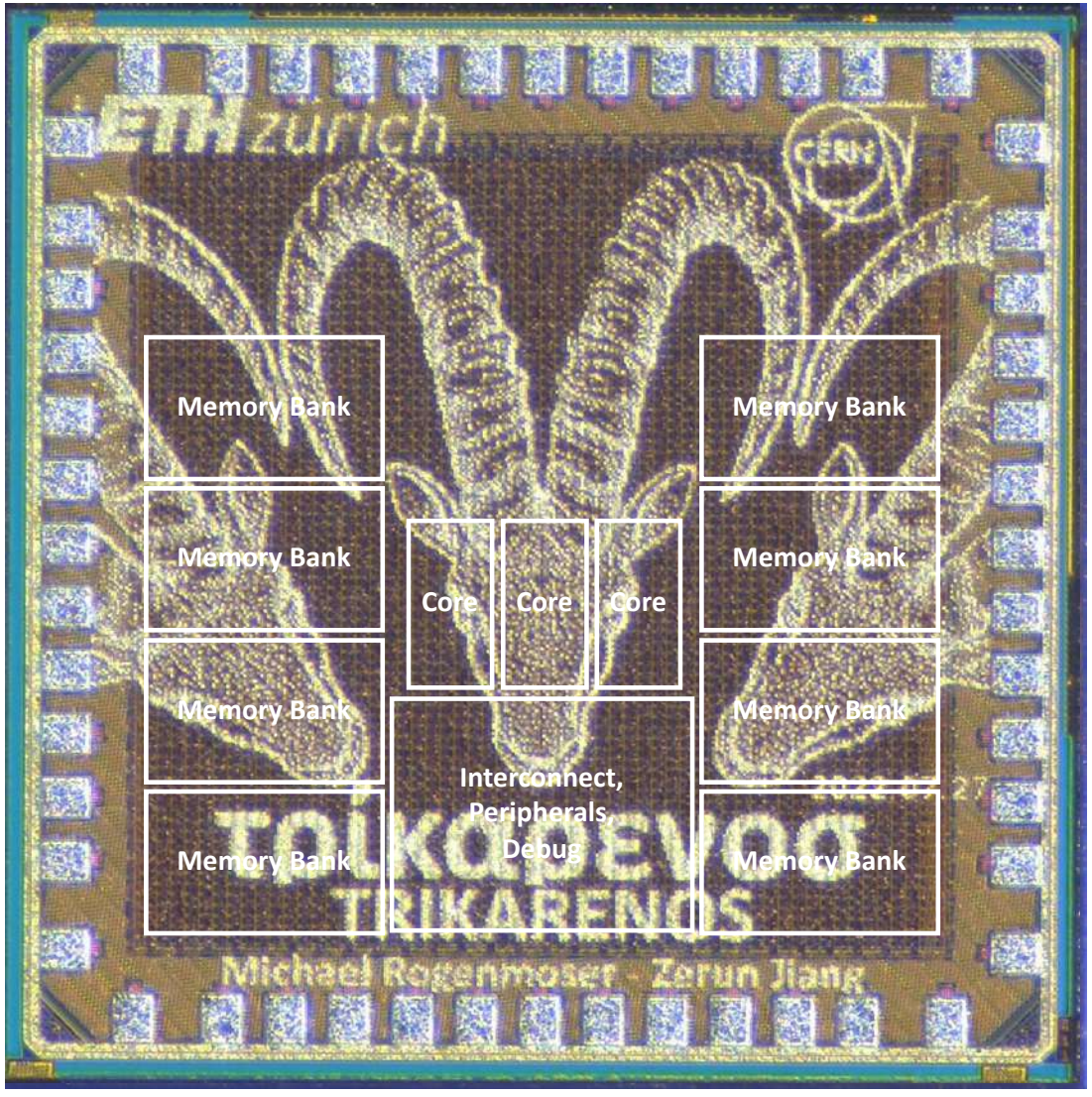
0x0000	0x0001	0x0002	0x0003
0x0004	0x0005	0x0006	0x0007
0x0008	0x0009	0x000A	0x000B
0x000C	0x000D	0x000E	0x000F
0x0010	0x0011	0x0012	0x0013
0x0014	0x0015	0x0016	0x0017
0x0018	0x0019	0x001A	0x001B
0x001C	0x001D	0x001E	0x001F
0x0020	0x0021	0x0022	0x0023
0x0024	0x0025	0x0026	0x0027

