



PULP
Parallel Ultra Low Power

PULP: an Open Hardware Platform

The story so far

HPCA 2018 - Vienna

25.02.2018



Frank K. Gürkaynak

Florian Zaruba

Andreas Kurth

Francesco Conti



Multitherman

 **PRECOMP**
Open Transprecision Computing

 **ExaNode**

¹Department of Electrical, Electronic
and Information Engineering

ETH zürich

²Integrated Systems Laboratory

<http://pulp-platform.org>

A word from our sponsors

This workshop has been supported by

- ERC-Advanced grant **Multitherman** (291125)
- H2020 Project **Exanode** (671578)
European Exascale Processor &
Memory Node Design
- H2020 Project **Oprecomp** (732631)
Open Transprecision Computing



Multitherman



Multitherman



Parallel Ultra Low Power (PULP)

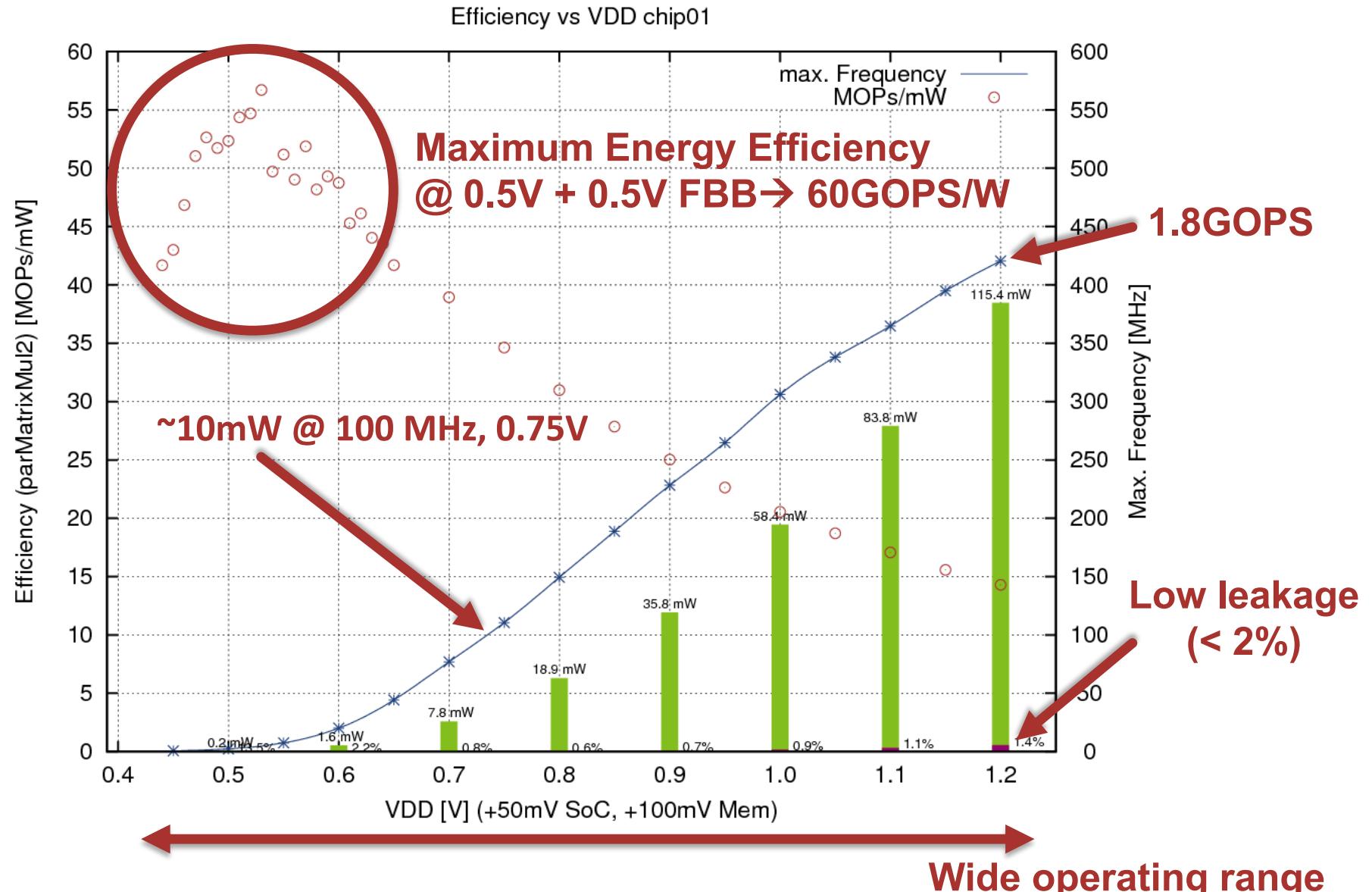
- Project started in **2013** by Luca Benini
- A collaboration between University of Bologna and ETH Zürich
- Key goal is

**How to get the most BANG
for the ENERGY consumed
in a computing system**

- We were able to start with a clean slate, no need to remain compatible to legacy systems.

