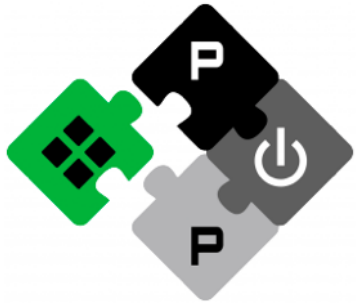


PULP PLATFORM

Open Source Hardware, the way it should be!

Leveraging the PULP Platform to Build Reliable Systems

Luca Bertaccini (ETH Zurich) <lbertaccini@iis.ee.ethz.ch>

Michael Rogenmoser (ETH Zurich) <michaero@iis.ee.ethz.ch>

ETH zürich

 http://pulp-platform.org  @pulp_platform  https://www.youtube.com/pulp_platform

Agenda

- *Luca Bertaccini (ETHZ): "PULP: An Energy-Efficient Open-Source RISC-V Based IoT End Node"*
- *Michael Rogenmoser (ETHZ): "Adding Reliability Features to PULP Systems"*

Agenda

- *Luca Bertaccini (ETHZ): "PULP: An Energy-Efficient Open-Source RISC-V Based IoT End Node"*
- *Michael Rogenmoser (ETHZ): "Adding Reliability Features to PULP Systems"*

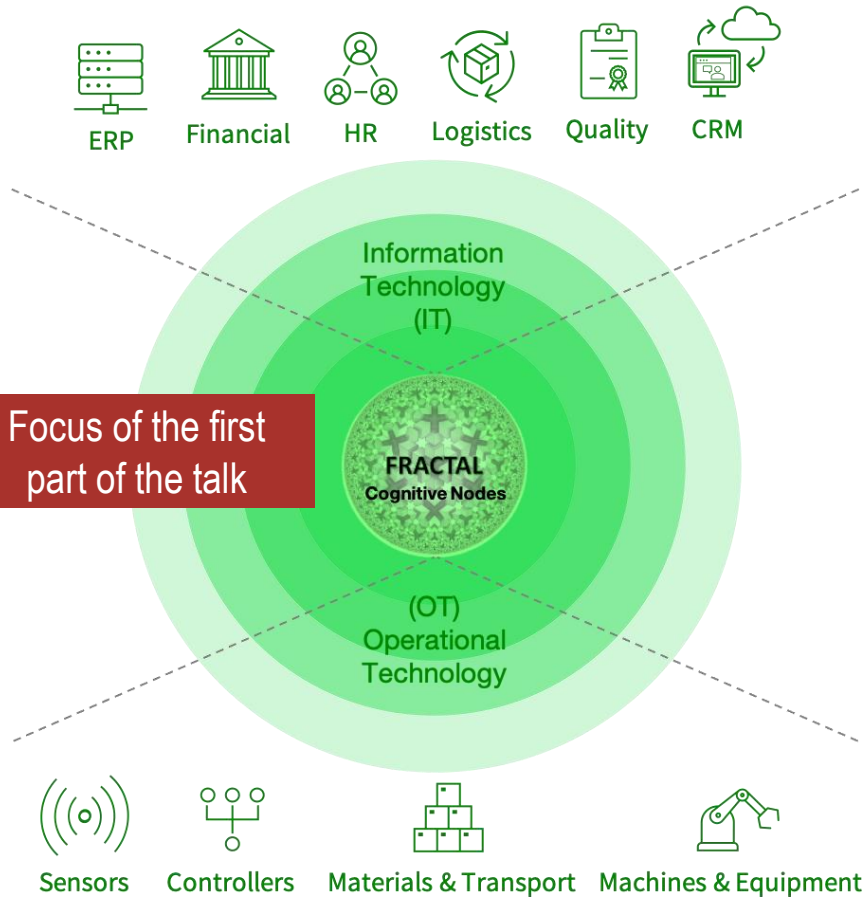
A Low-Power Fractal End Node

A reliable cognitive computing node:

- AI capabilities in the power envelope of an MCU

- Reliability features on the edge device

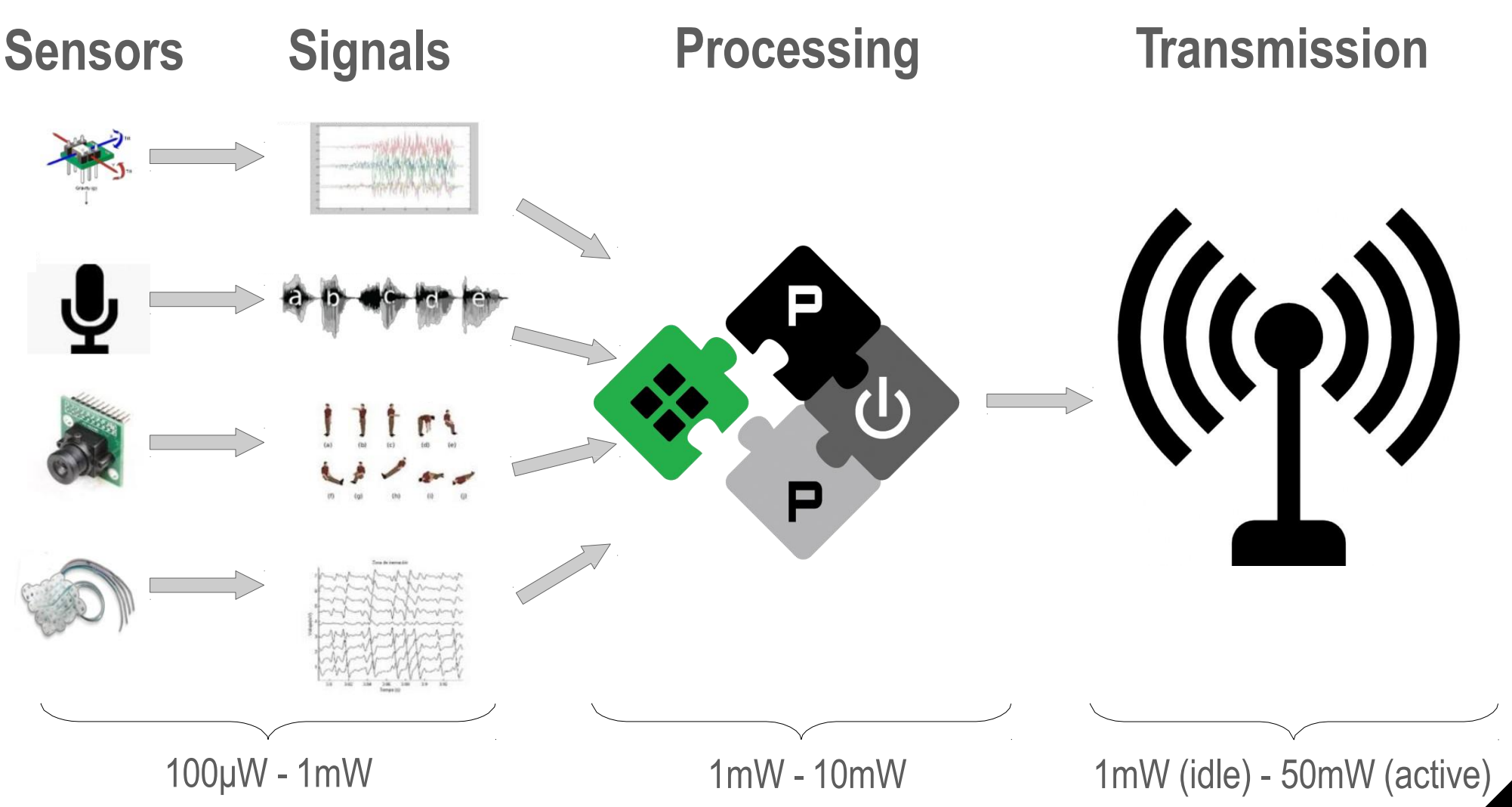
Focus of the first part of the talk



https://fractal-project.eu

Supported in part by the European Union's H2020 Fractal #877056

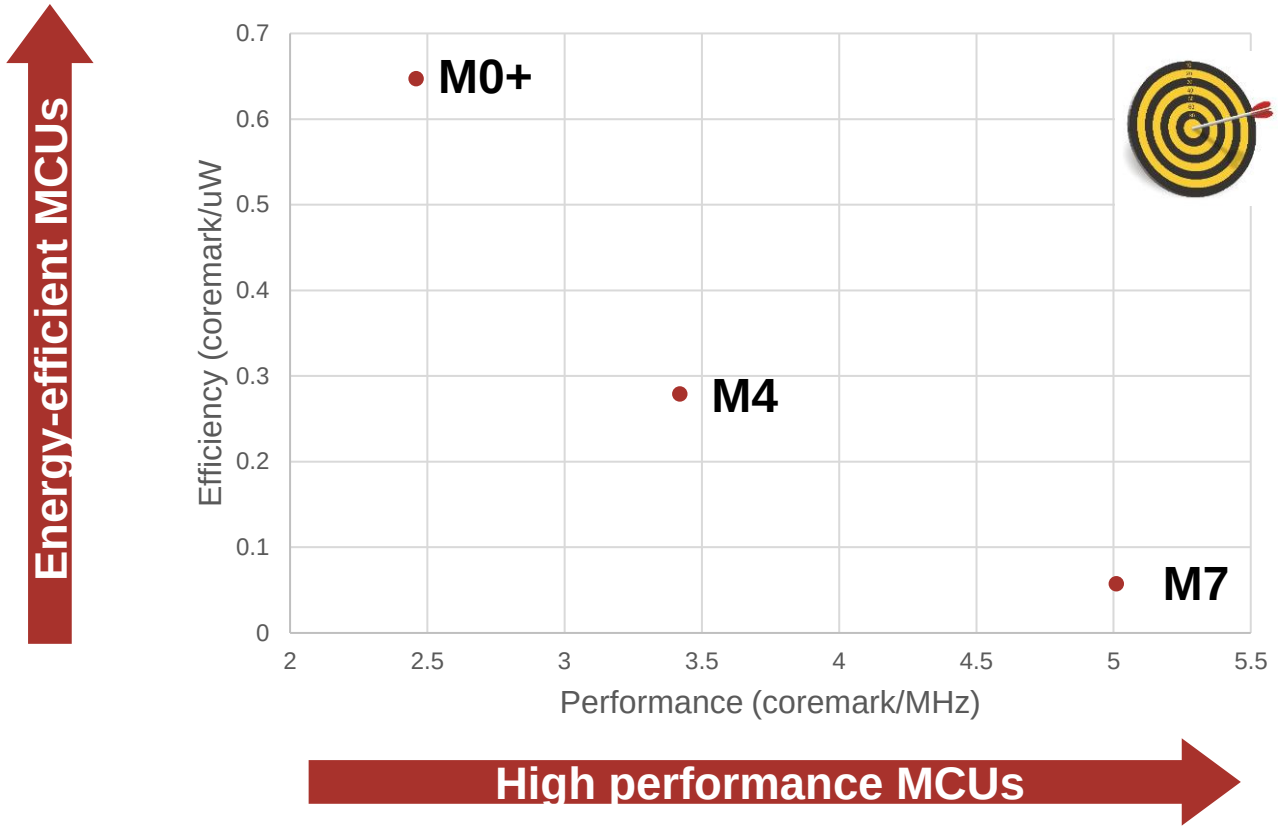
Edge Processing



How can we build an energy-efficient IoT end node with AI capabilities?

Energy efficiency @ GOPS is the Challenge

ARM Cortex-M MCUs: M0+, M4, M7 (40LP, typ, 1.1V)*



How??

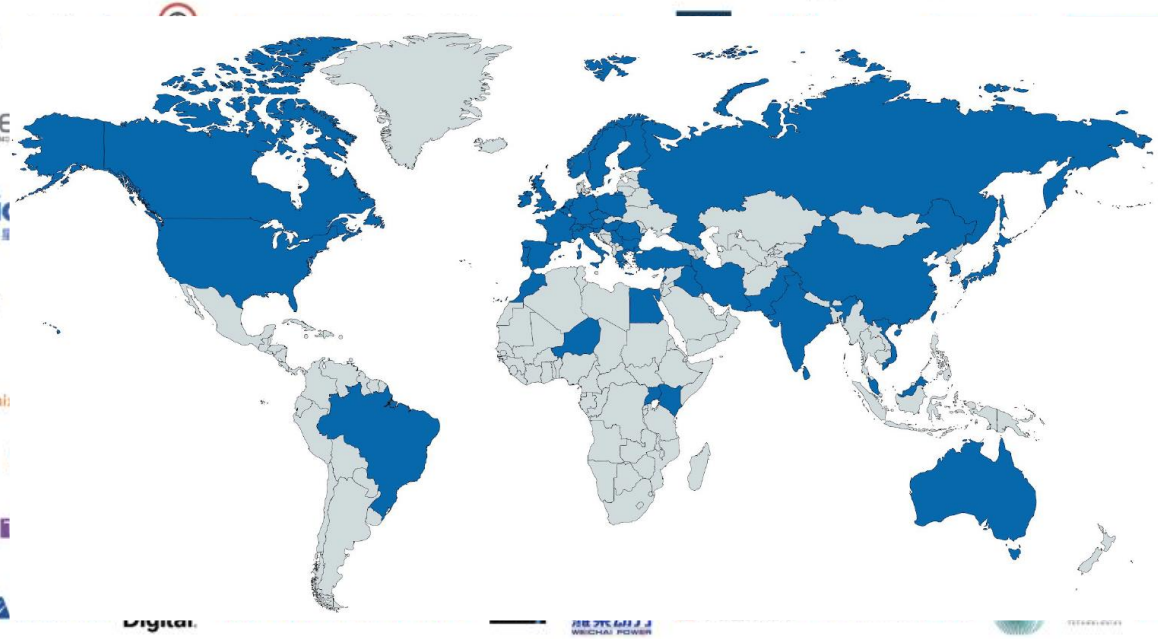
*data from ARMs web

RISC-V: an Open Extensible ISA

A modern, open, free ISA, **extensible *by construction***

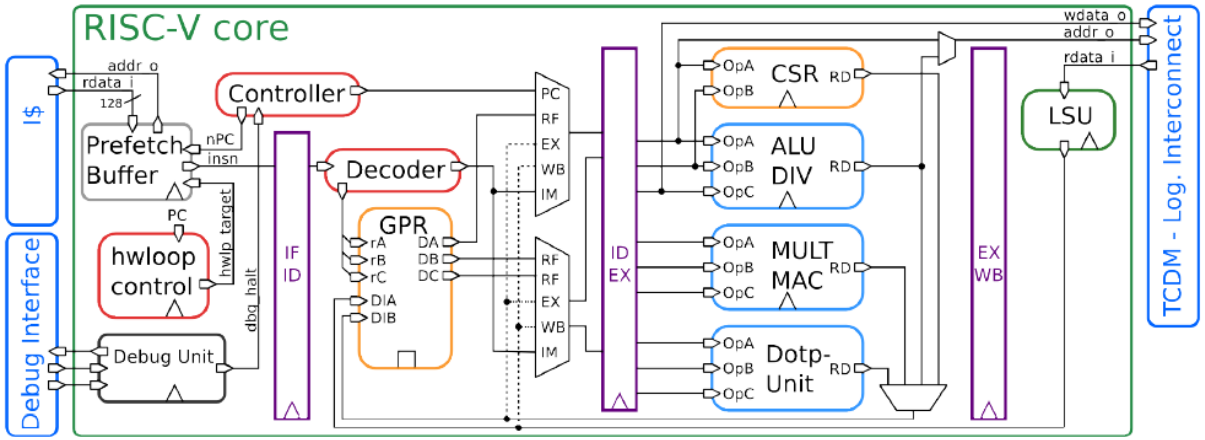
Endorsed and Supported by 600+ Members

Changed the picture on Computing Systems Research!



RI5CY Processor: in-order 4-stage Core

3-cycle ALU-OP, 4-cyle MEM-OP→IPC loss: LD-use, Branch



V1: Baseline RISC-V RV32IMC (not good for ML)

V2: HW loops, Post modified Load/Store, Mac

RISC-V → V1

V2

RI5CY ISA extensions improve ML performance

```
for (i = 0; i < 100; i++)
    d[i] = a[i] + b[i];
```

8-bit workload

Baseline

```
mv    x5, 0
mv    x4, 100
Lstart:
  lb    x2, 0(x10)
  lb    x3, 0(x11)
  addi  x10,x10, 1
  addi  x11,x11, 1
  add   x2, x3, x2
  sb    x2, 0(x12)
  addi  x4, x4, -1
  addi  x12,x12, 1
bne     x4, x5, Lstart
```

11 cycles/output

Auto-incr load/store

```
mv    x5, 0
mv    x4, 100
Lstart:
  lb    x2, 0(x10!)
  lb    x3, 0(x11!)
  addi  x4, x4, -1
  add   x2, x3, x2
  sb    x2, 0(x12!)
bne     x4, x5, Lstart
```

8 cycles/output

HW Loop

```
lp.setupi 100, Lend
  lb    x2, 0(x10!)
  lb    x3, 0(x11!)
  add   x2, x3, x2
Lend:  sb x2, 0(x12!)
```

5 cycles/output

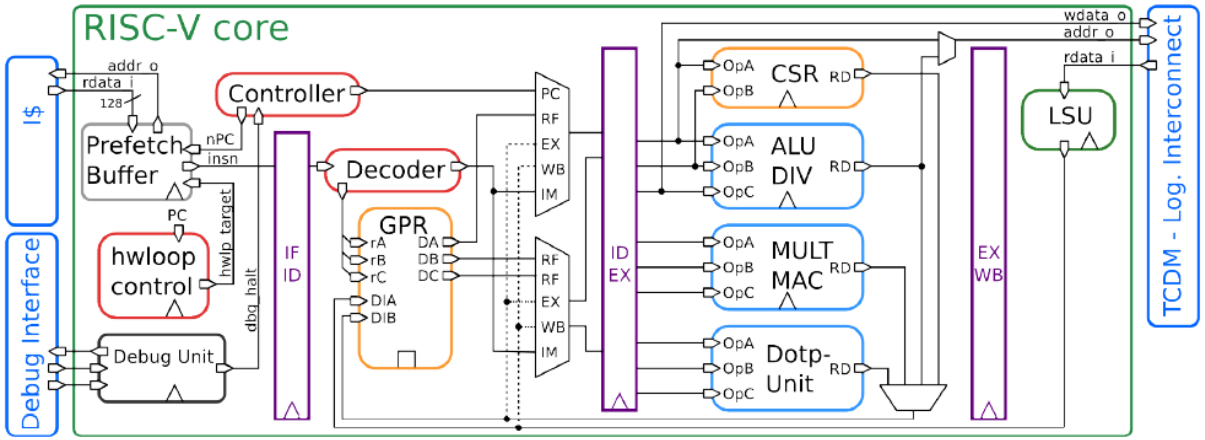
Packed-SIMD

```
lp.setupi 25, Lend
  lw    x2, 0(x10!)
  lw    x3, 0(x11!)
  pv.add.b x2, x3, x2
Lend: sw x2, 0(x12!)
```

1,25 cycles/output

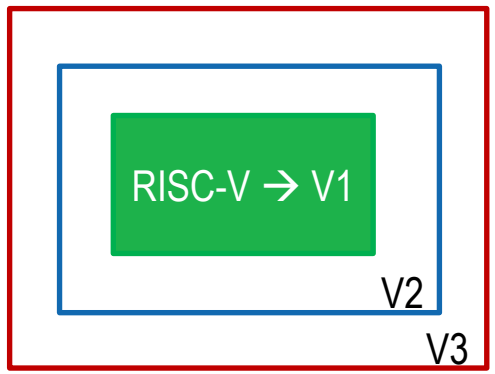
RI5CY Processor: in-order 4-stage Core

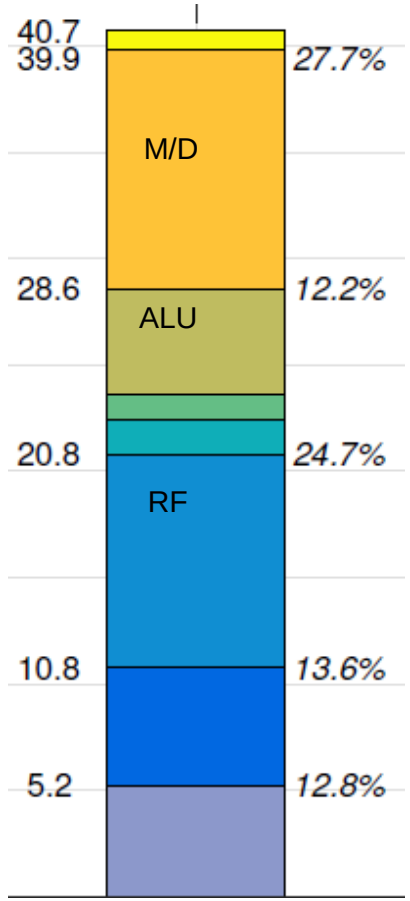
3-cycle ALU-OP, 4-cyle MEM-OP→IPC loss: LD-use, Branch



V1: Baseline RISC-V RV32IMC (not good for ML)
V2: HW loops, Post modified Load/Store, Mac
V3: SIMD 2/4 + DotProduct + Shuffling, Bit manipulation, Lightweight fixed point

XPULP 25 kGE → 40 kGE (1.6x) but 9+ times DSP!