- Multiple errors in a single word lead to unrecoverable errors
- Scan Memory Bank
- Re-write faulty word if error is detected
- Defer permission to external accesses
- Log all corrections (and uncorrectable words)

Trikarenos – PULPissimo with Reliability



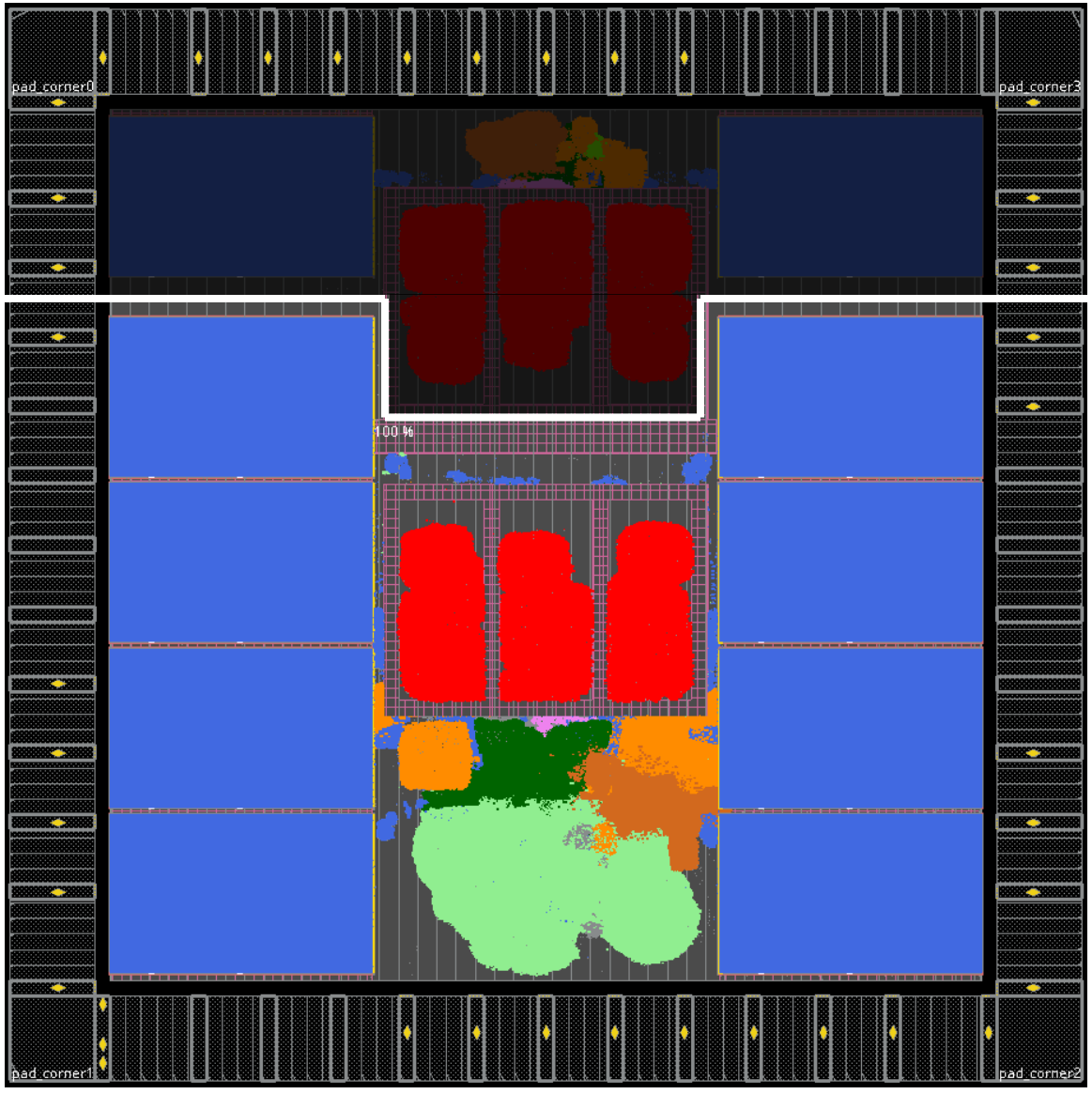
Trikarenos – ASIC implementation



- TSMC 28HPC+
  - Shown to have high TID tolerance
- 2 mm^2 @250 MHz
- 3 separate Ibex cores
- 256 KiB Memory in 8 word-interleaved banks