Energy efficiency is the key driver in the PULP project

Measurement results from PULPv1 a four core cluster in ST FDSoI
28nm technology running a parallel matrix multiplications



To reach our goal we work on the following

- **We concentrate on programmable systems**
 - Can not have custom hardware, need to be flexible
 - We need to make the system accessible to application developers
- **Scalable over a wide operating range**
 - Work just as well when processing 0.001 GOPS as 1000 GOPS
- **Don't waste** idle energy
 - Eliminate sources where cores and systems are idly wasting energy
- **Make good use** of options provided by the **technology**
 - Body biasing, power gating, library, memory selection etc..
- Take advantage of **heterogeneous acceleration**
 - Allow an architecture where accelerators can be added efficiently

Structure of this workshop

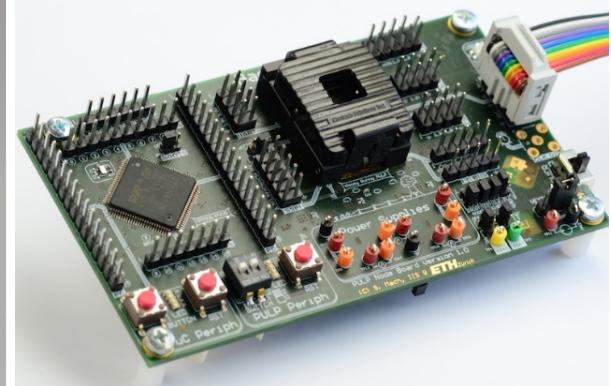
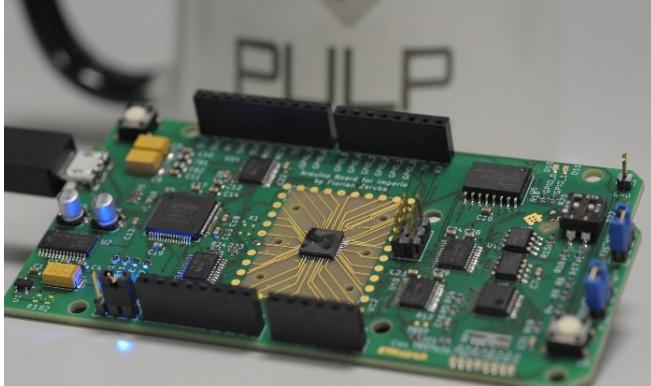
- Open source hardware and our role (Frank)
- The PULP family tree (Frank)
- Our RISC-V cores: Ariane, RI5CY and friends (Florian)
- Break – Demos
- Accelerators in PULP (Francesco)
- Our Programmable Multi-Core Accelerator – HERO (Andreas)
- Programming PULP (Andreas)

Please interrupt at any time to ask questions

PULP is maintained by a large group

- Luca Benini holds a dual appointment in ETH Zürich and Bologna
- Total team about 50-60 members (60% in Zürich, 40% in Bologna)
 - 1 Professor
 - 3 Assistant Professors and 2 Senior Scientists
 - 8 Post Doctoral researchers
 - 30+ Ph.D. students
 - 6 Technical staff and 2 embedded staff in industrial partners
- Most of this team works on projects that are related to PULP
 - Core group of 15 designers that concentrate on PULP development
 - Many others contribute to application development, software support
- In ETH Zürich, the Microelectronics Design Center supports the project
 - Permanent staff of 4
 - Maintains design flows for ASIC and FPGA