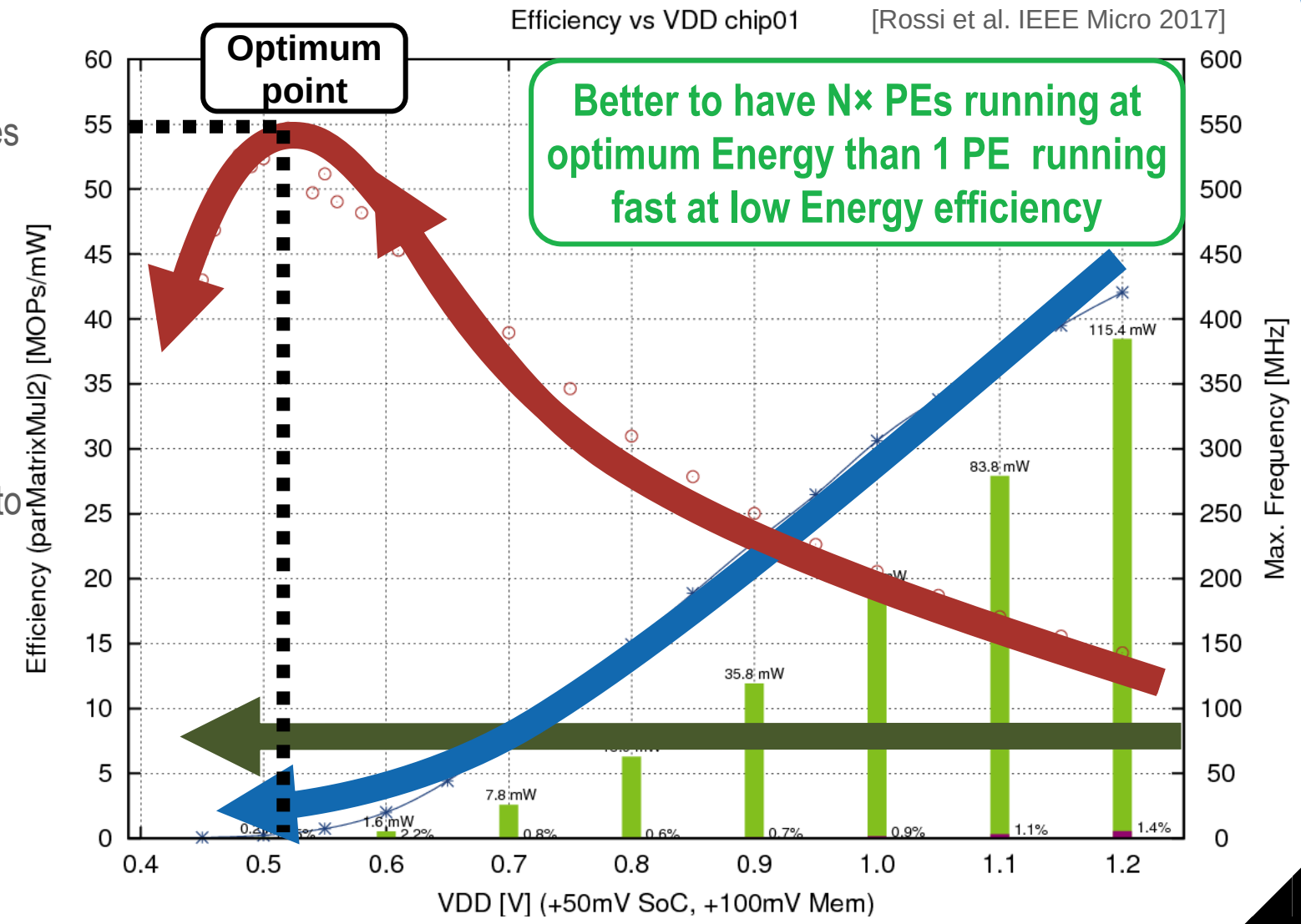




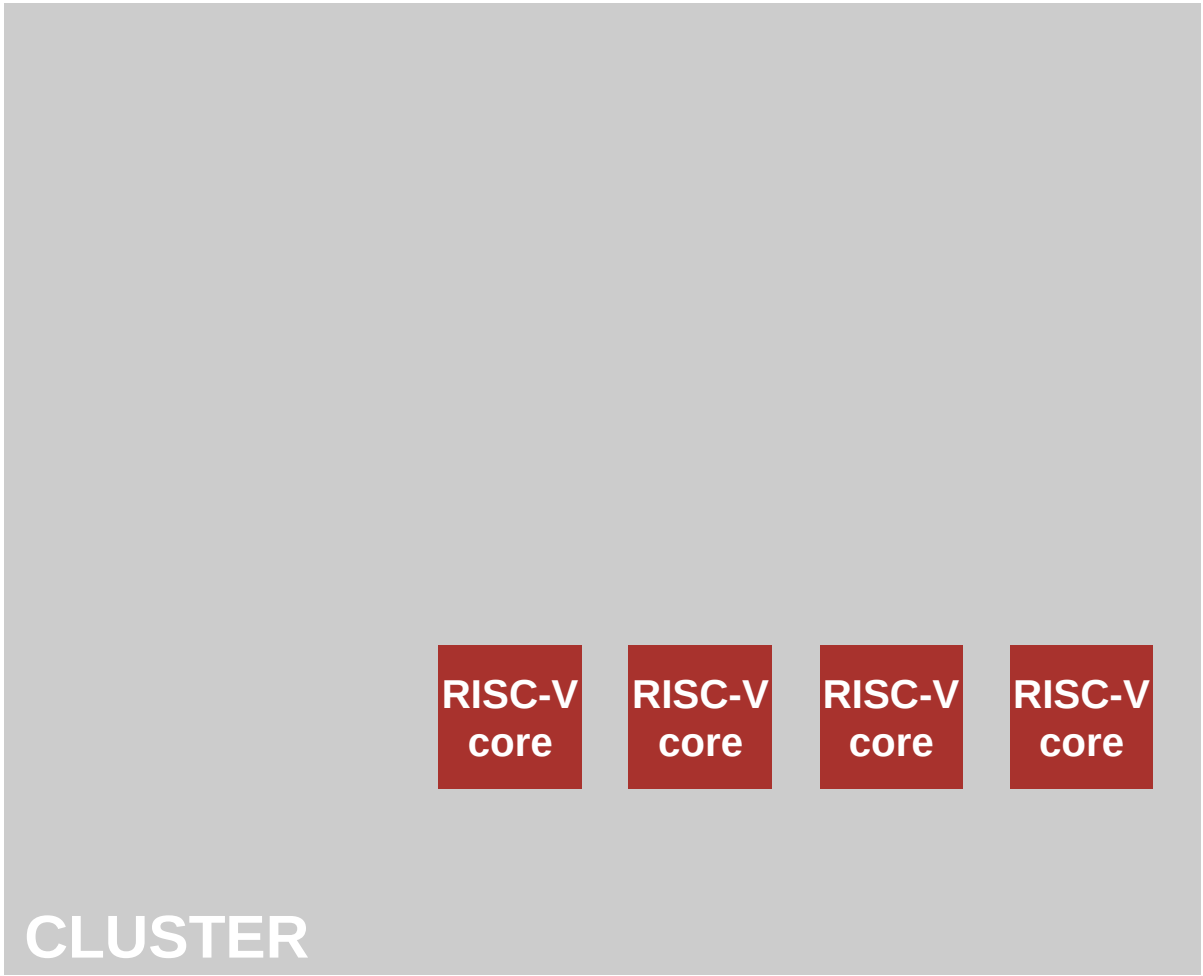
70% RF+DP

ML & Parallel + Near-threshold

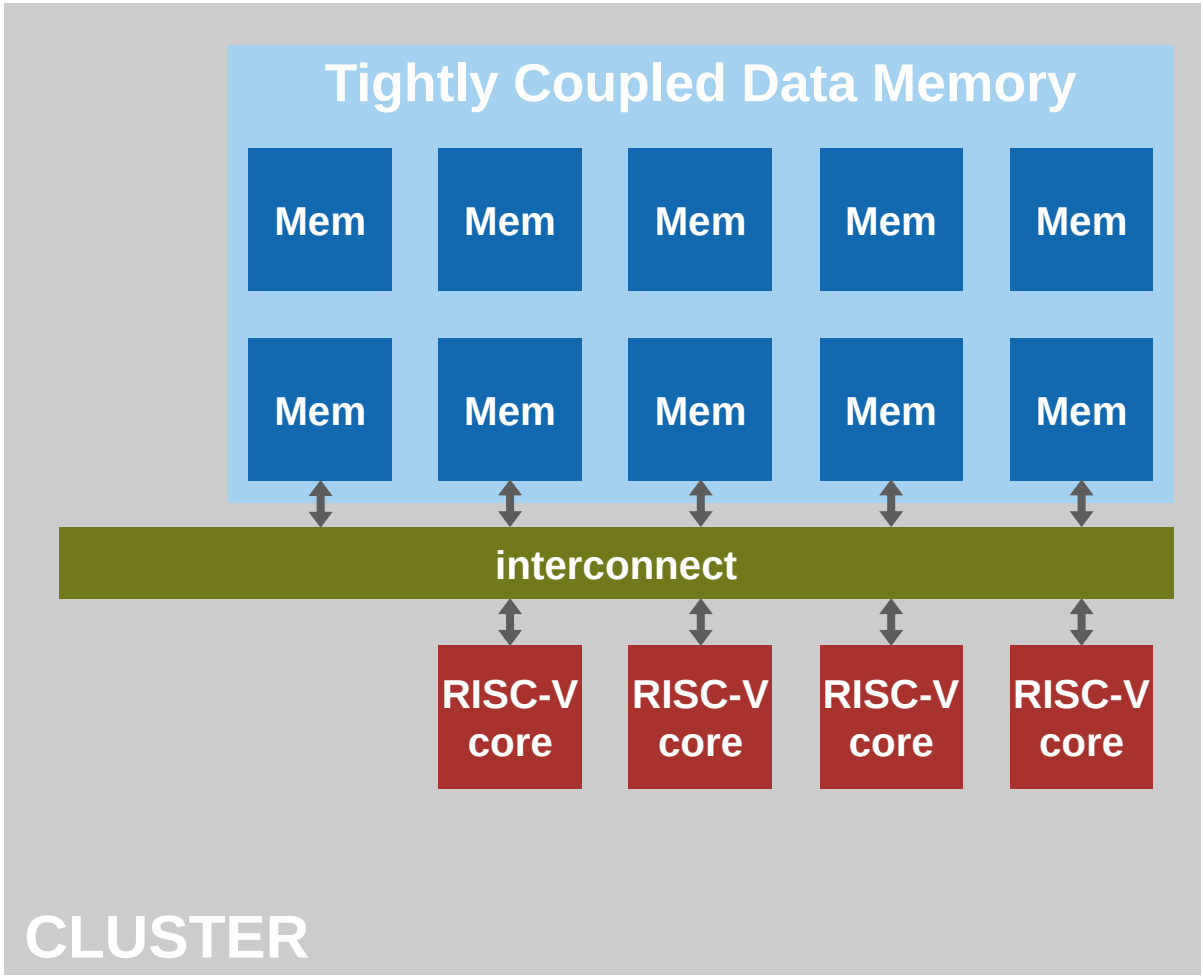
- As VDD decreases, operating speed decreases
- However efficiency increases→ more work done per Joule
- Until leakage effects start to dominate
- Put more units in parallel to get performance up and keep them busy with a parallel workload
- **ML is massively parallel and scales well (P/S ↑ with NN size)**



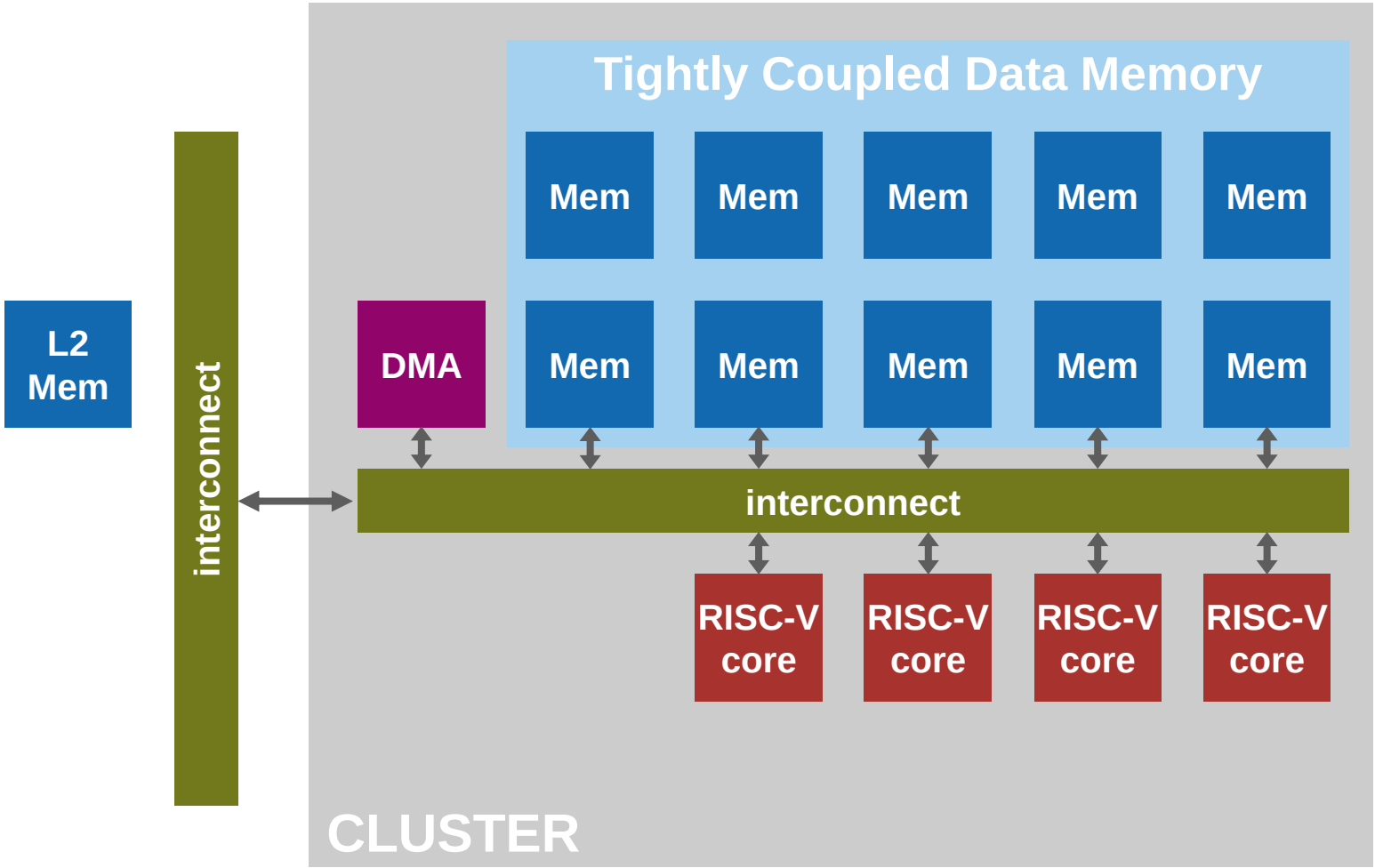
PULP cluster contains multiple RISC-V cores (1-16)



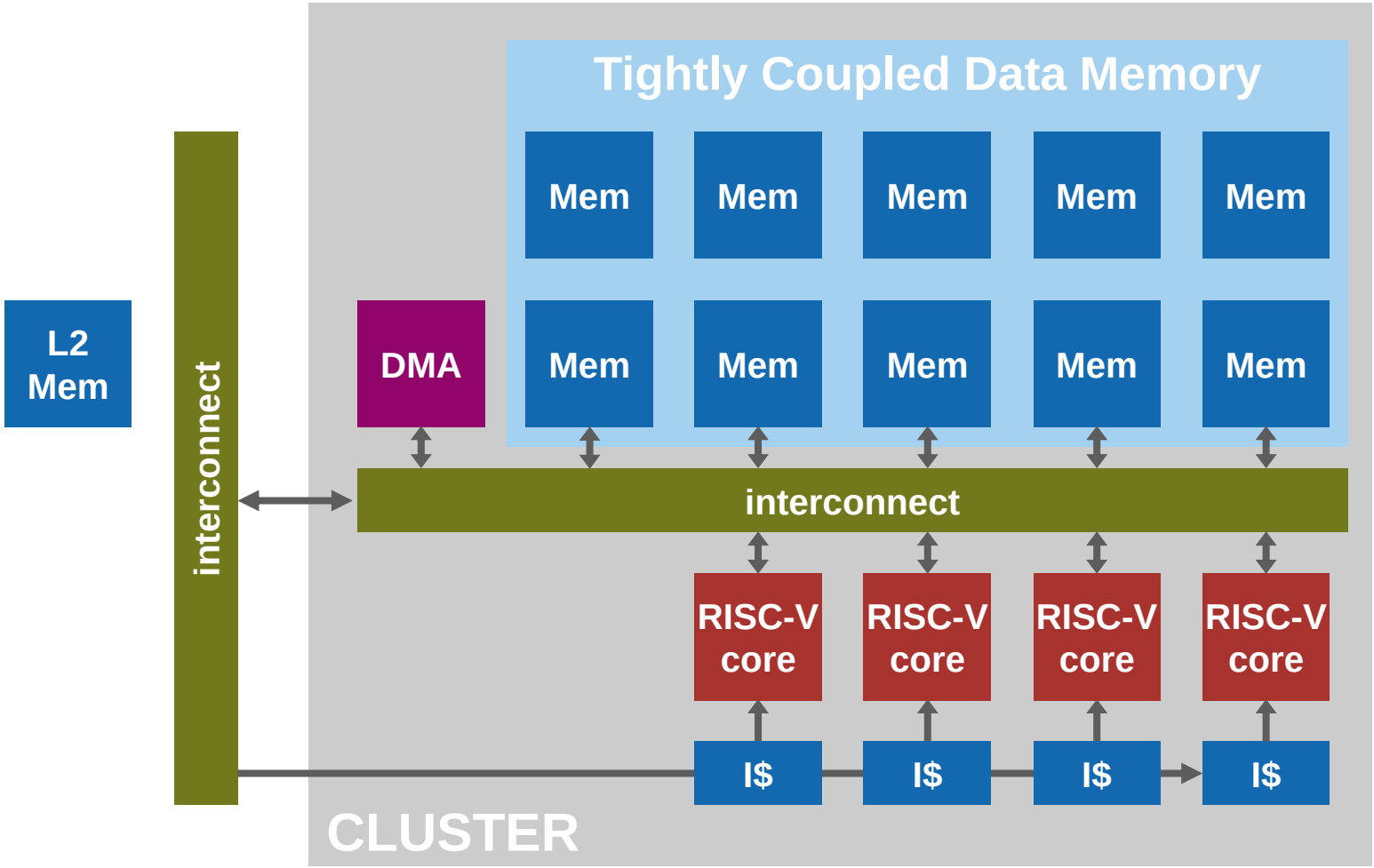
All cores can access all memory banks in the cluster