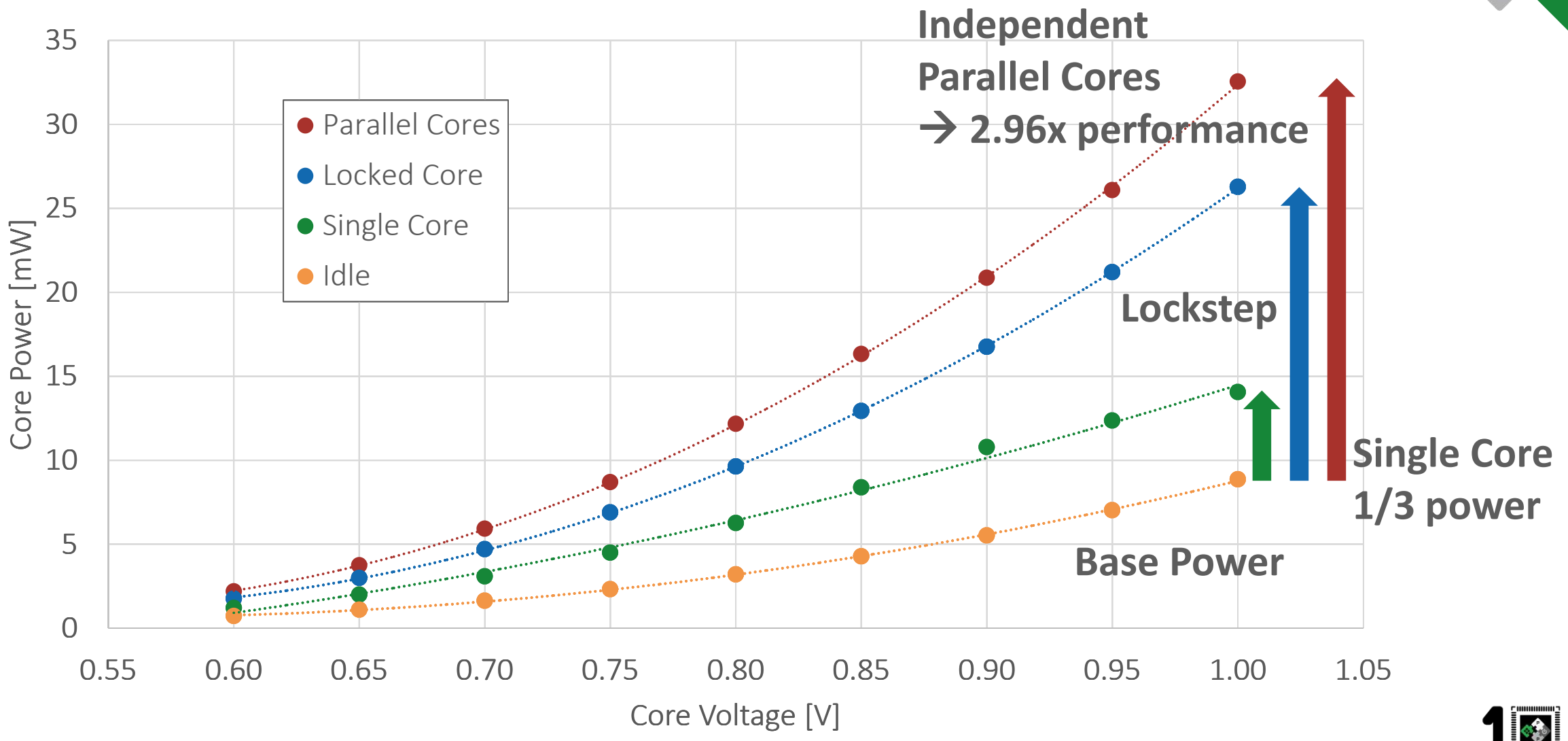
Internal structure

- Spatially separated cores with a keepout zone

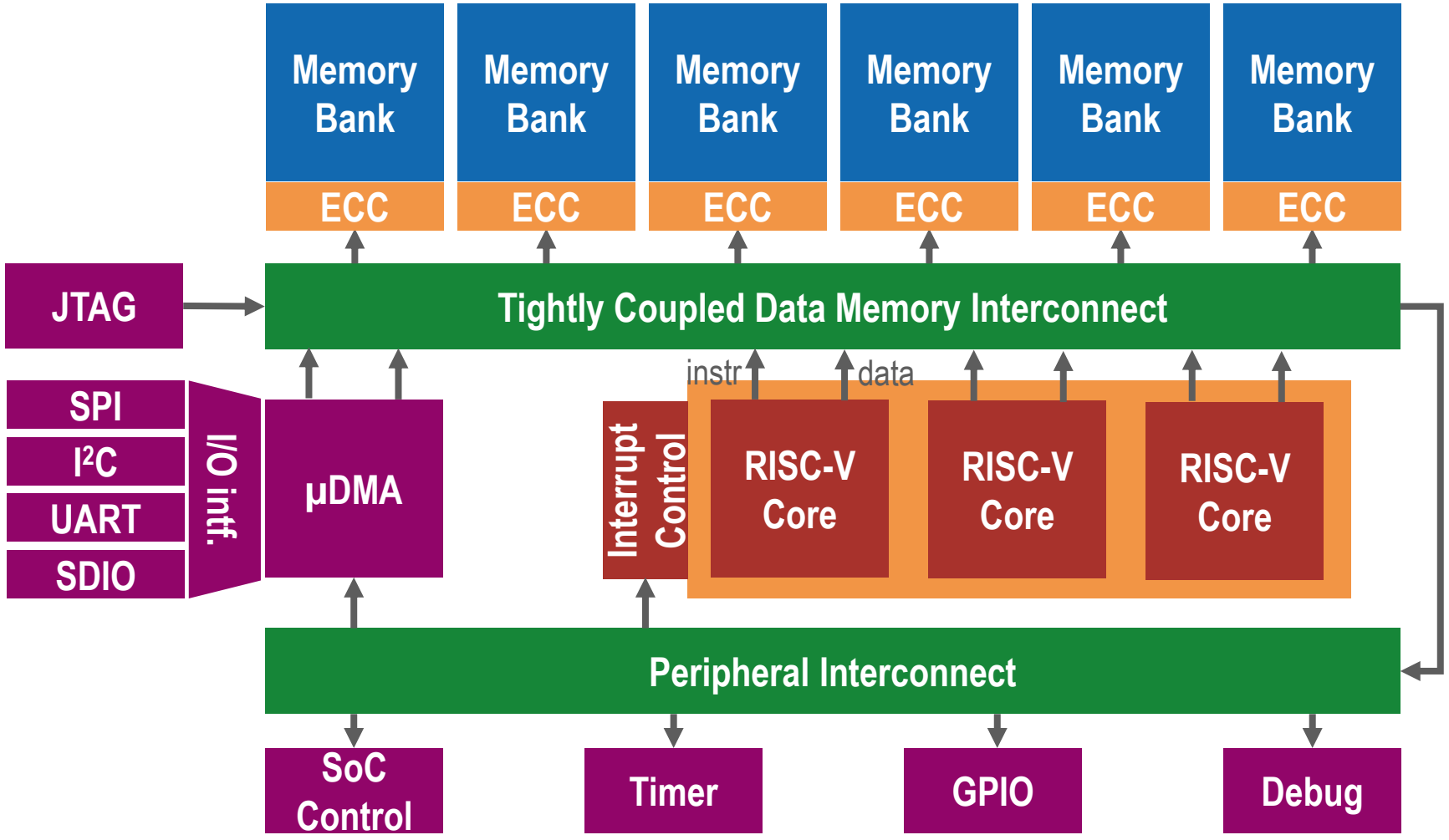


- Legend:
  - Cores
  - HMR Unit
  - Memory (w/ ECC en-/decode)
  - Interconnect
  - Debugger
  - Logging & control registers, ROM, …
  - I/O