We have designed over 20 PULP based ASICs already



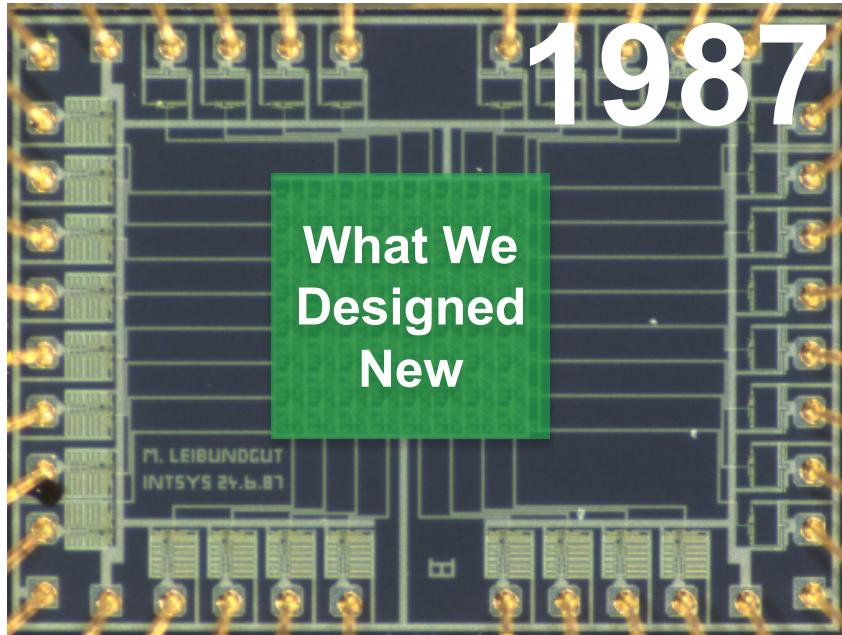
www.pulp-platform.org



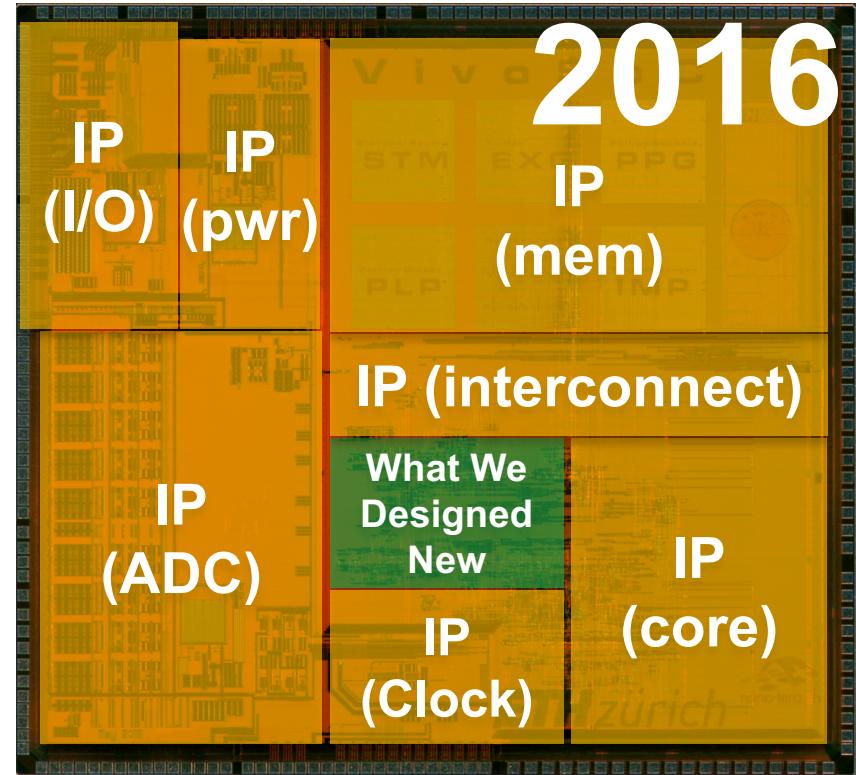
Why is Open Hardware different than Open Software

- From gnu.org www site:
<http://www.gnu.org/philosophy/free-hardware-designs.html>
- **Software** is the operational part of a device that can be copied and changed in a computer
- **Hardware** is the operational part that can't be.
- You can not produce HW directly, you need
 - manufacturing plants
 - know-how
 - and volumeto be able to manufacture HW **with reasonable cost**.

The way we do IC design has changed



Parallel Multiplier 3, Fast CMOS multiplier, Faselec 3µm, 2.5mm x 2.0mm
http://asic.ethz.ch/1986/Parallel_Multiplier3.html



VivoSoC2, Biomedical signal Acquisition SoC, SMIC130, 4.7mm x 4.7mm
<http://asic.ethz.ch/2016/Vivosoc2.html>

- There is a need for silicon proven high quality IPs
- This will allow more groups to be able to design SoCs

Open Hardware is a necessity, not an ideological crusade

- **The way we design ICs has changed, big part is now infrastructure**
 - Processors, peripherals, memory subsystems are now considered infrastructure
 - Very few (if any) groups design complete IC from scratch
 - High quality building blocks (IP) needed