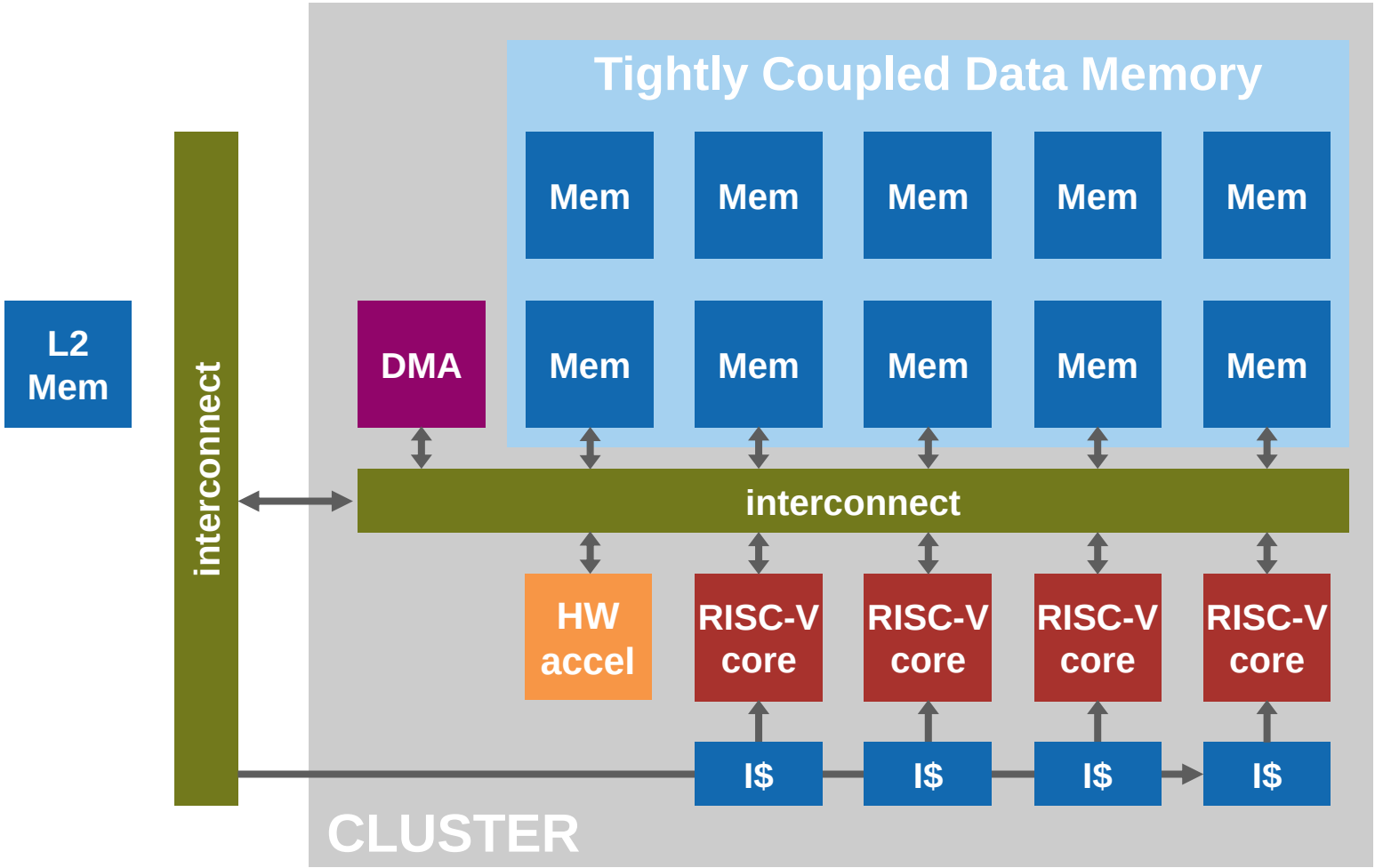
Data is copied from a higher level through DMA



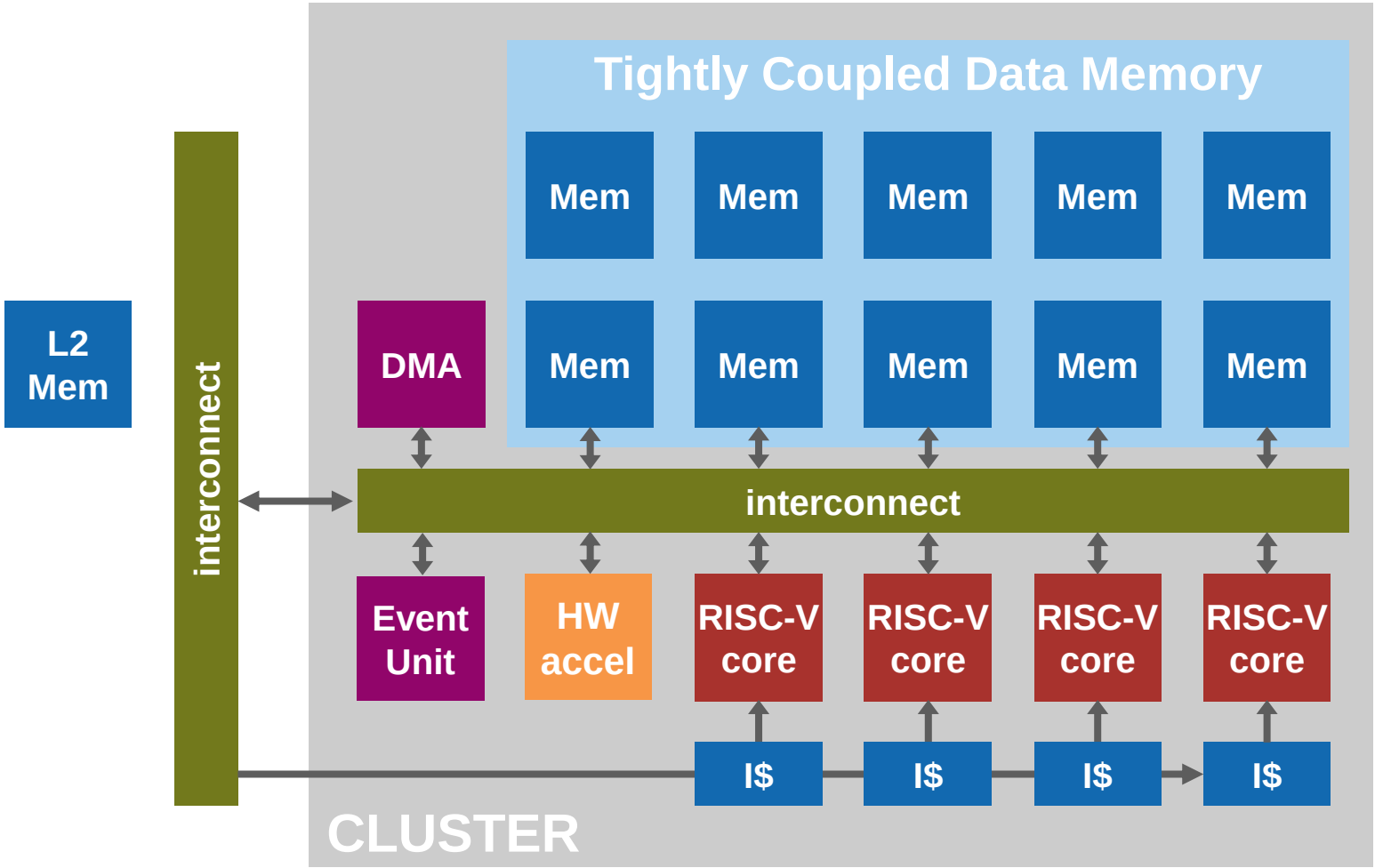
A (shared) instruction cache fetches from L2



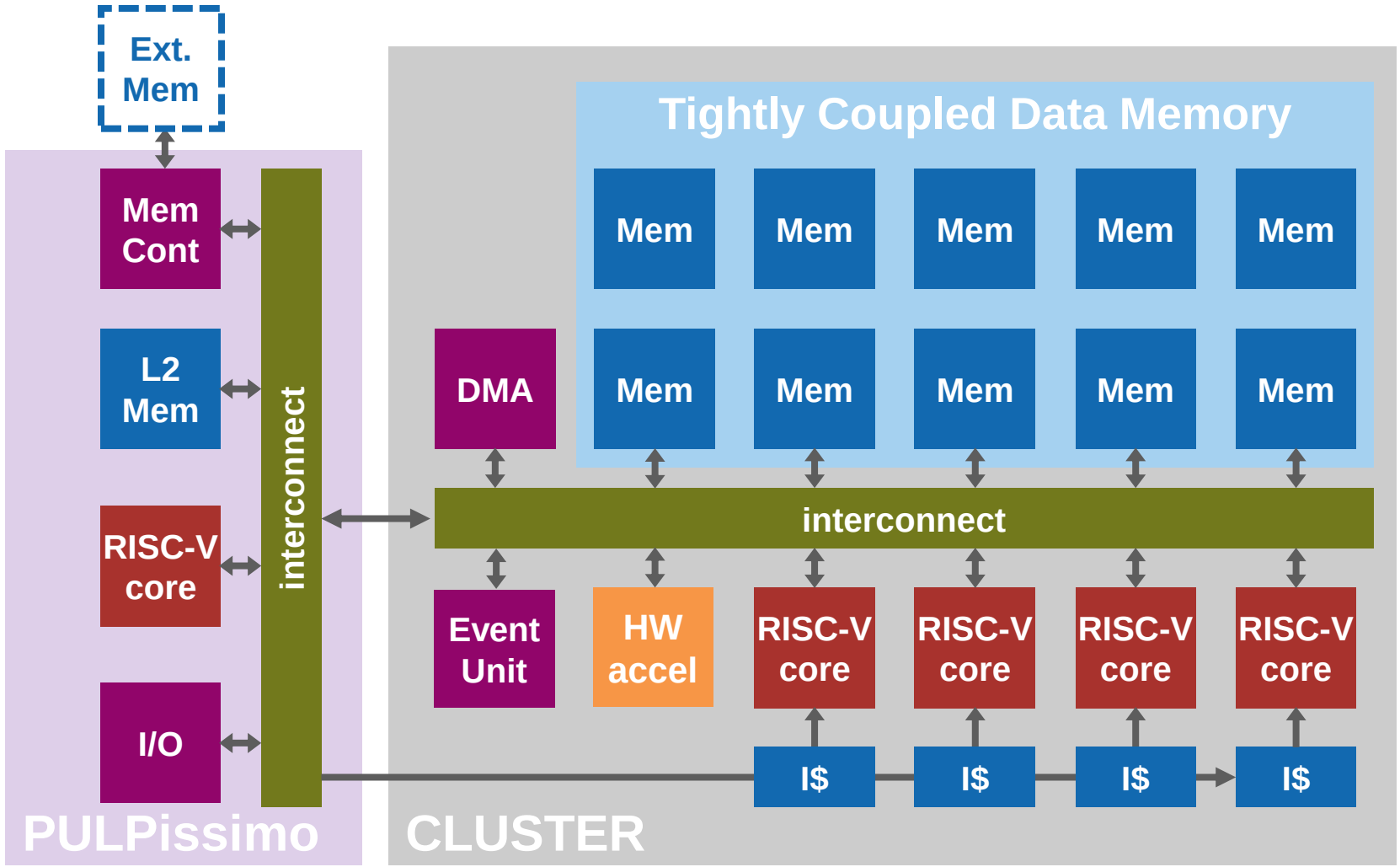
Hardware Accelerators can be added to the cluster



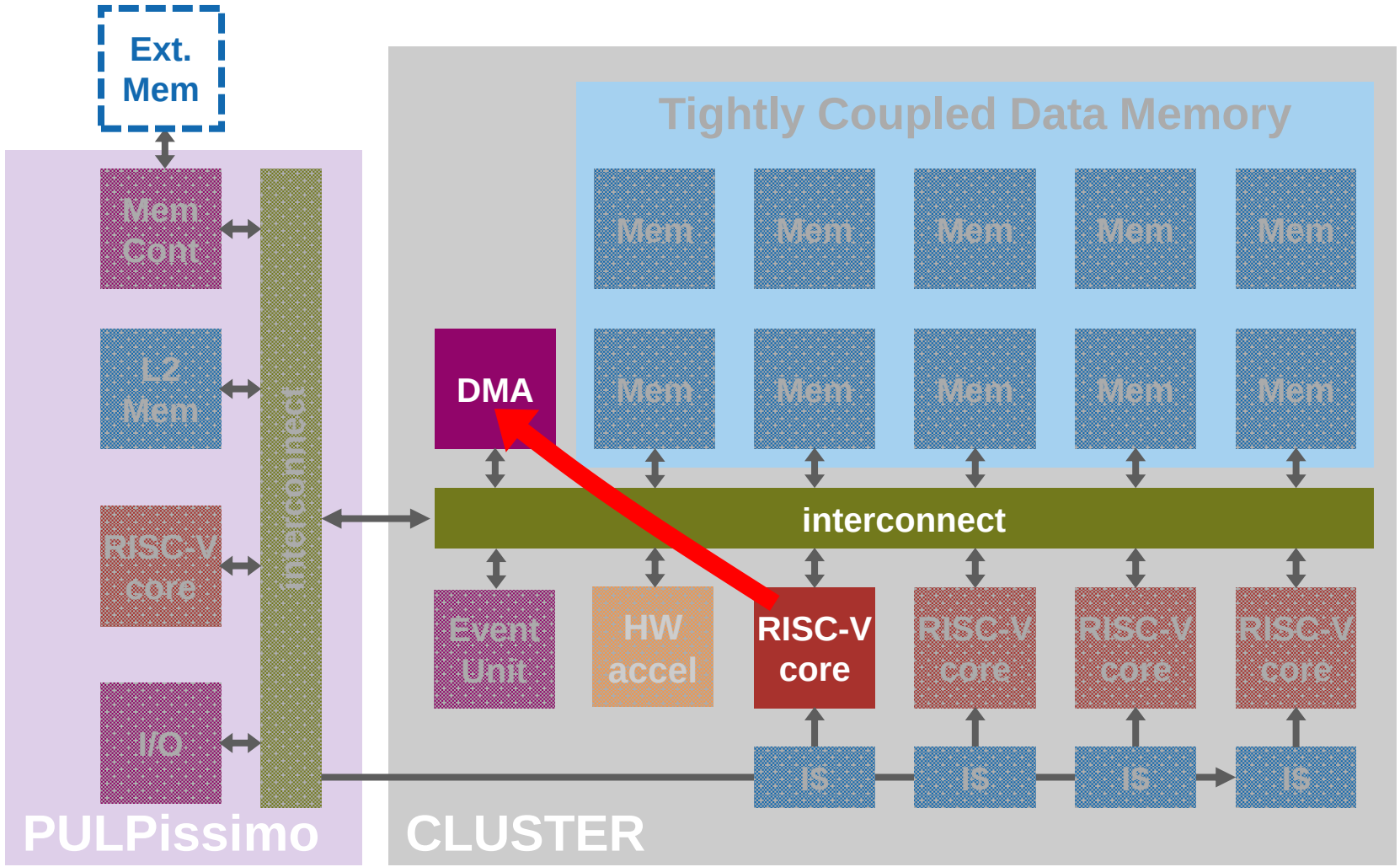
Event unit to manage resources (fast sleep/wakeup)



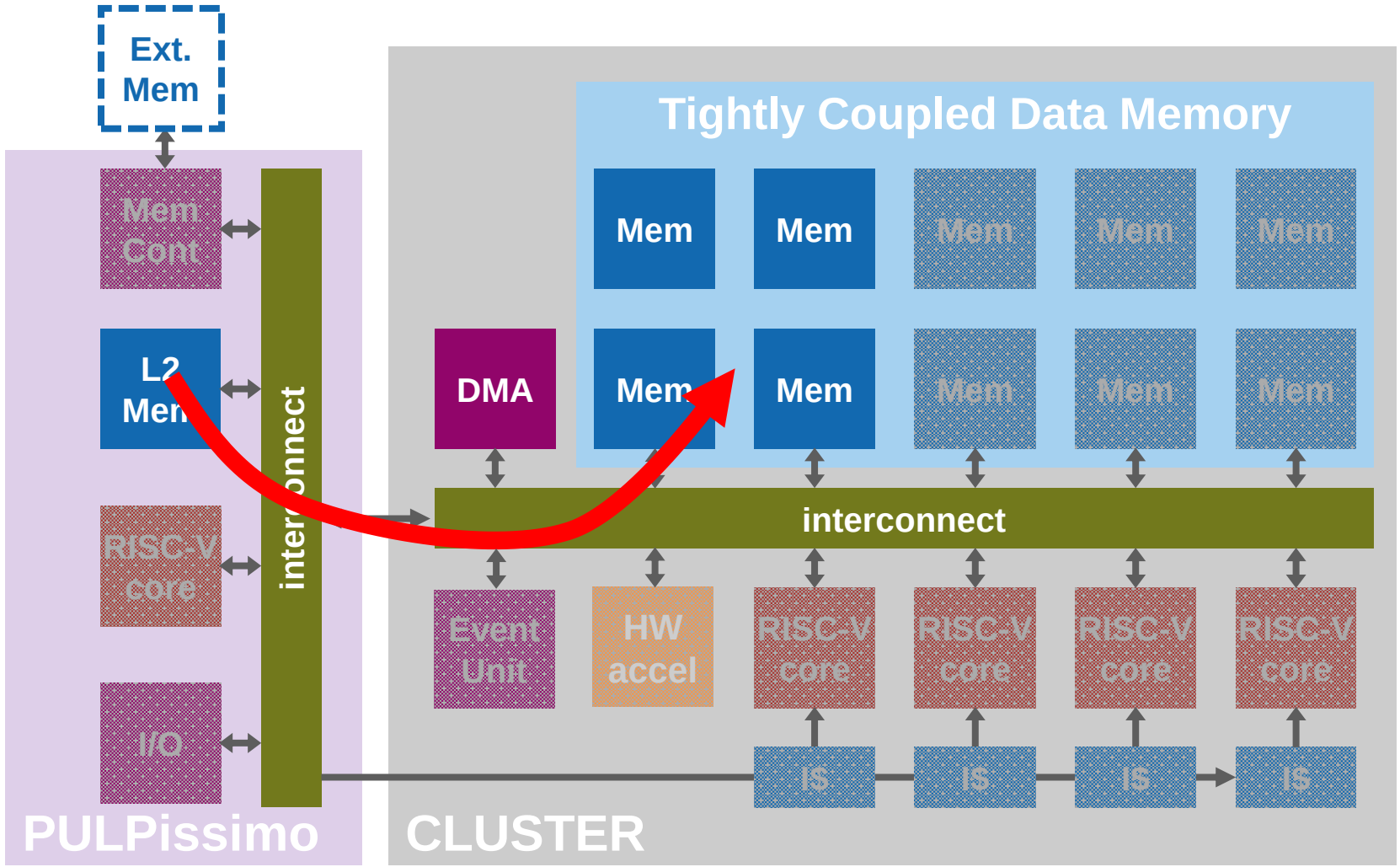
A microcontroller system (PULPissimo) for I/O