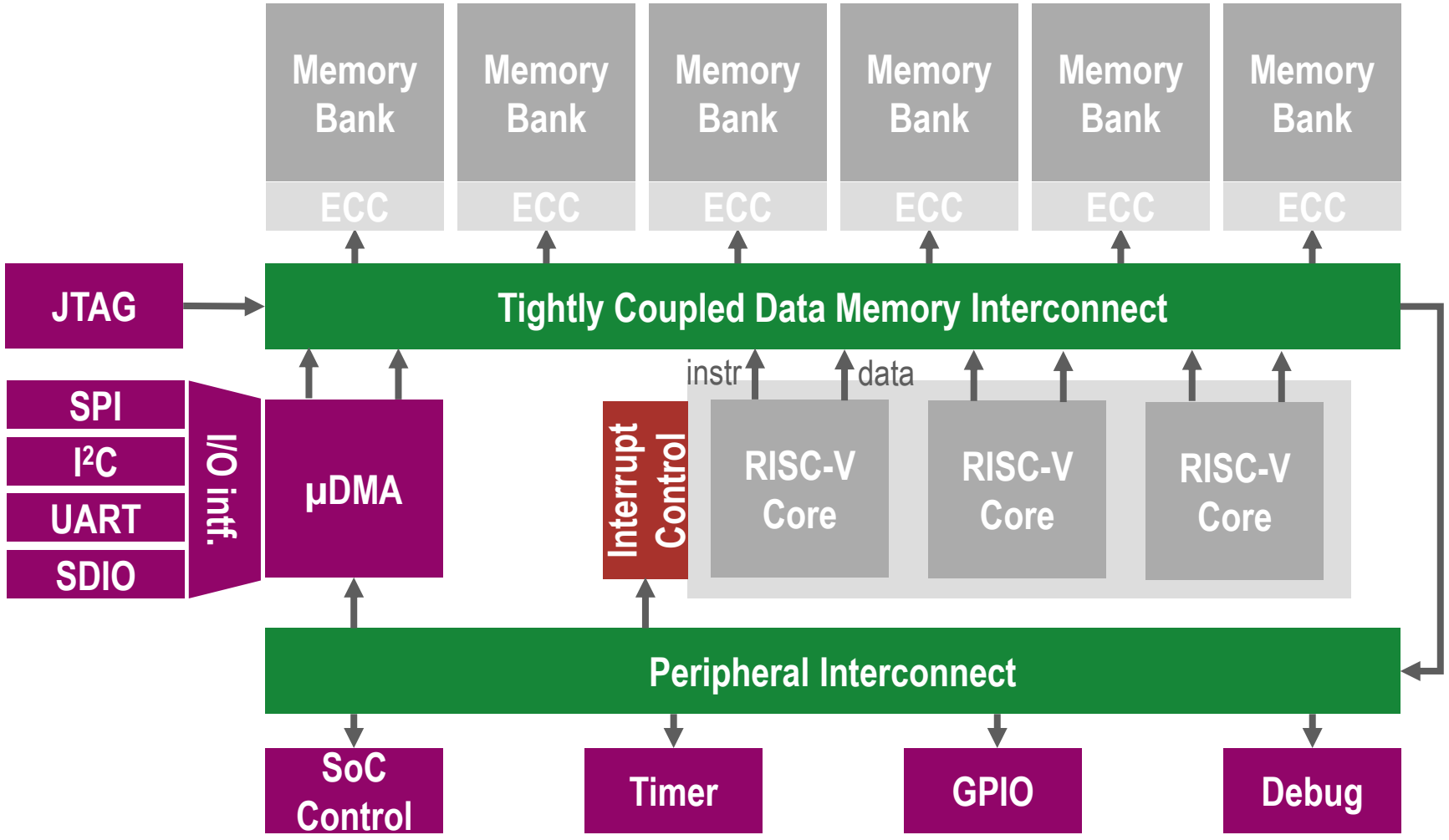
Power Consumption at max Frequency



Open points – PULP is a playground



Open points – PULP is a playground

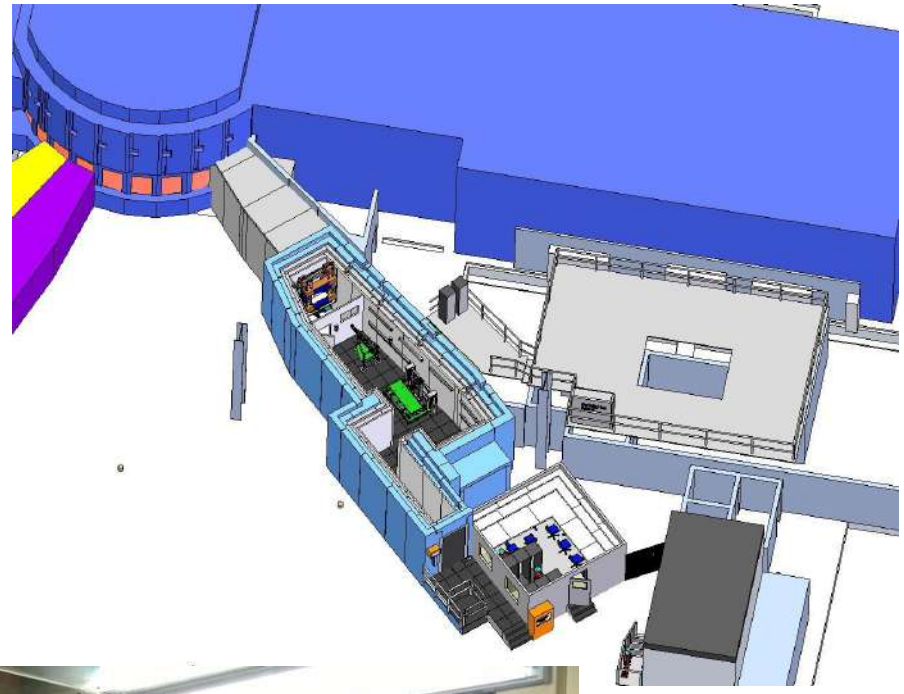


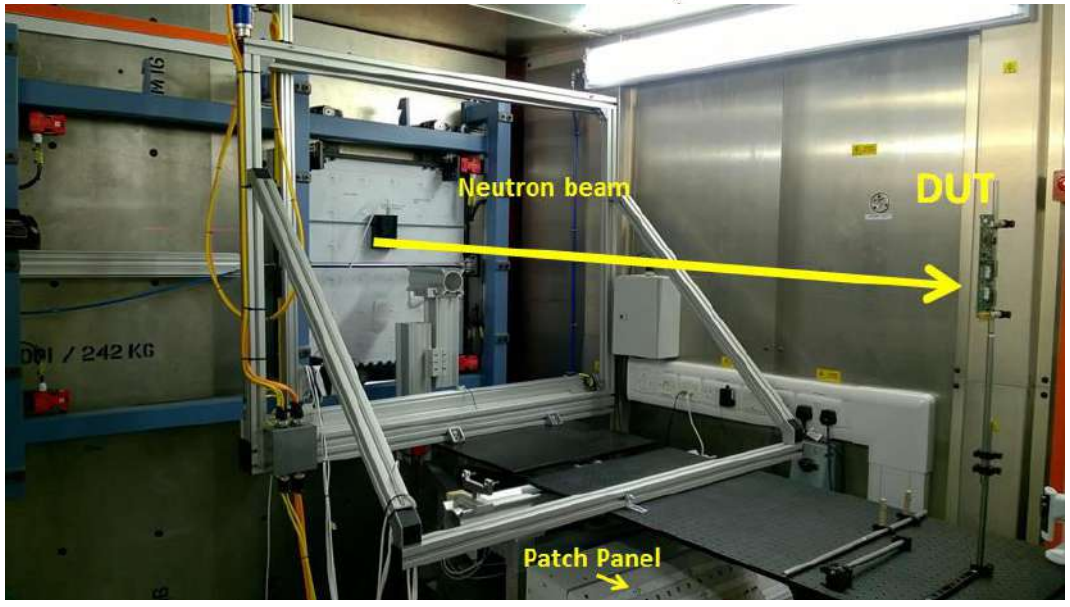
On-chip protection is not complete…

- Protect on-chip communication
  - TCDM Interconnect
  - AXI Interconnect
  - Network-on-Chip
- Protect System Peripherals
  - On-chip peripherals & control registers
  - I/O peripherals
- Scale-up to larger components & systems
  - Application-scale host core
  - Clusters & multi-cluster systems
  - Accelerators