Open Hardware is a necessity, not an ideological crusade

- The way we design ICs has changed, big part is now infrastructure
 - Processors, peripherals, memory subsystems are now considered infrastructure
 - Very few (if any) groups design complete IC from scratch
 - High quality building blocks (IP) needed
- We need an easy and fast way to collaborate with people
 - Currently complicated agreements have to be made between all partners
 - In many cases, too difficult for academia and SMEs

Current HW only supports security through obscurity

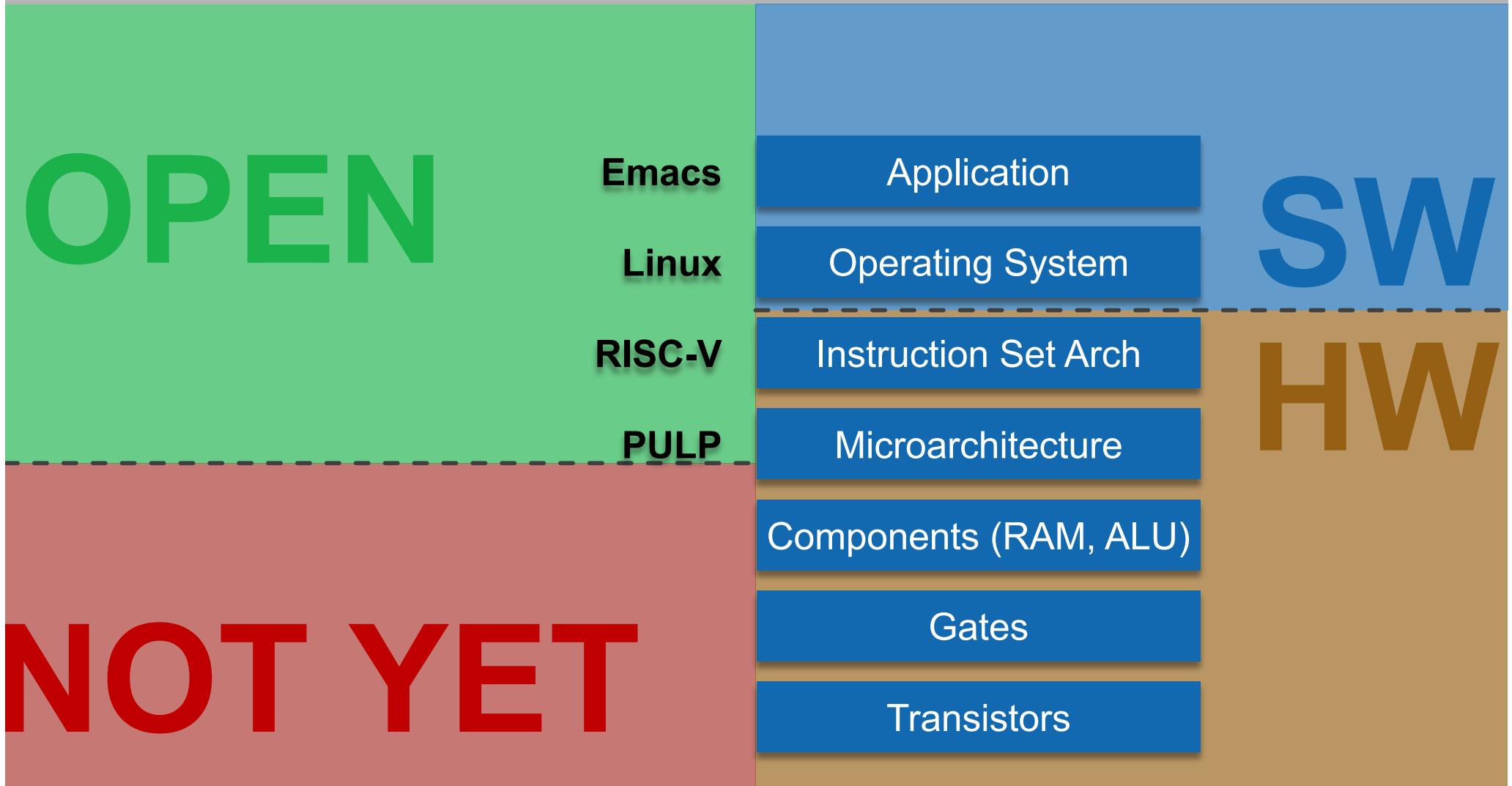
- Systems are built on hardware blocks where you do not know what exactly is inside
 - Open standards have proven themselves in SW
Why should HW be any different?
 - If you really want, you can still ‘obscure’ HW,
but open HW gives you a choice!
 - Many bugs, features with unintentional
consequences are hiding inside HW
- Open HW will allow a larger community to verify building blocks
 - Better verification, more reliable hardware
- This slide was originally presented before Spectre and Meltdown