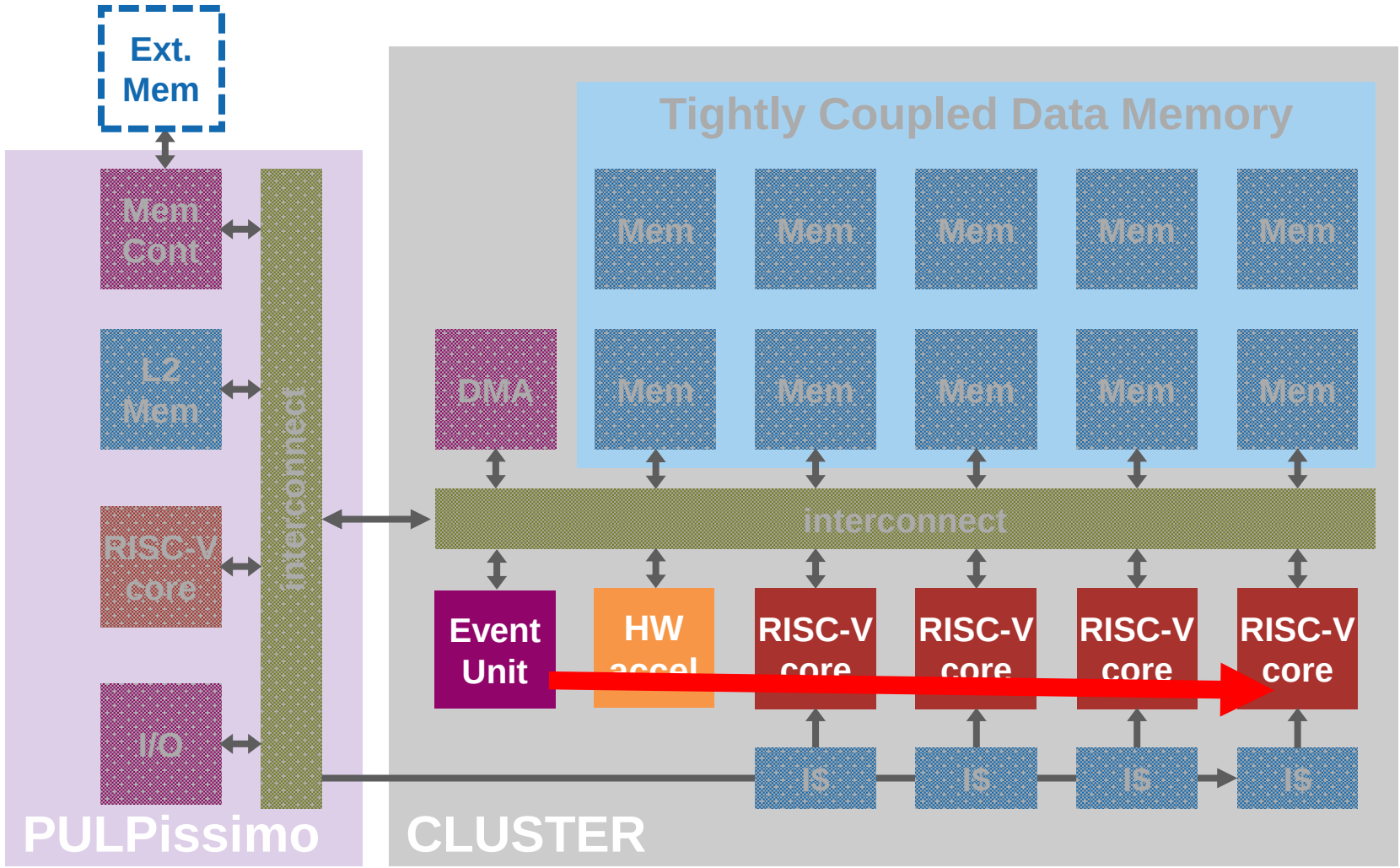
How do we work: Initiate a DMA transfer



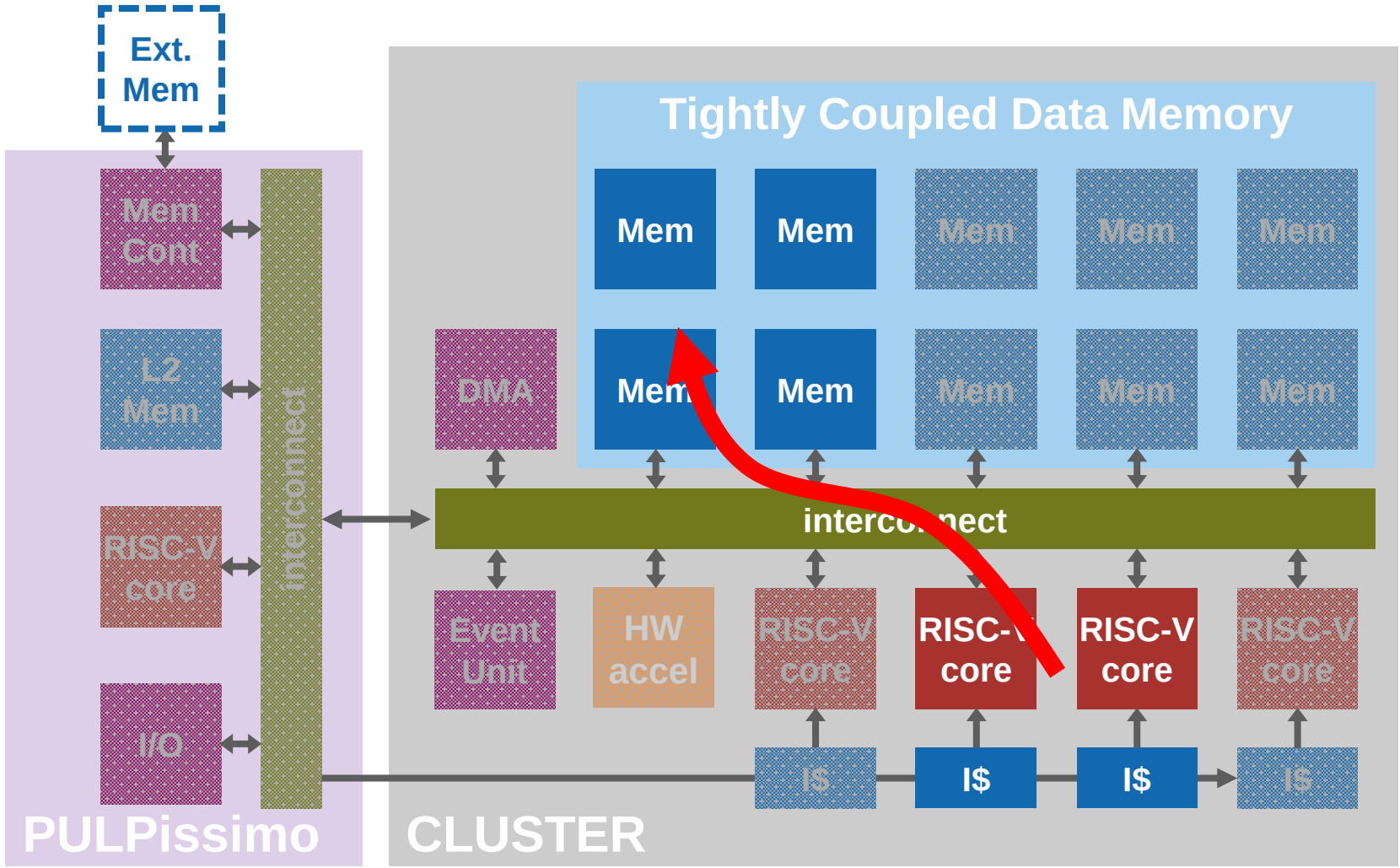
Data copied from L2 into TCDM



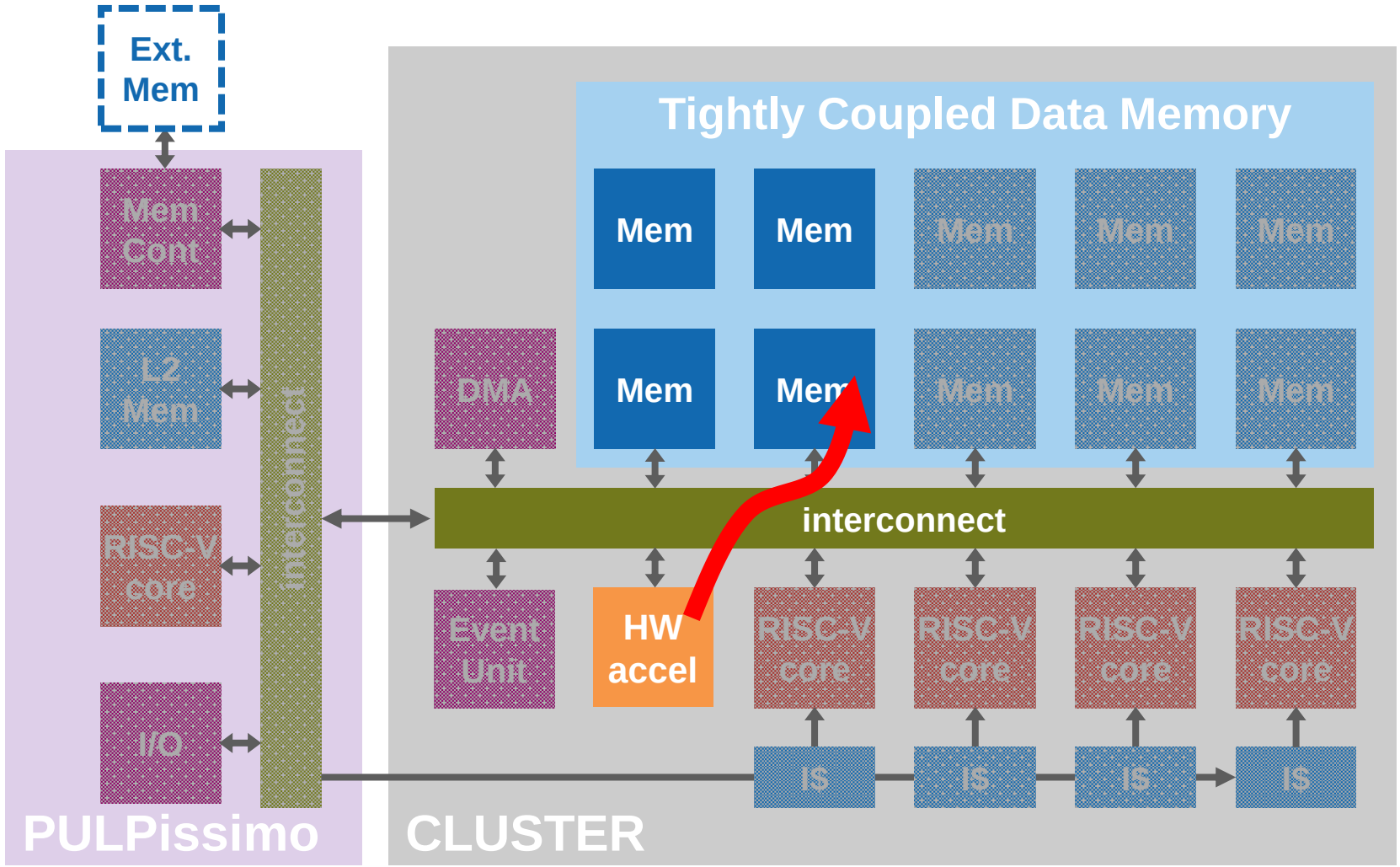
Once data is transferred, event unit notifies cores/accel



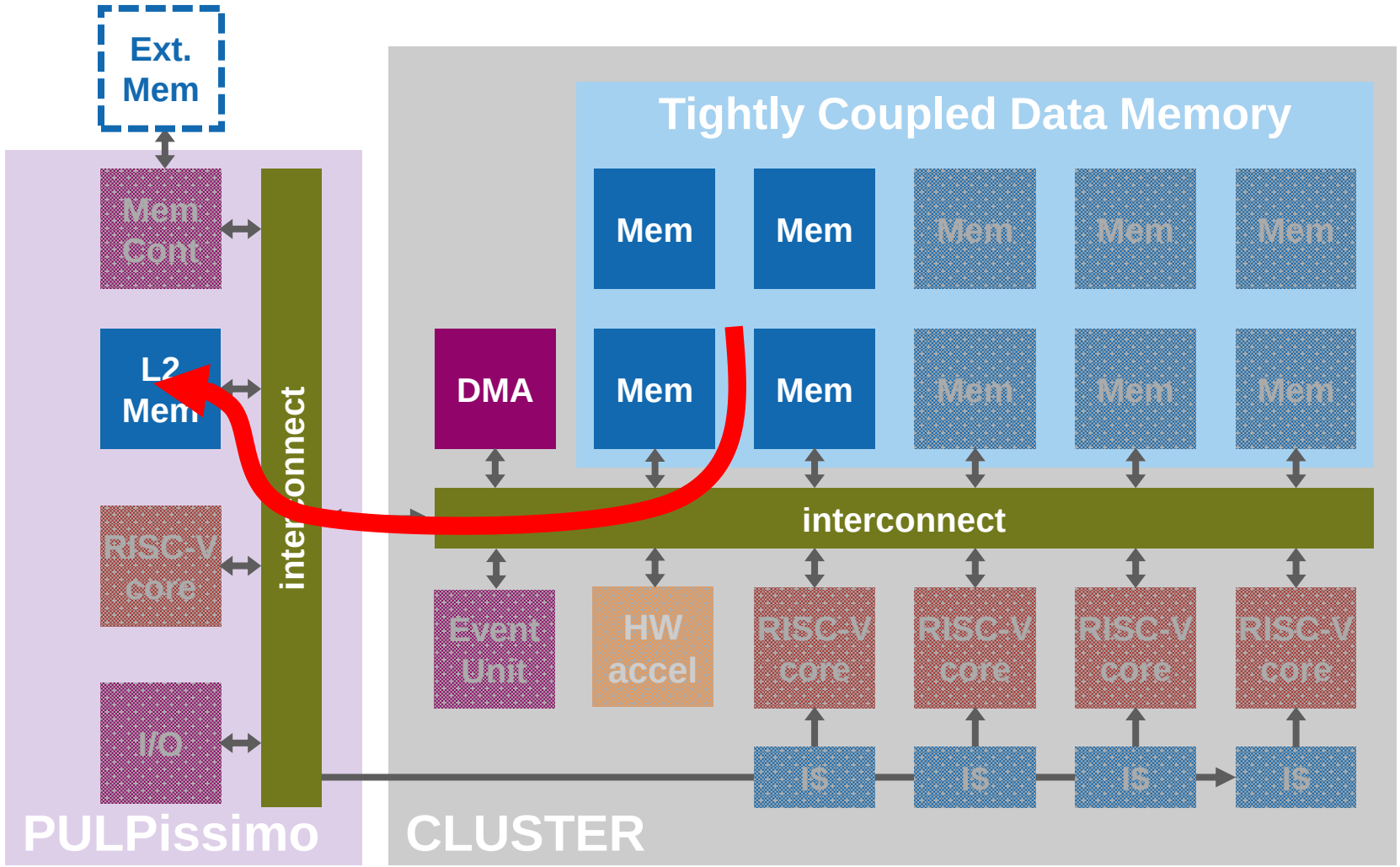
Cores can work on the data transferred



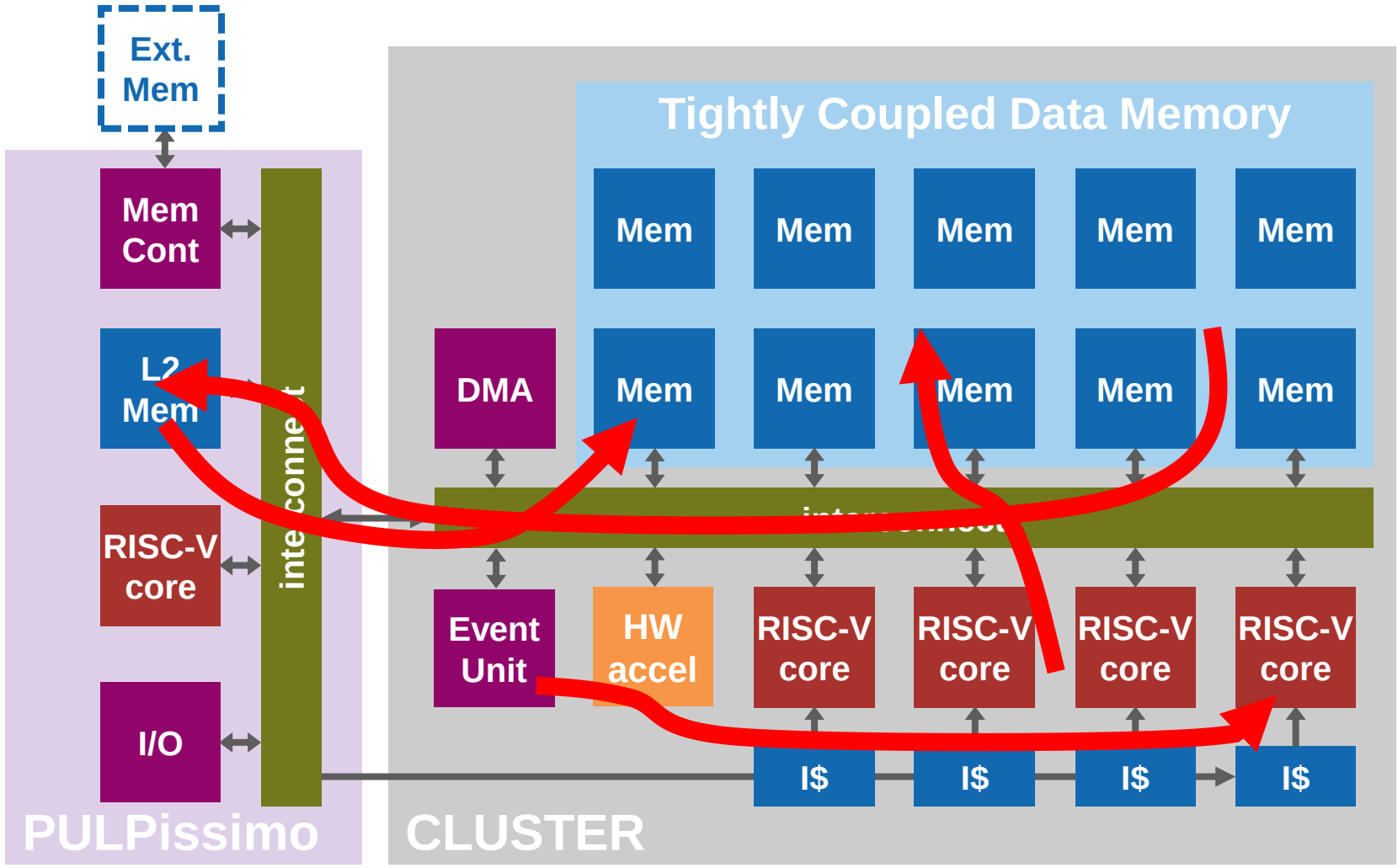
Accelerators can work on the same data



Once our work is done, DMA copies data back



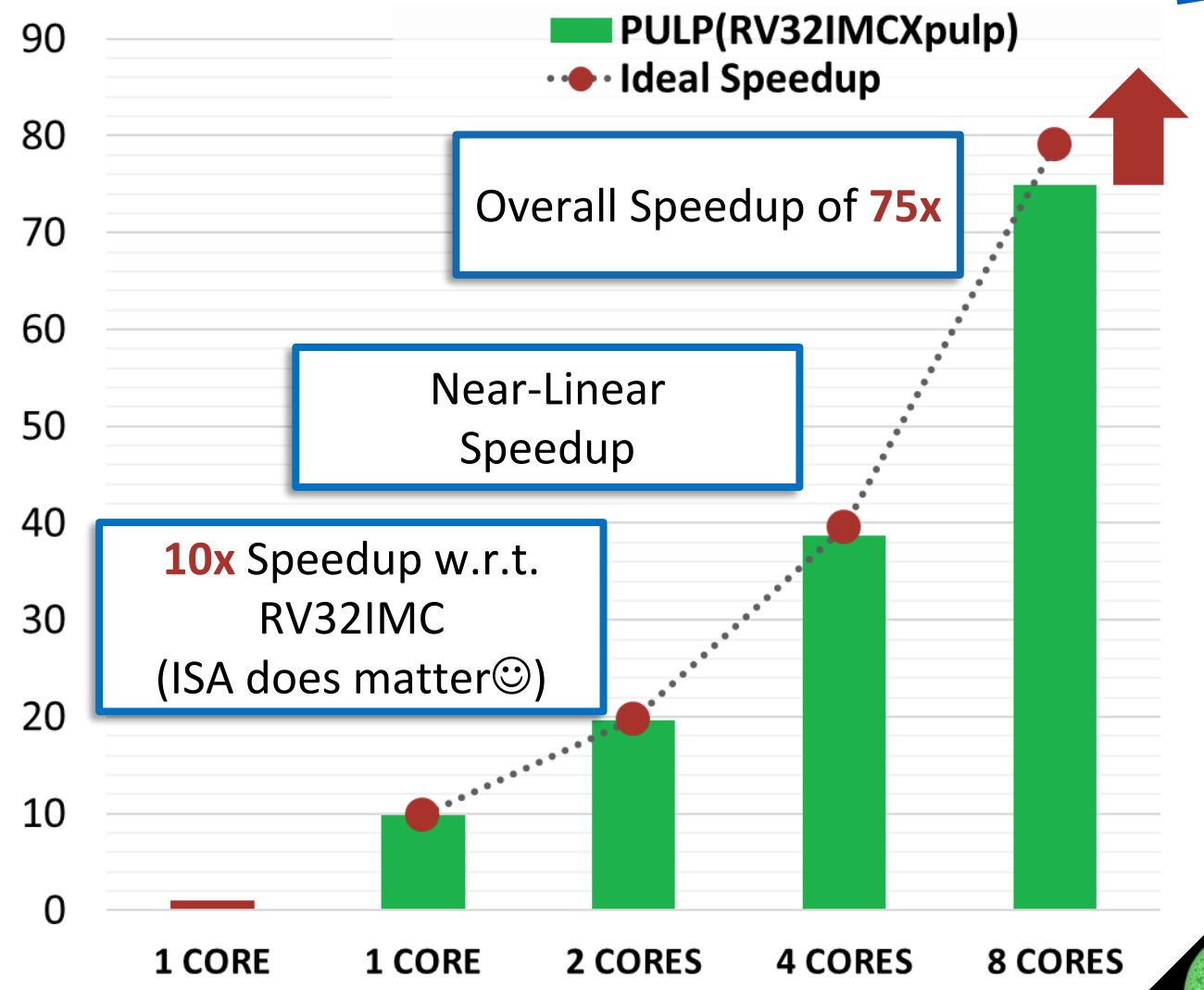
During normal operation all of these occur concurrently



Open-source: **github.com/pulp-platform/pulp**

Results: RV32IMCXpulp vs RV32IMC (DNN)

- **8-bit convolution**
  - Open source DNN library
- **10x through xPULP**
  - Extensions bring real speedup
- **Near-linear speedup**
  - Scales well for regular workloads.
- **75x overall gain**
  - 2 orders of magnitude with DOTP+LW (**122x**)
  - Sub-byte (nibble, crumb) supported (**537x**, **939x**)



PULP: Cores + Interco + IO + HWCE → Open Platform

HiPEAC

RISC-V Cores

RI5CY	Ibex	Snitch	Ariane + Ara
32b	32b	32b	64b

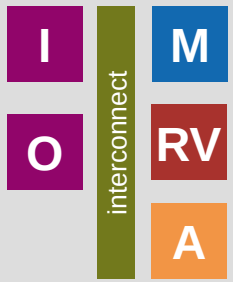
Peripherals

JTAG	SPI
UART	I2S
DMA	GPIO

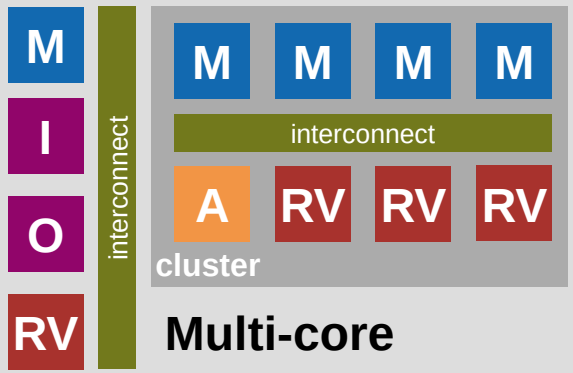
Interconnect

- Logarithmic interconnect
- APB – Peripheral Bus
- AXI4 – Interconnect

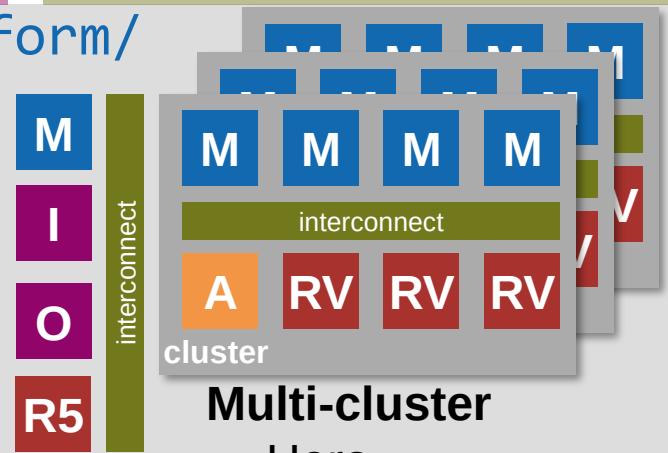
Platforms https://github.com/pulp-platform/



Single Core
- PULPino
- PULPissimo



Multi-core
- Open-PULP
- PULP-PM



Multi-cluster
- Hero
- MANTICORE

IOT → HPC

Accelerators

ML-HWCE (inference)	Neurostream (ML training)	HWCrypt (crypto)	PULPO (1st ord. opt)

ETH zürich

FRACTAL

How can we leverage this open-source platform to build reliable systems?