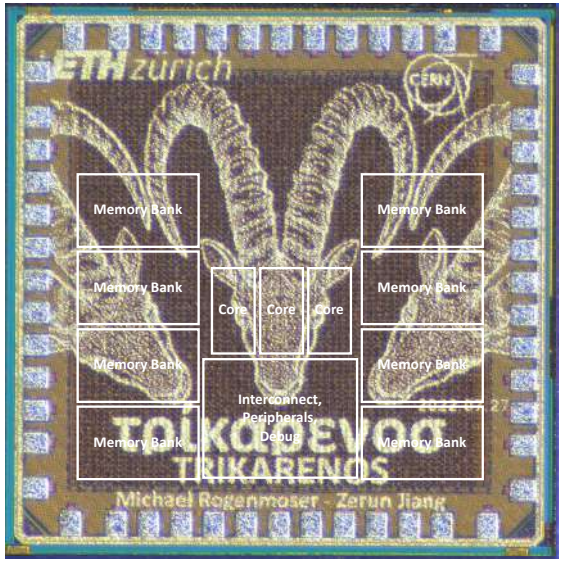
How do we test reliability?

- Simulation-based fault injection
  - SET & SEU
  - Very slow
  - Initial setup working
- Scan-chain Fault Injection
  - SEU only
  - Work-in-Progress
- Radiation Beam
  - Representative fault environment
  - Initial plans
- Space?

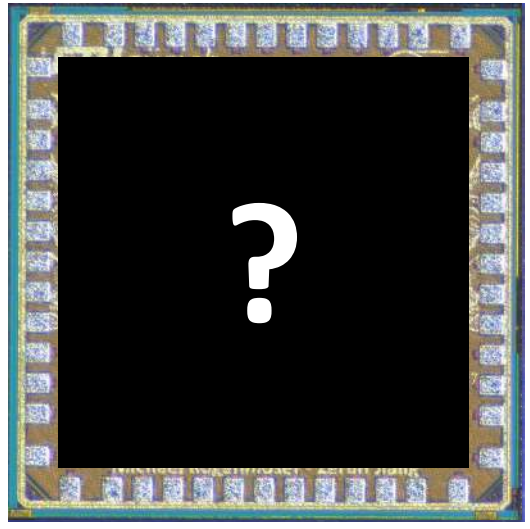




Are we going to space?