ETH zürich

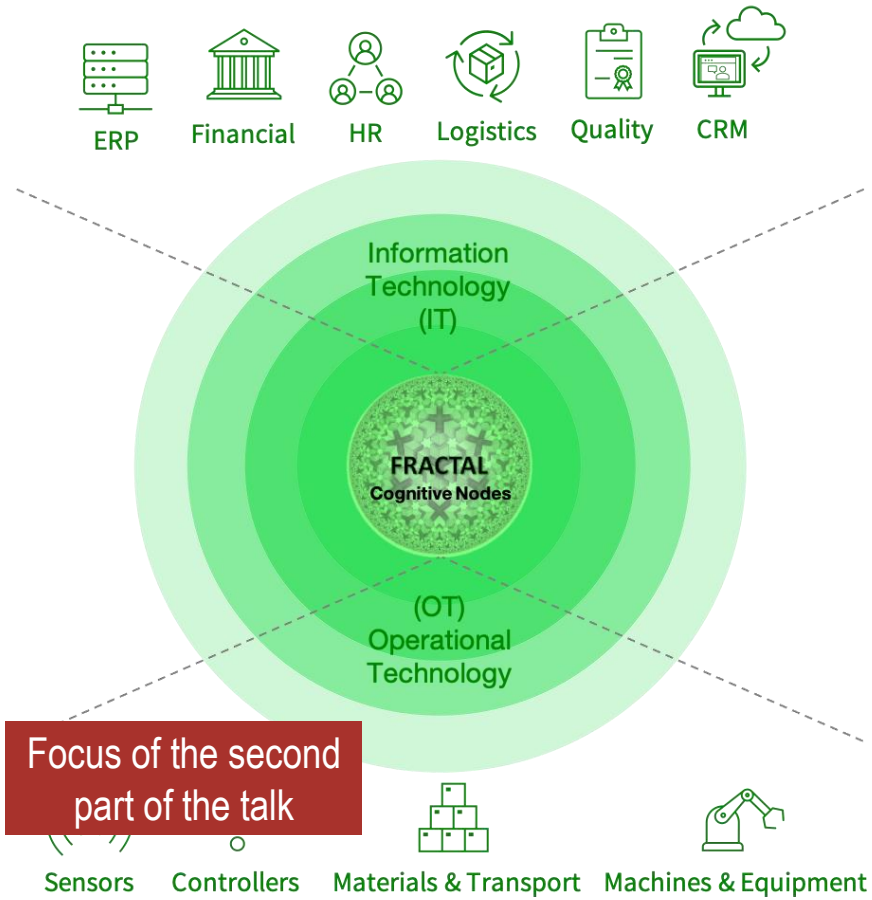
Agenda

- *Luca Bertaccini (ETHZ): "PULP: An Energy-Efficient Open-Source RISC-V Based IoT End Node"*
- ***Michael Rogenmoser (ETHZ): "Adding Reliability Features to PULP Systems"***

A Low-Power Fractal End Node

A reliable cognitive computing node:

- AI capabilities in the power envelope of an MCU
- Reliability features on the edge device



Focus of the second part of the talk

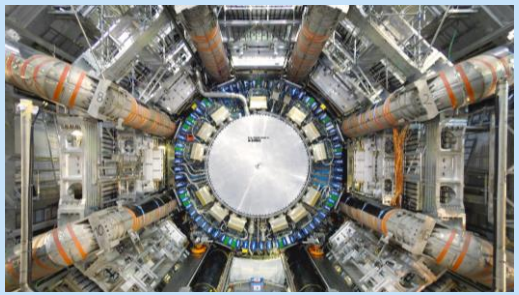
https://fractal-project.eu

Supported in part by the European Union's H2020 Fractal #877056

Critical & Hostile Environments

Space Environment

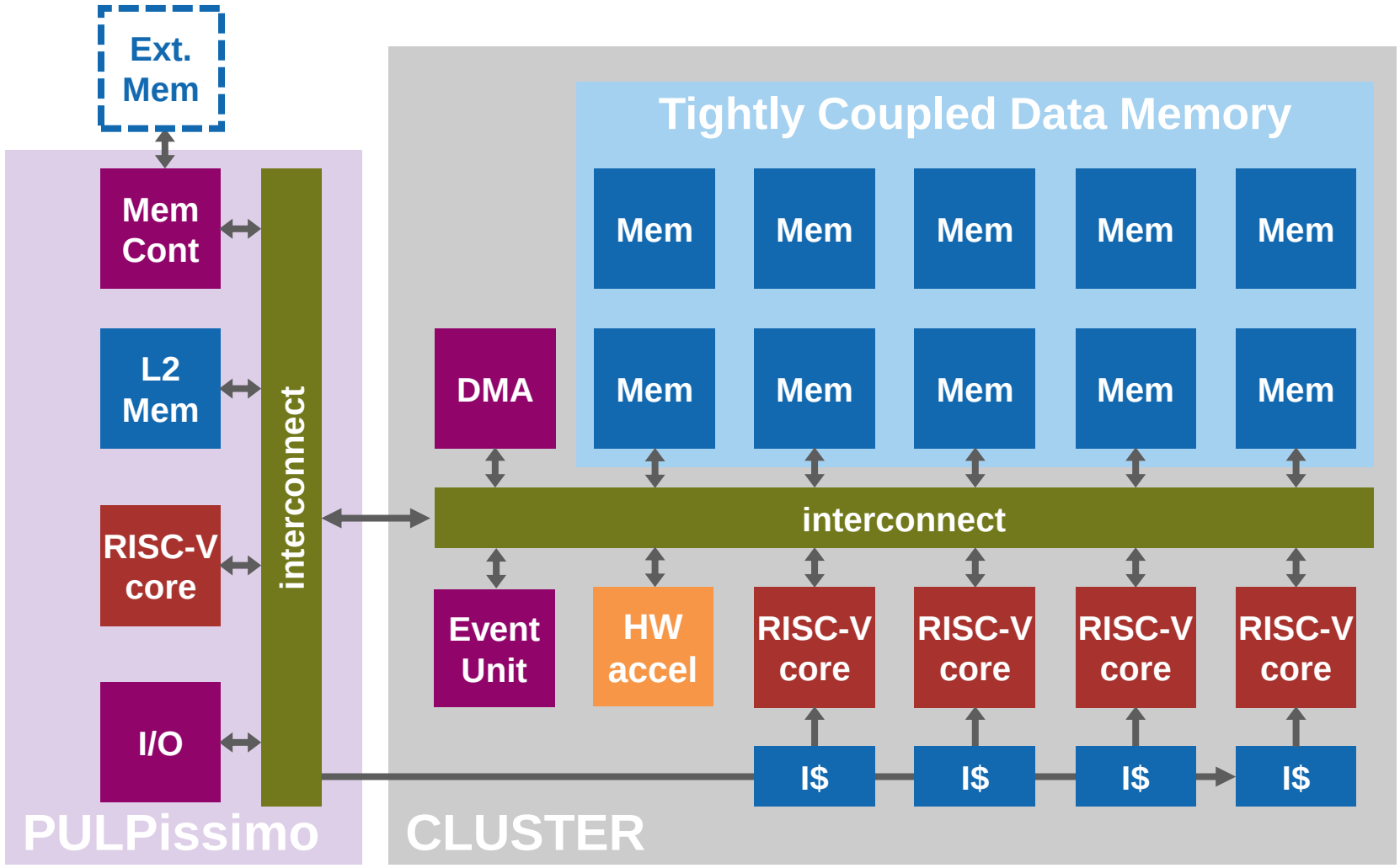
Automotive



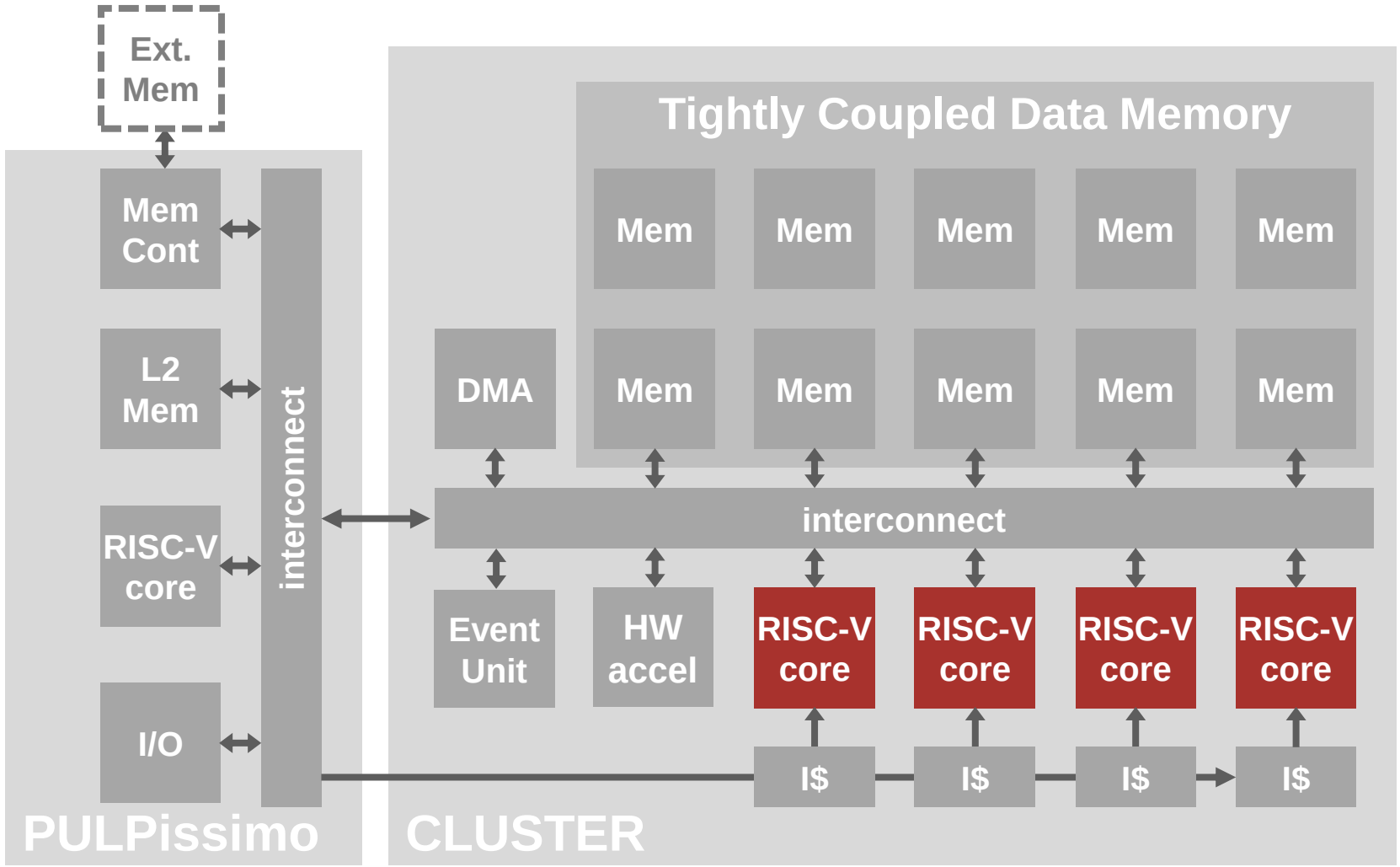
Particle Accelerator

And many more…

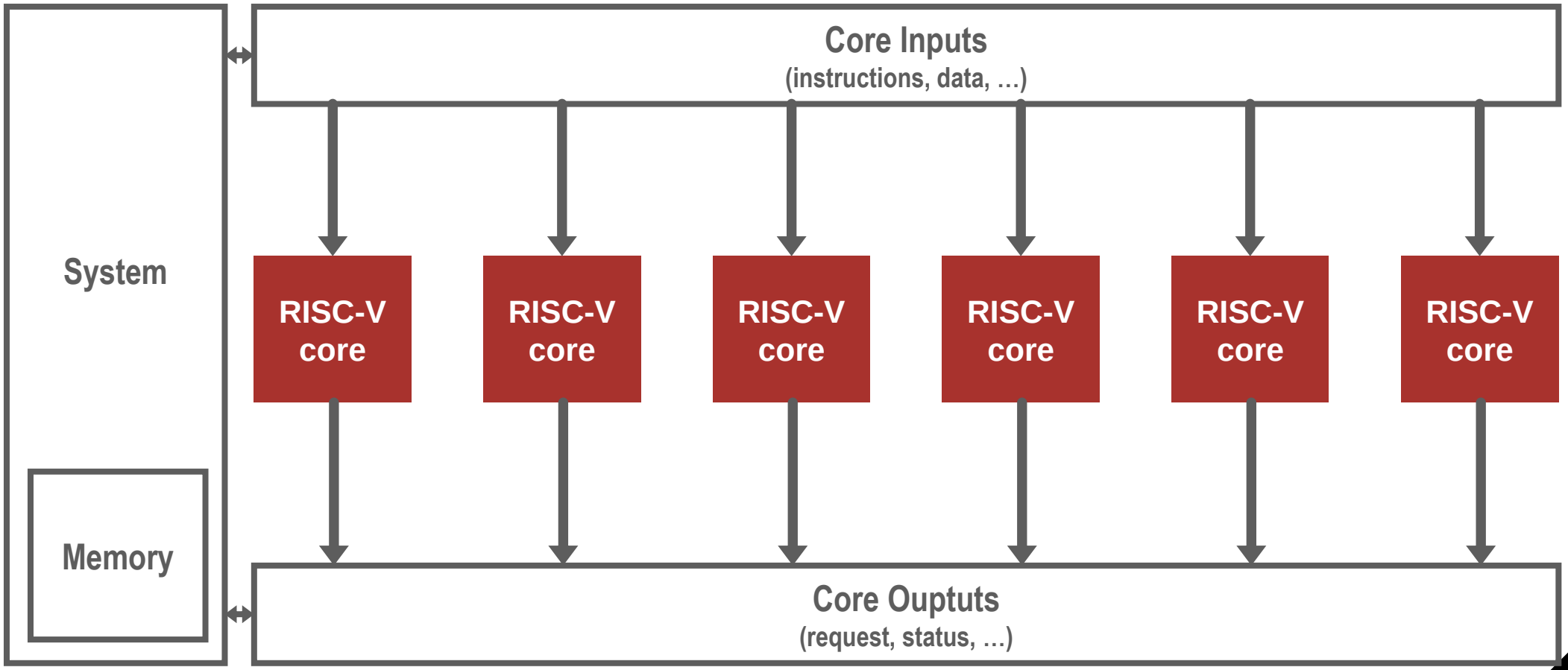
Leverage the PULP cluster



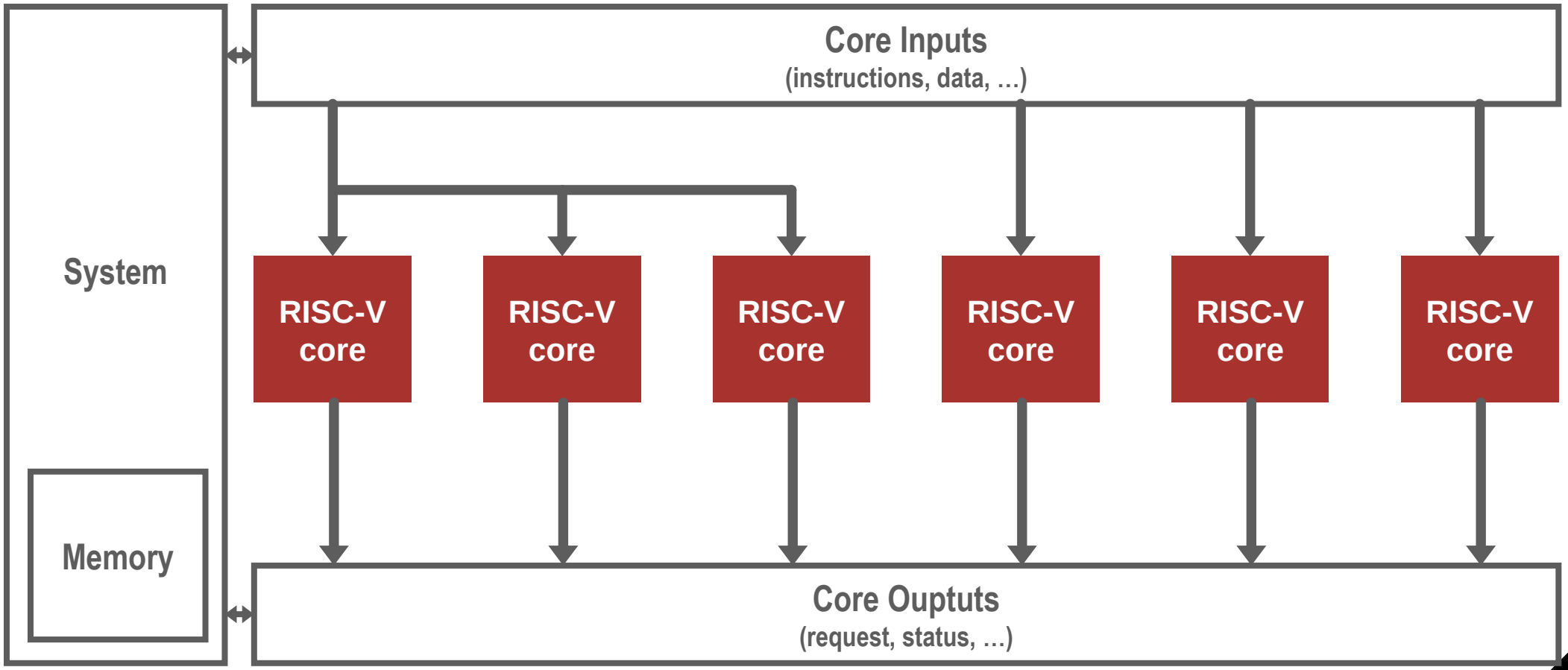
Leverage the PULP cluster



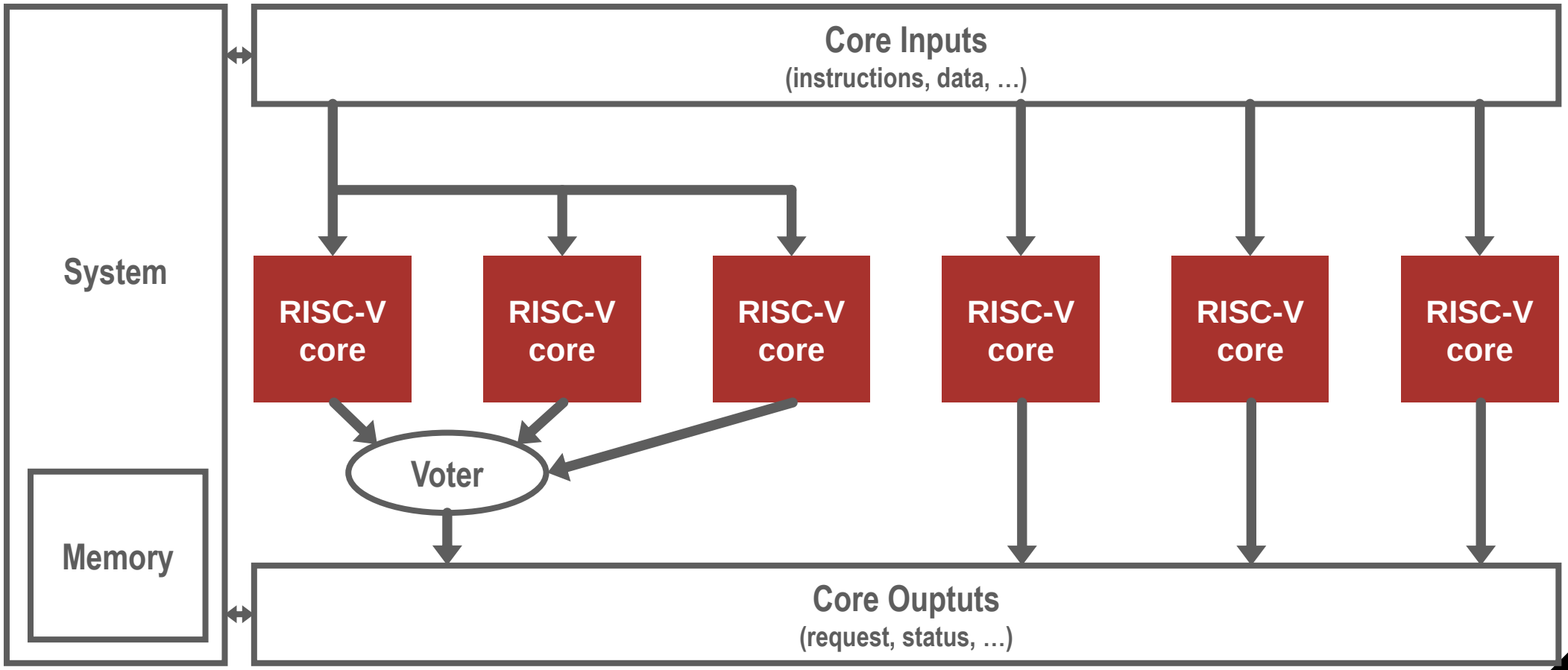
Protecting the Cores



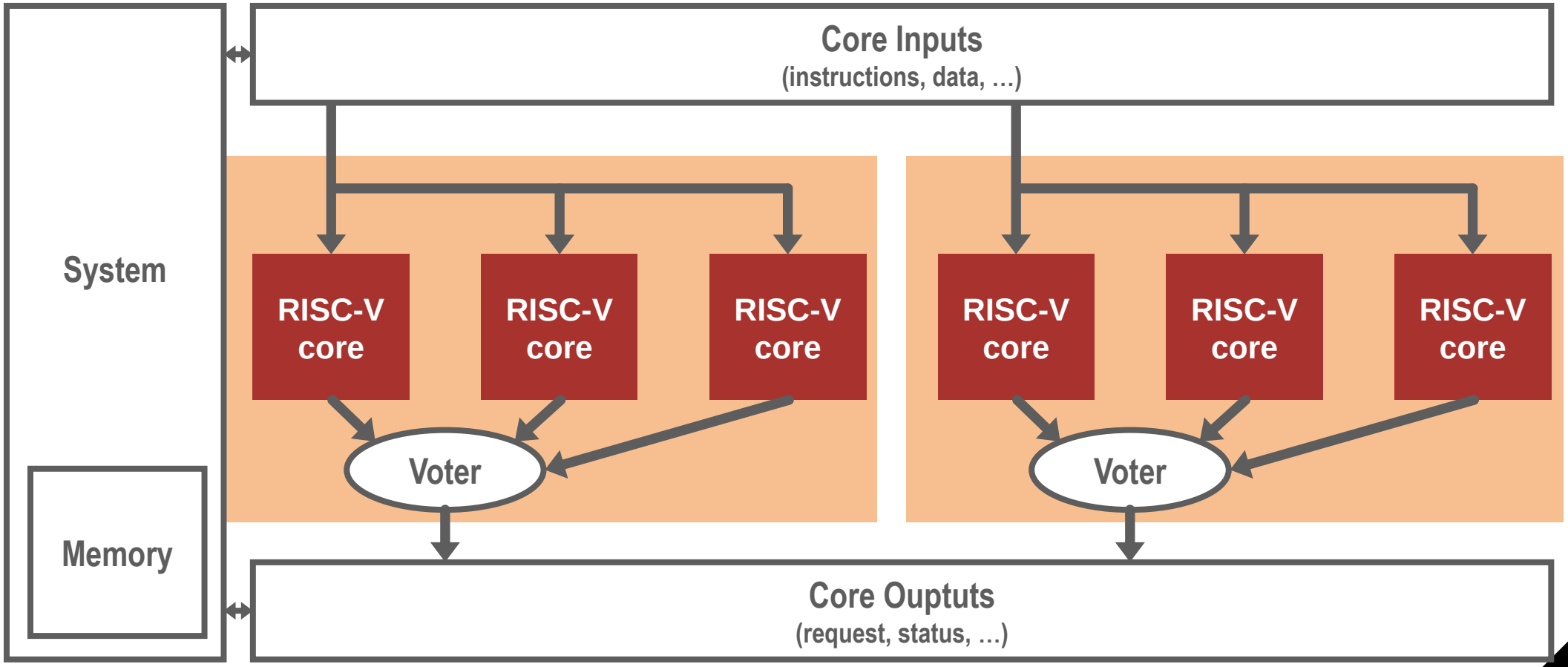
Protecting the Cores



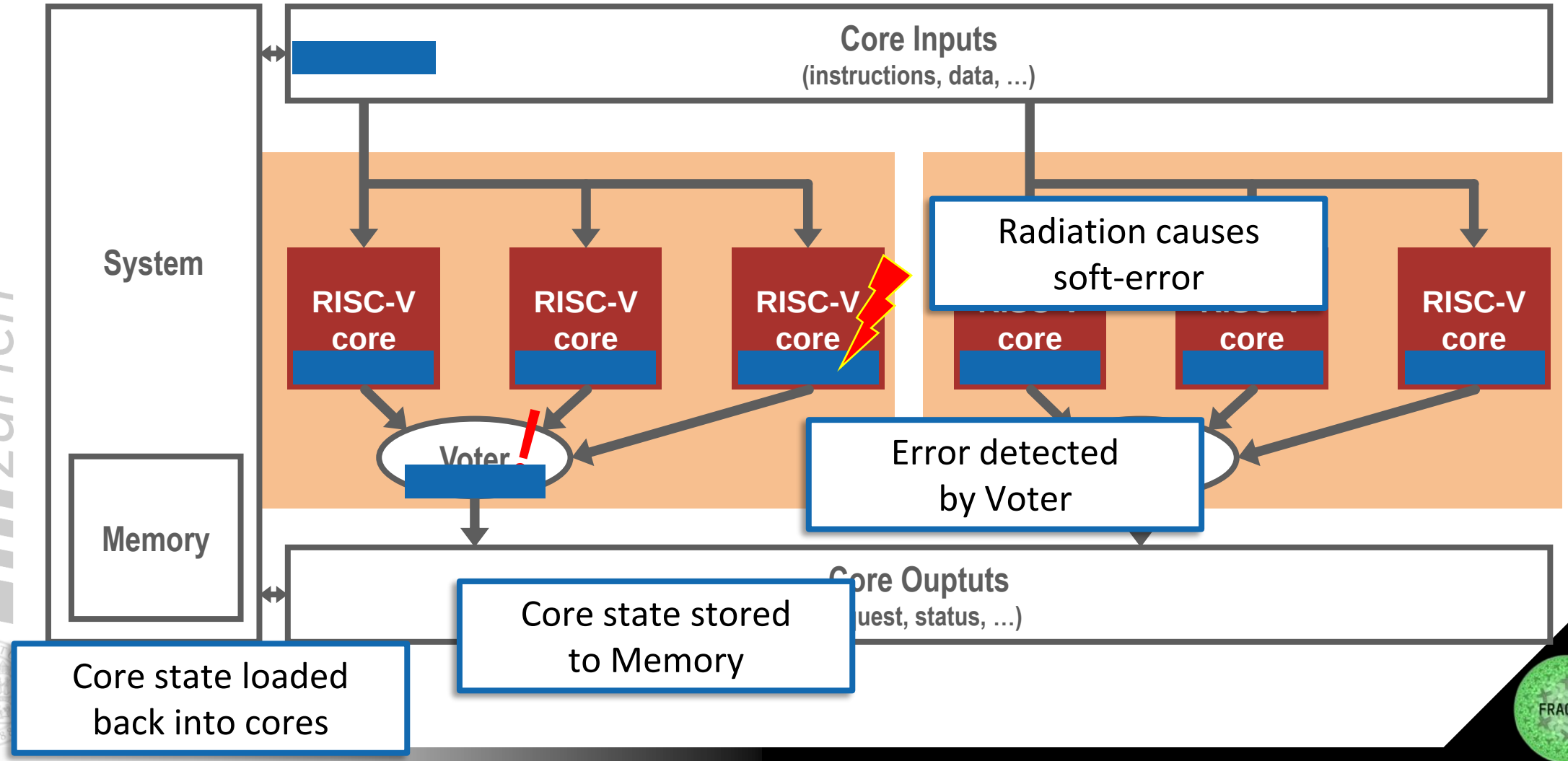
Protecting the Cores



Triple-Core Lock Step

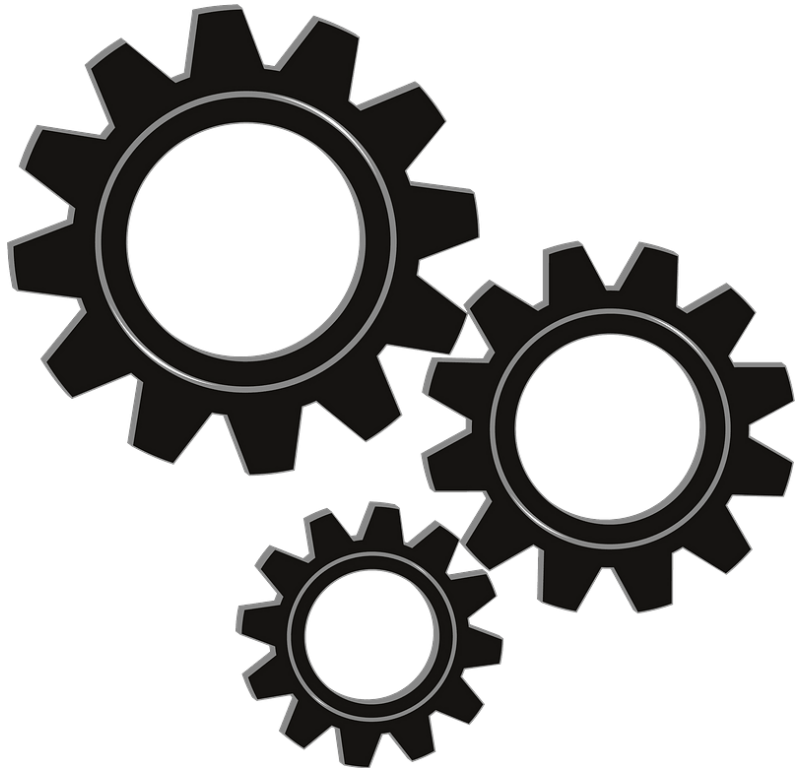


Re-synchronization



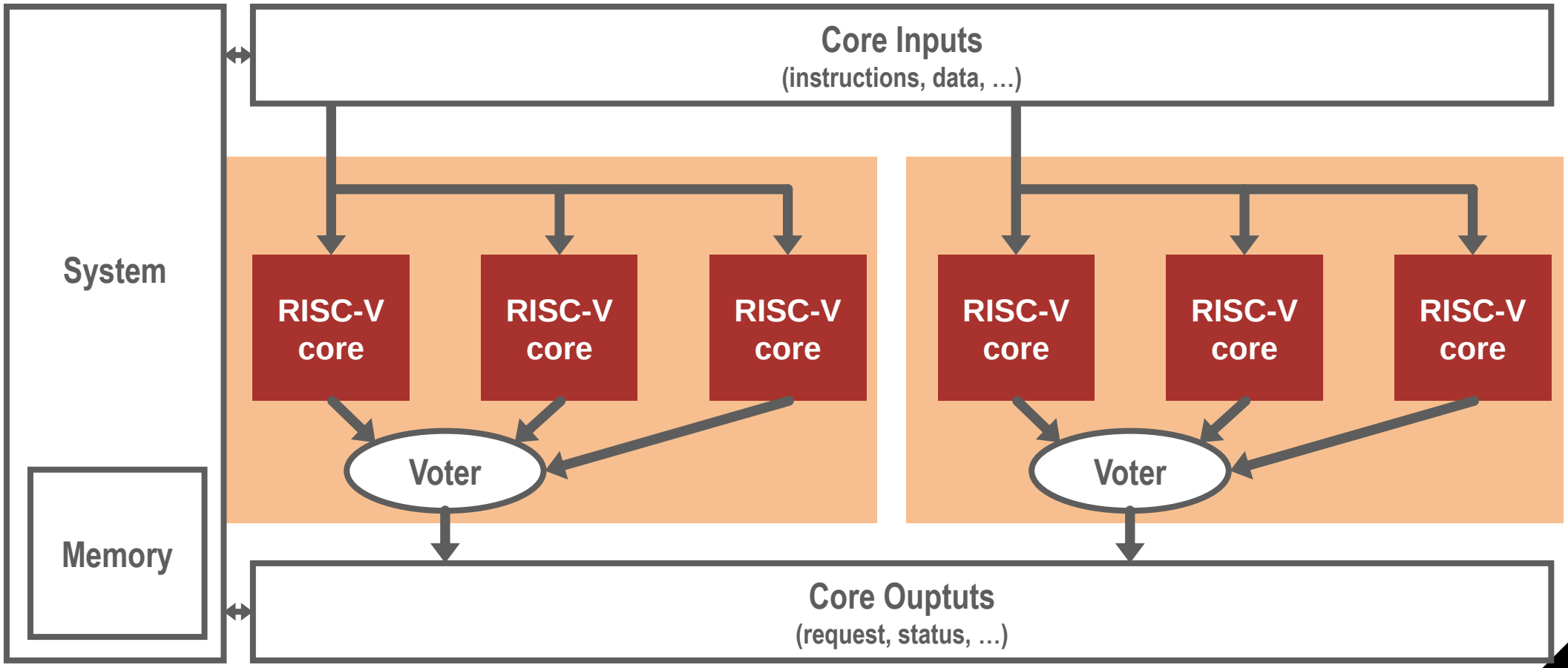
Redundancy Requirement

- **Mission-Critical Application**
  - Must be correct

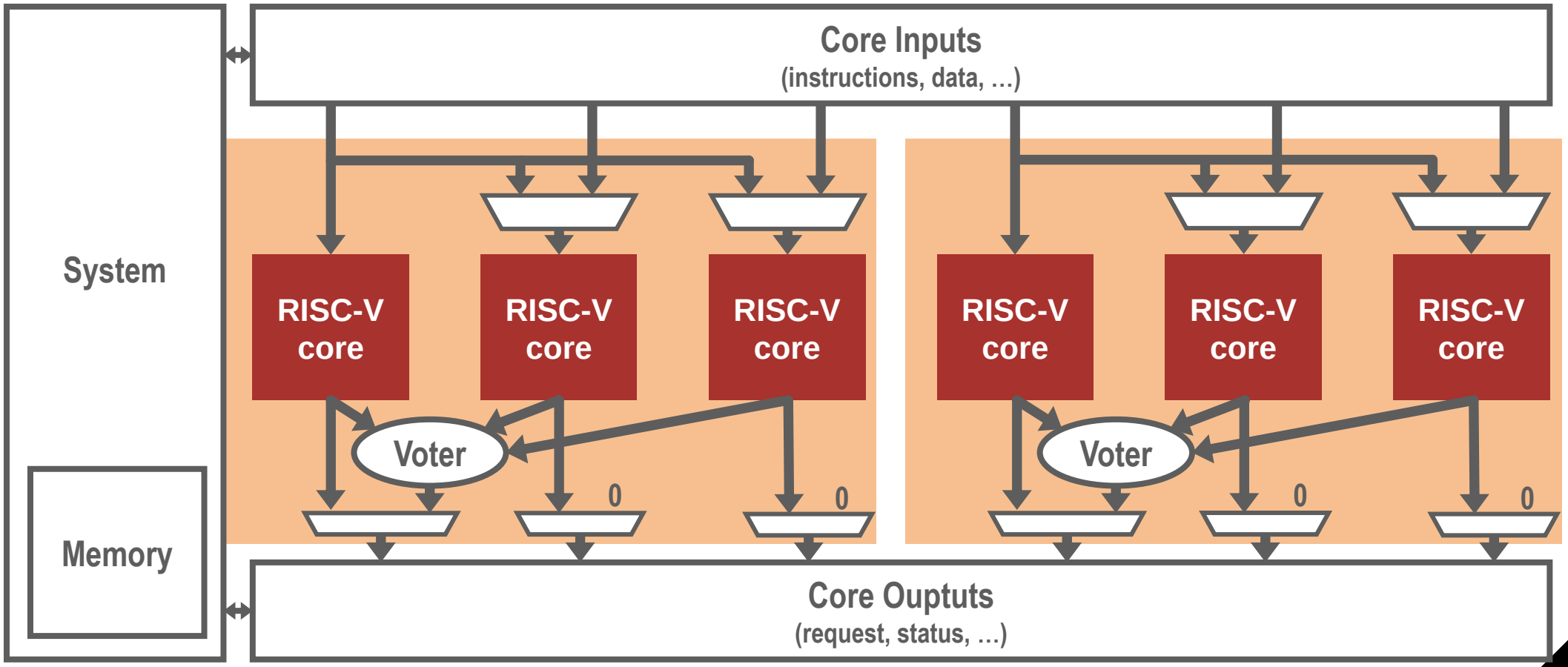


- **Data Processing Algorithm**