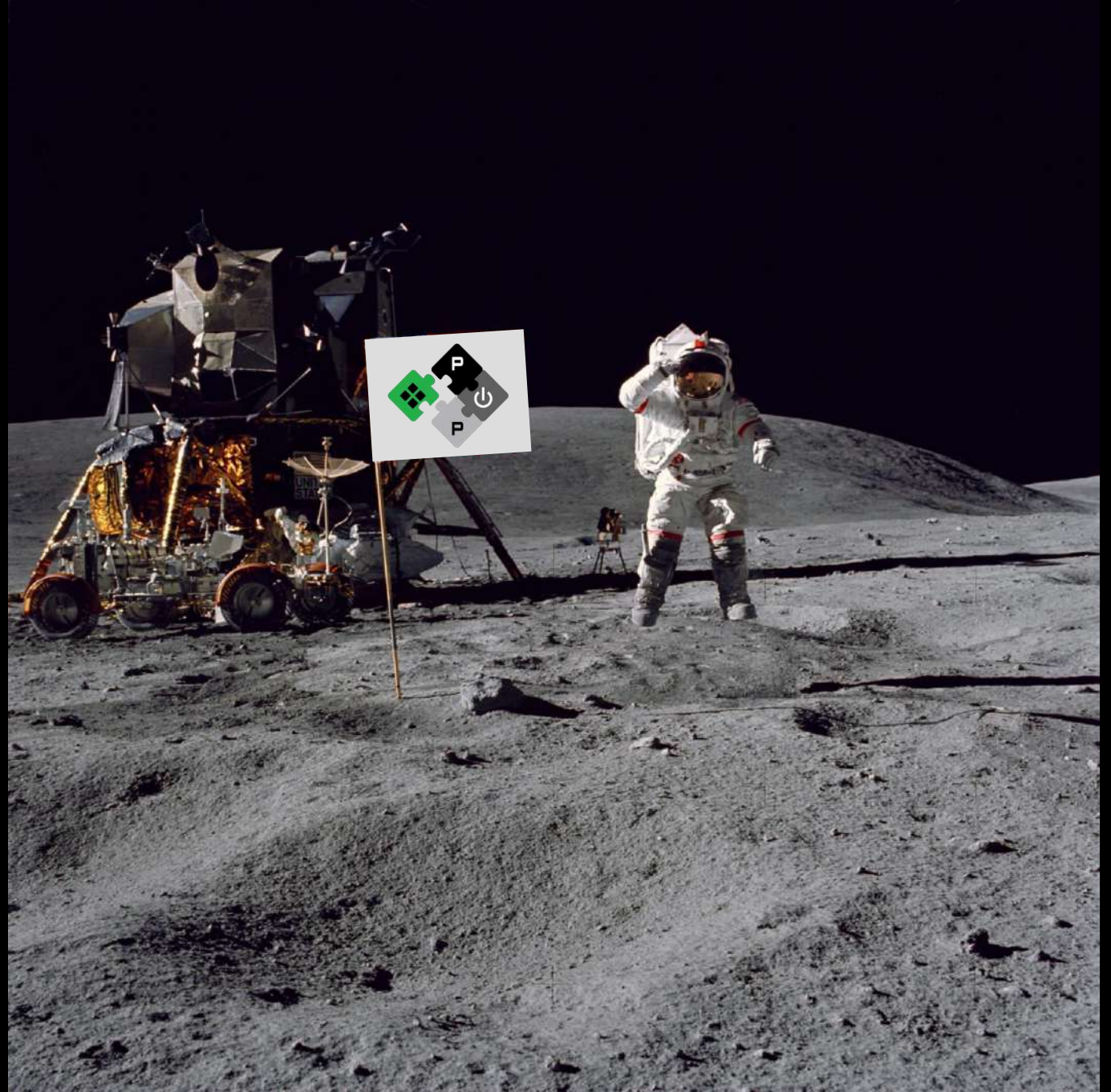


ThalesAlenia Space
a Thales / Leonardo company



aris
space to grow

ETH zürich | SPACE



PULP

Parallel Ultra Low Power

Luca Benini, Ahmad Mirsalari, Alessandro Capotondi, Alessandro Nadalini, Alessandro Ottaviano, Alessio Burrello, Alfio Di Mauro, Andrea Borghesi, Andrea Cossettini, Angelo Garofalo, Arpan Prasad, Chi Zhang, Corrado Bonfanti, Cristian Cioflan, Cyril Koenig, Daniele Palossi, Davide Rossi, Fabio Montagna, Florian Glaser, Francesco Conti, Georg Rutishauser, Germain Haugou, Gianna Paulin, Giuseppe Tagliavini, Hanna Müller, Jannis Schoenleber, Lorenzo Lamberti, Luca Bertaccini, Luca Colagrande, Luca Valente, Maicol Caini, Manuel Eggimann, Manuele Rusci, Marco Bertuletti, Marco Guermandi, Matheus Cavalcante, Matteo Perotti, Mattia Sinigaglia, Michael Rogenmoser, Moritz Scherer, Moritz Schneider, Nazareno Bruschi, Nils Wistoff,, Paul Scheffler, Philipp Mayer, Robert Balas, Samuel Riedel, Segio Mazzola, Sergei Vostrikov, Simone Benatti, Thomas Benz, Thorir Ingolfsson, Tim Fischer, Victor Javier Kartsch Morinigo, Victor Jung, Viviane Potocnik, Vlad Niculescu, Xiaying Wang, Yichao Zhang, Yvan Tortorella, Frank K. Gürkaynak, all our past collaborators

and many more that we forgot to mention

http://pulp-platform.org

@pulp_platform