  - Individual errors can be tolerated

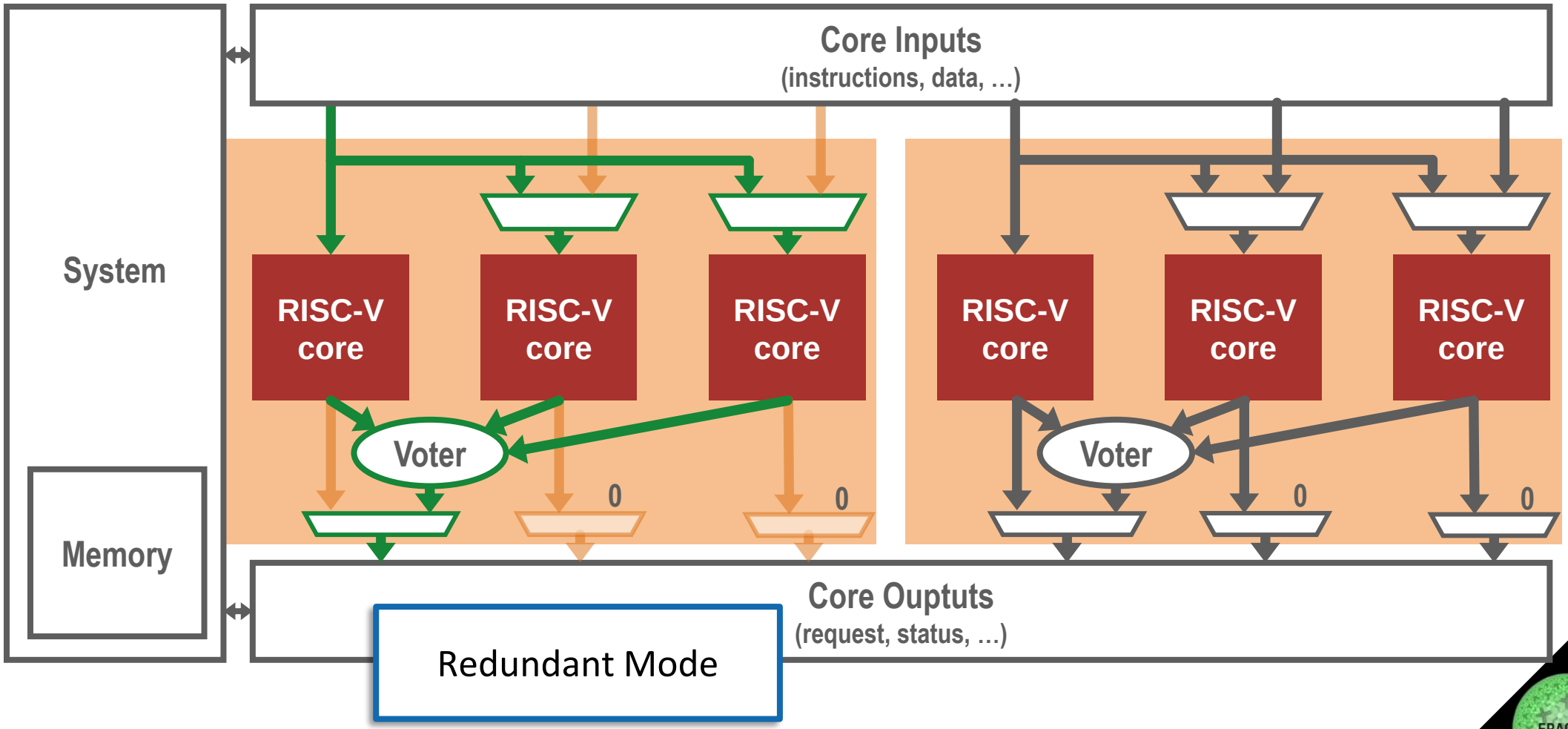
On-Demand Redundancy Grouping



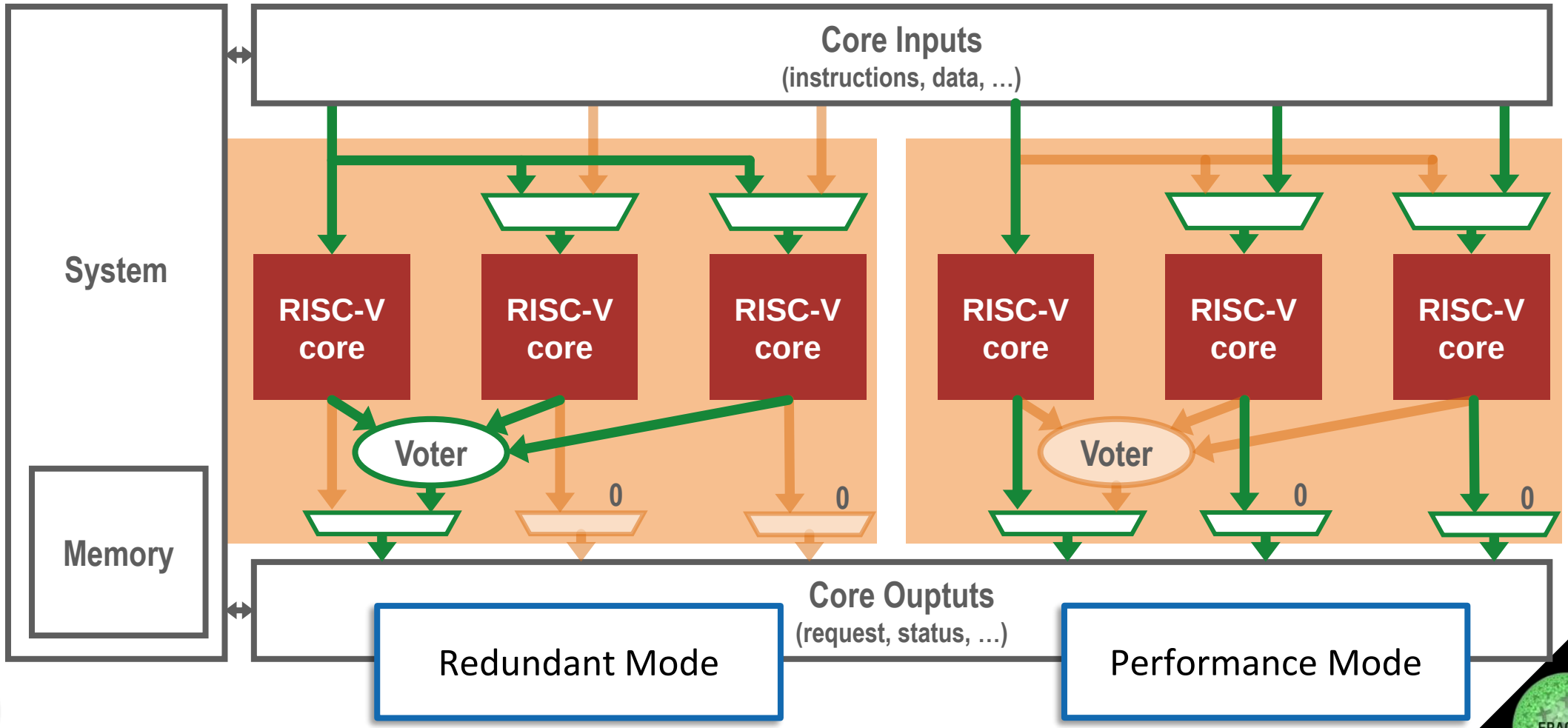
On-Demand Redundancy Grouping



On-Demand Redundancy Grouping

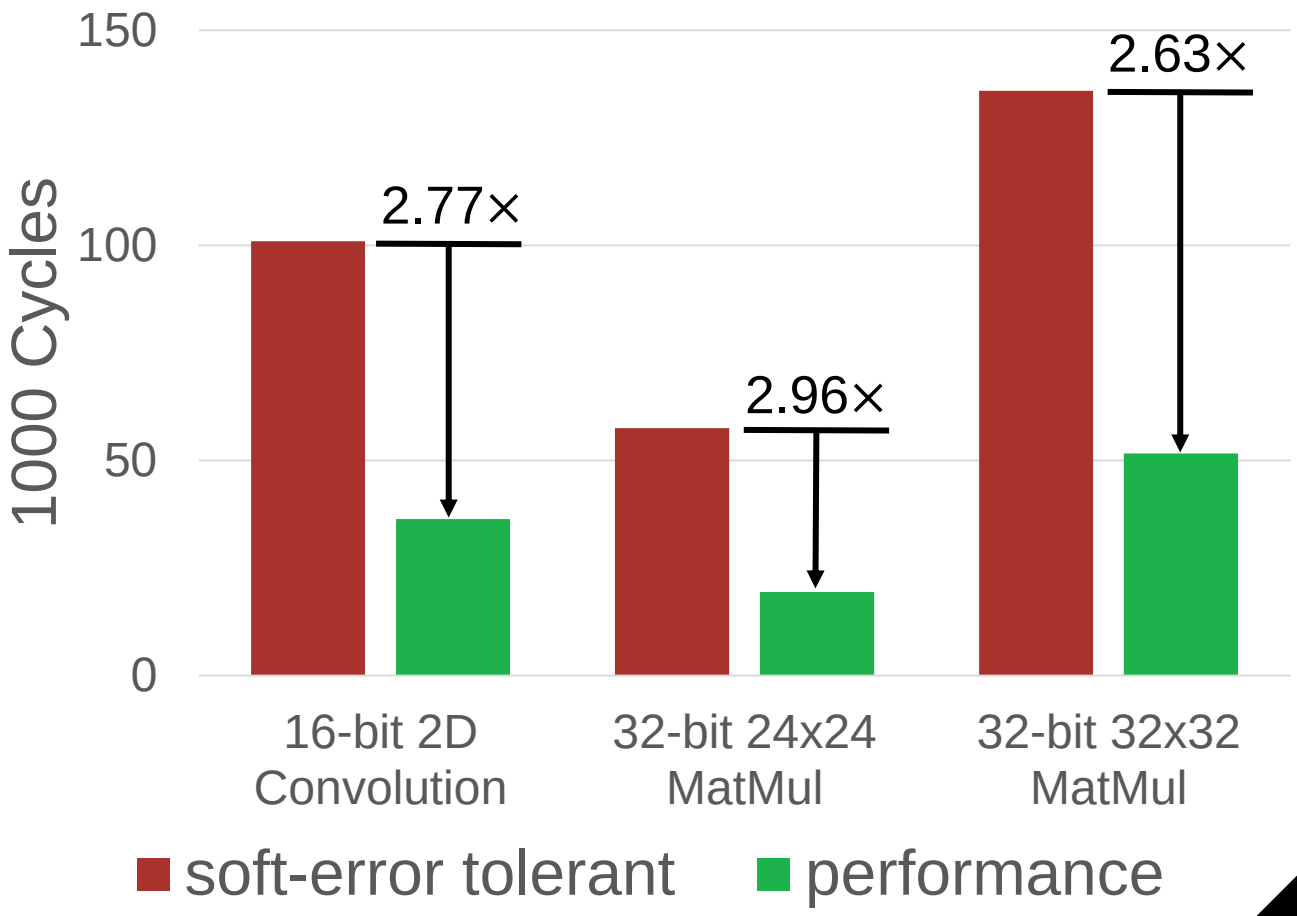


On-Demand Redundancy Grouping

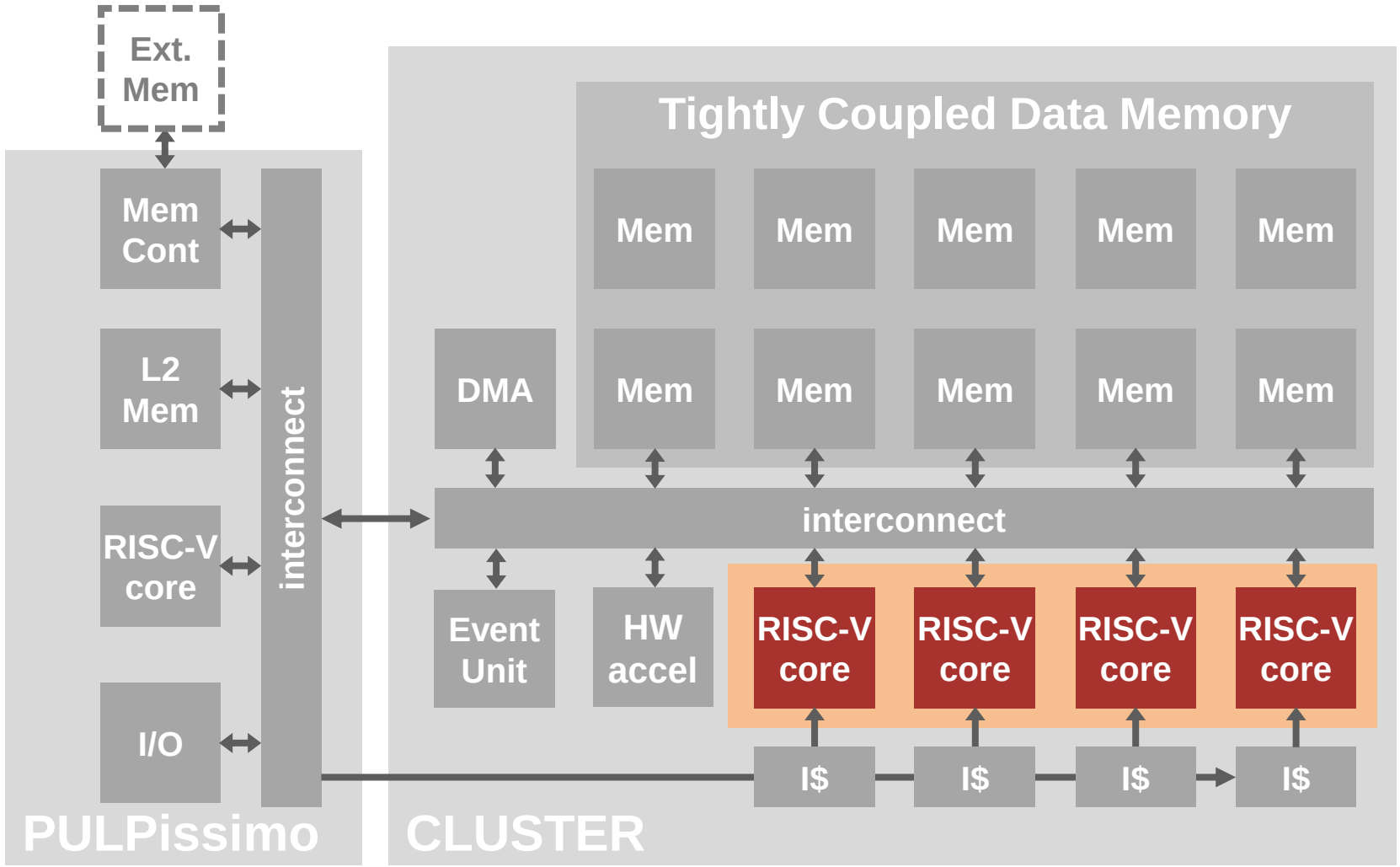


On-Demand Redundancy Grouping

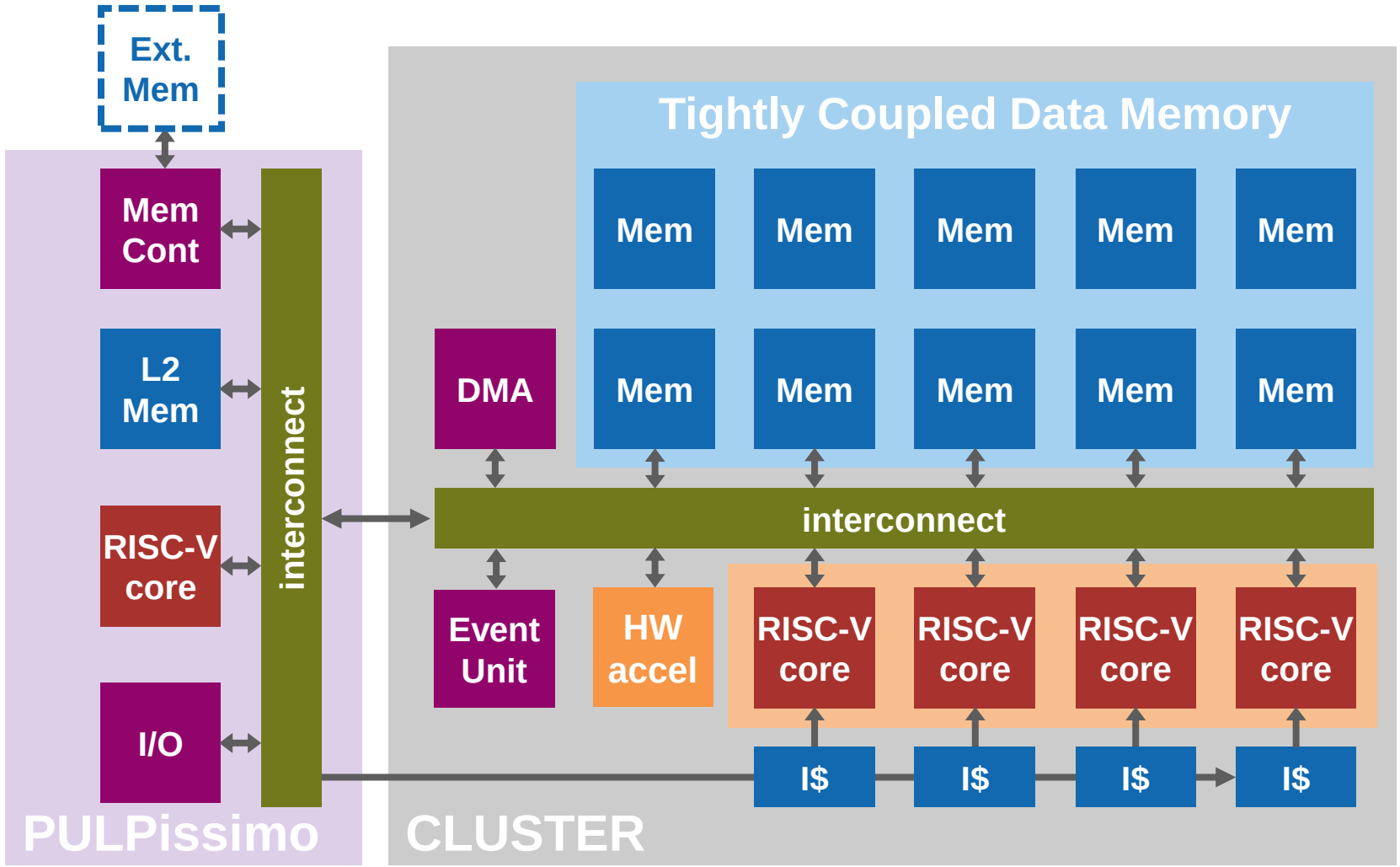
- Up to **3x speedup** for non-critical tasks
- **<60'000 cycles** to switch modes
- **<1% area overhead** for the cluster
  - ODRG for 3 cores requires ~11% area of a single core



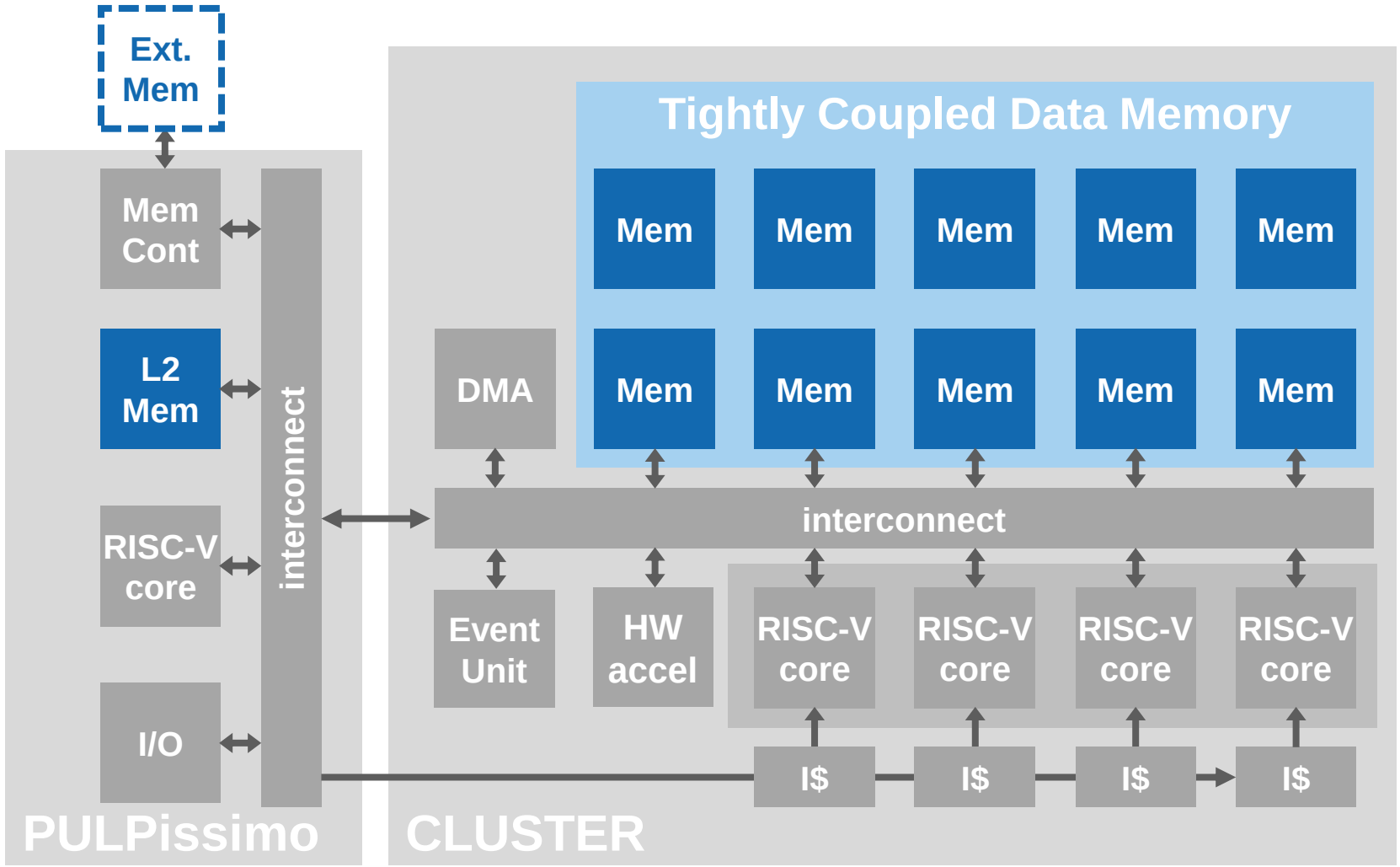
Leverage the PULP cluster



Leverage the PULP cluster



Leverage the PULP cluster

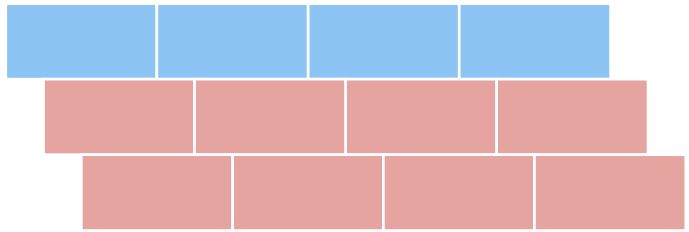


Memory Protection Options

- No Protection

32-bit word

- TMR

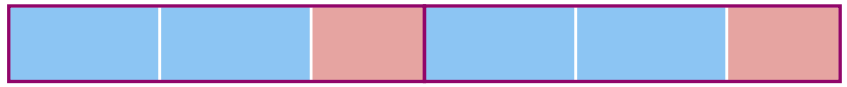


+200%

- **Hsiao Error Correcting Codes:**
- **Single Error Correction, Double Error Detection (SECDED)**
- 8-bit ECC

+62.5%

- 16-bit ECC



+37.5%

- 32-bit ECC

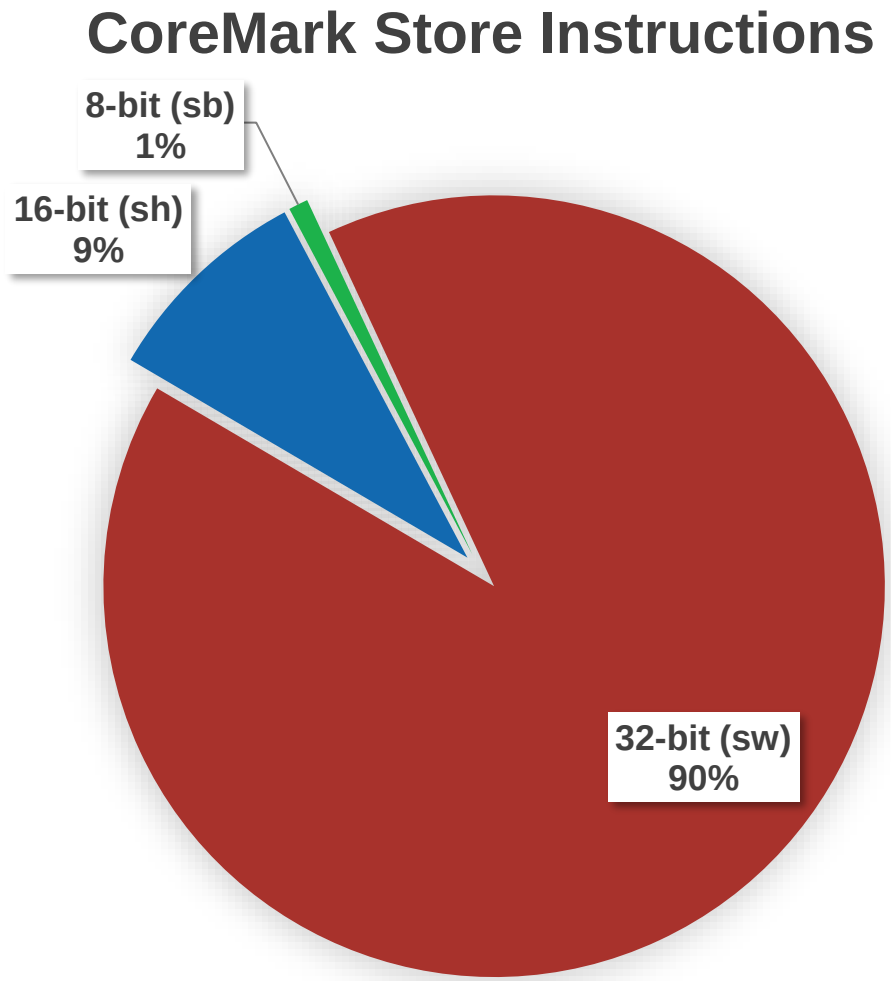
+21.9%

ECC Load-and-Store

Byte Store

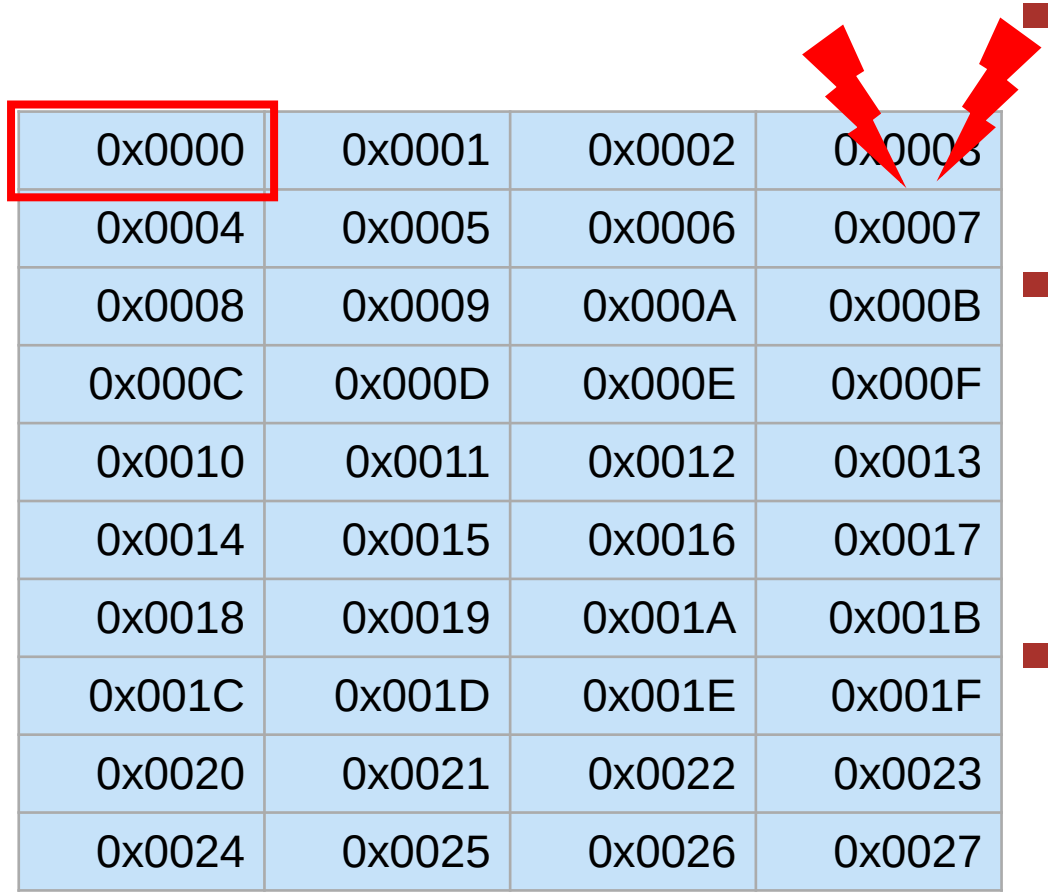
Byte Store with ECC

L1 Memory



- 32-bit ECC Used
- Buffer storage operation
  - Delay following transaction, not current transaction, to shift & reduce impact
- <1% cycle increase
  - Various tests, such as 8-bit Matrix-Matrix Multiplication

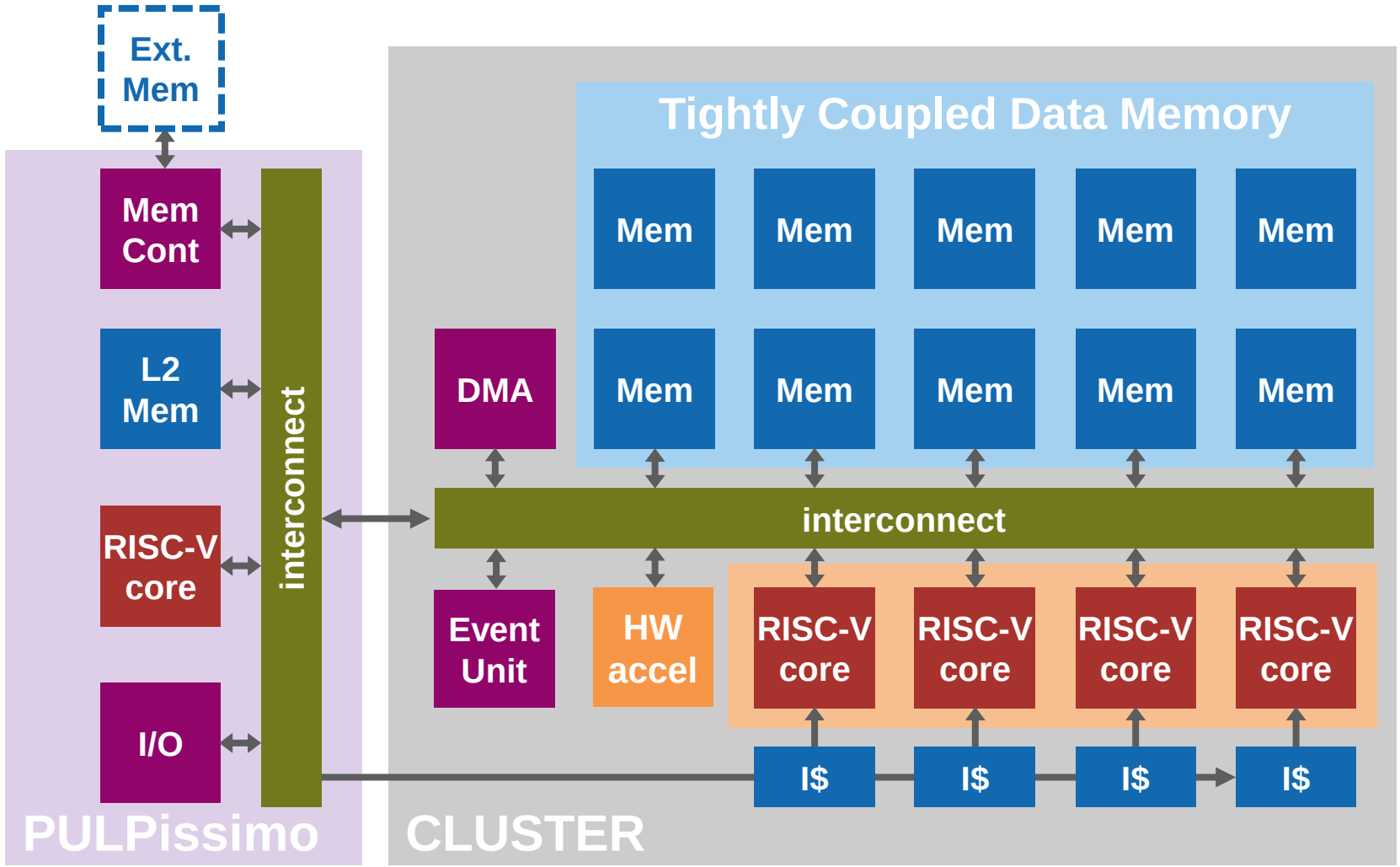
ECC Scrubber



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

- Scan Memory Bank
- Re-write faulty word if error is detected
- Defer permission to external accesses

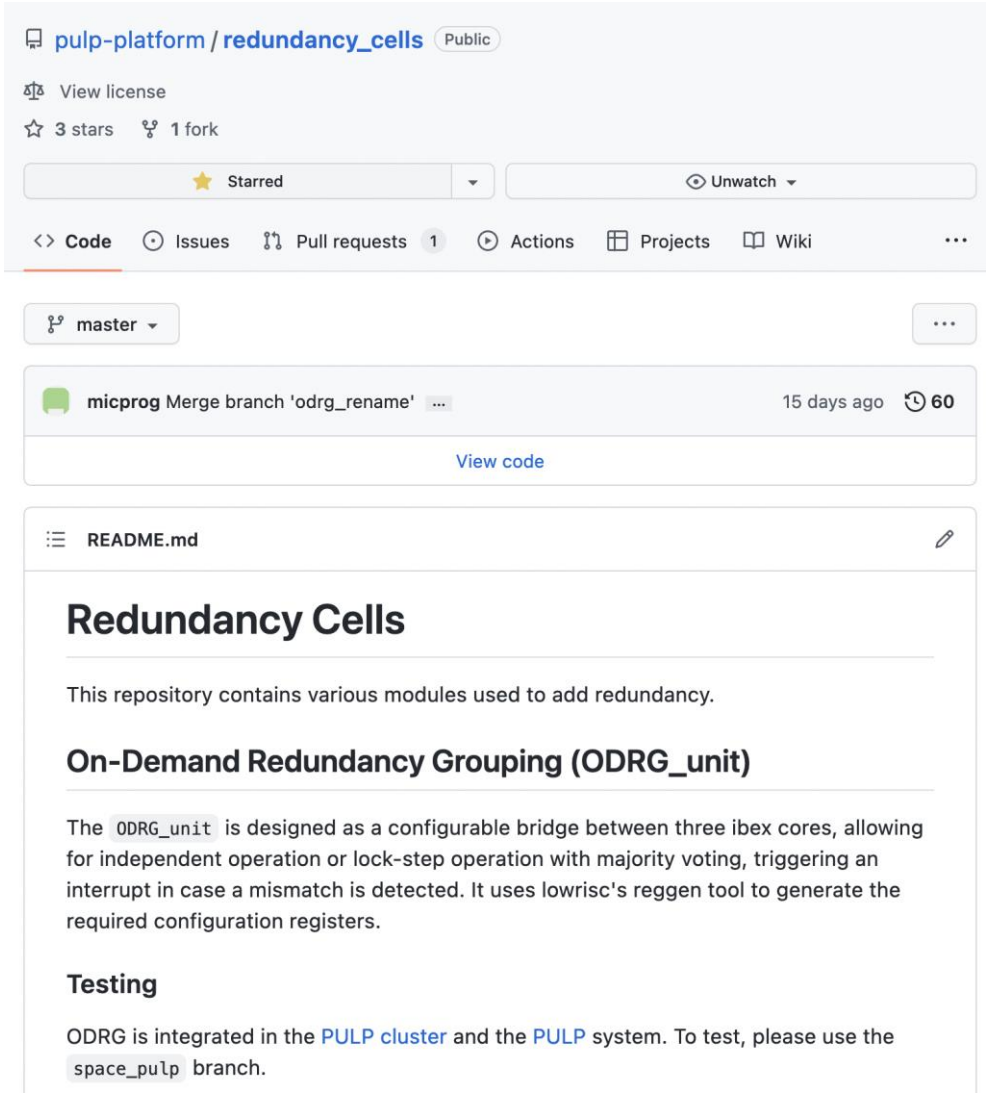
Leverage the PULP cluster



Ongoing Projects

- Fault-tolerant Host Processor
  - Triple-core PULPissimo
- Interconnect Protection
- Instruction Cache Protection
- Fault-tolerant Peripherals
- Adding a Watchdog Timer

Try it out on Github!



PULP

Parallel Ultra Low Power

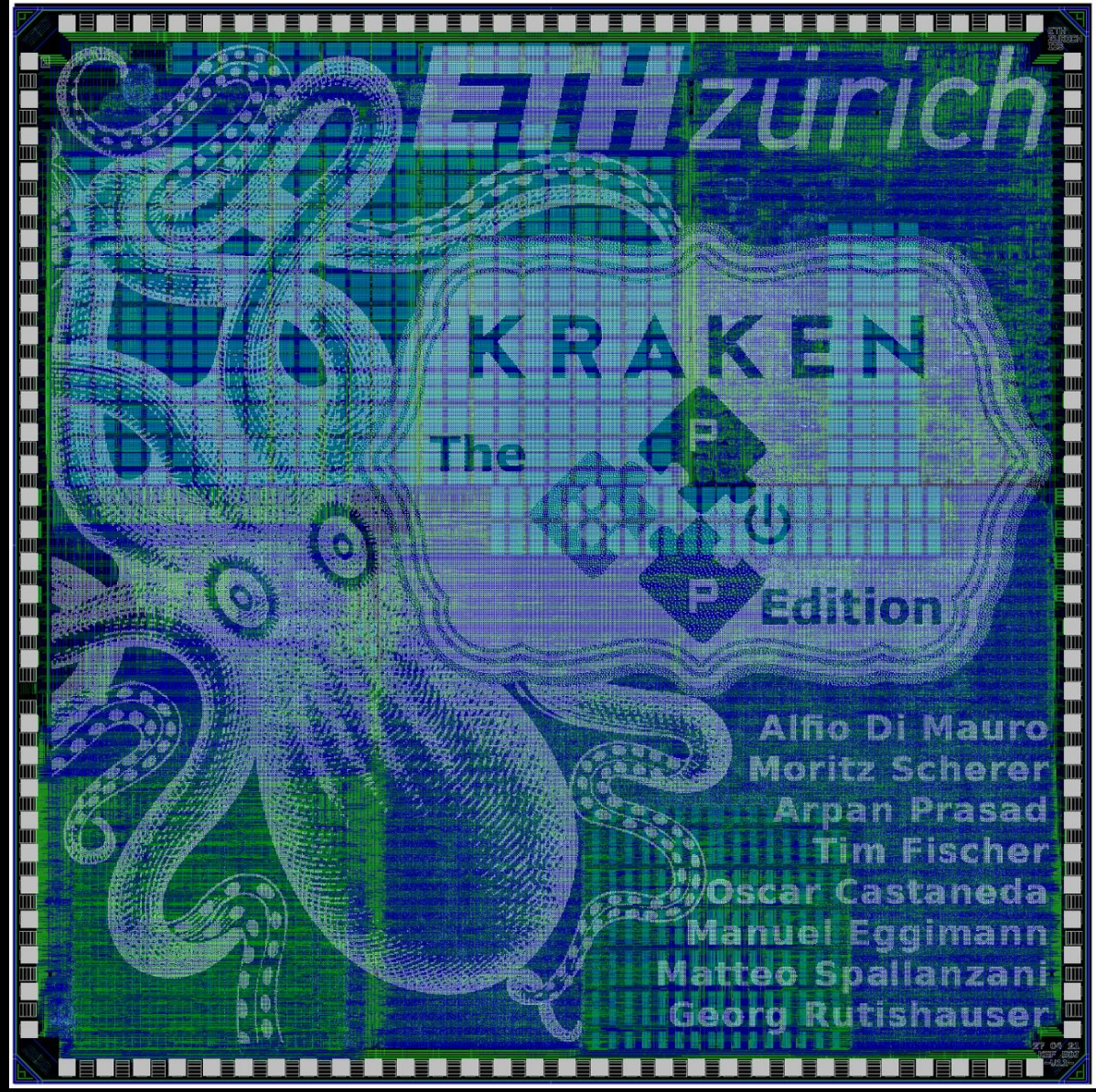
www.github.com/pulp-platform

ETH zürich

PULP

Parallel Ultra Low Power

Luca Benini, Alessandro Capotondi, Alessandro Ottaviano, Alessio Burrello, Alfio Di Mauro, Andrea Borghesi, Andrea Cossettini, Andreas Kurth, Angelo Garofalo, Antonio Pullini, Arpan Prasad, Bjoern Forsberg, Corrado Bonfanti, Cristian Cioflan, Daniele Palossi, Davide Rossi, Fabio Montagna, Florian Glaser, Florian Zaruba, Francesco Conti, Georg Rutishauser, Germain Haugou, Gianna Paulin, Giuseppe Tagliavini, Hanna Müller, Luca Bertaccini, Luca Valente, Luca Colagrande, Manuel Eggimann, Manuele Rusci, Marco Guermandi, Matheus Cavalcante, Matteo Perotti, Matteo Spallanzani, Michael Rogenmoser, Moritz Scherer, Moritz Schneider, Nazareno Bruschi, Nils Wistoff, Pasquale Davide Schiavone, Paul Scheffler, Philipp Mayer, Robert Balas, Samuel Riedel, Sergio Mazzola, Sergei Vostrikov, Simone Benatti, Stefan Mach, Thomas Benz, Thorir Ingolfsson, Tim Fischer, Victor Javier Kartsch Morinigo, Vlad Niculescu, Xiaying Wang, Yichao Zhang, Frank K. Gürkaynak,
all our past collaborators and many more that we forgot to mention



http://pulp-platform.org @pulp_platform