Open Hardware is a necessity, not an ideological crusade

- The way we design ICs has changed, big part is now infrastructure
 - Processors, peripherals, memory subsystems are now considered infrastructure
 - Very few (if any) groups design complete IC from scratch
 - High quality building blocks (IP) needed
- We need an easy and fast way to collaborate with people
 - Currently complicated agreements have to be made between all partners
 - In many cases, too difficult for academia and SMEs
- **Hardware is a critical for security, we need to ensure it is secure**
 - Being able to see what is really inside will improve security
 - Having a way to design open HW, will not prevent people from keeping secrets.

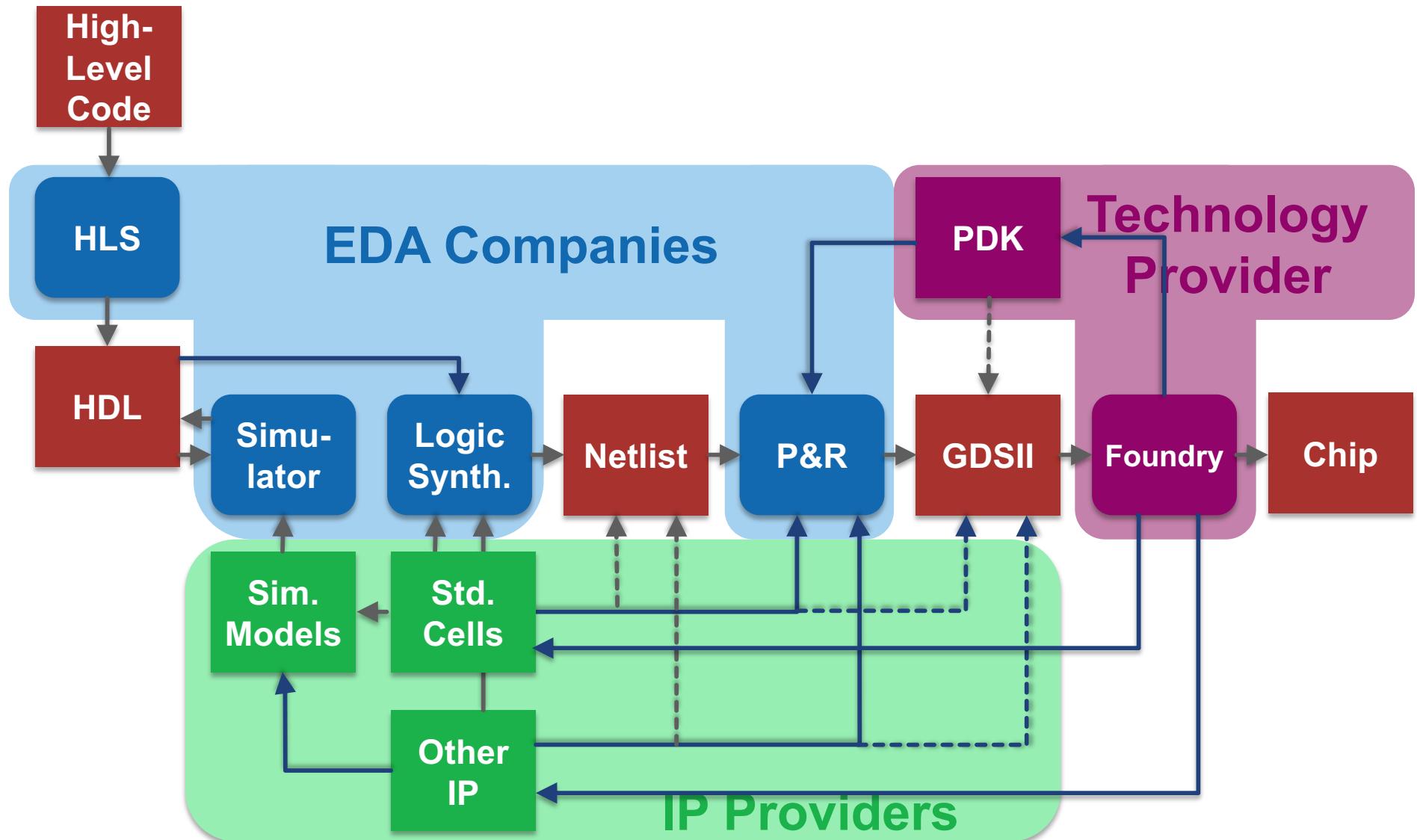
Where are we now?



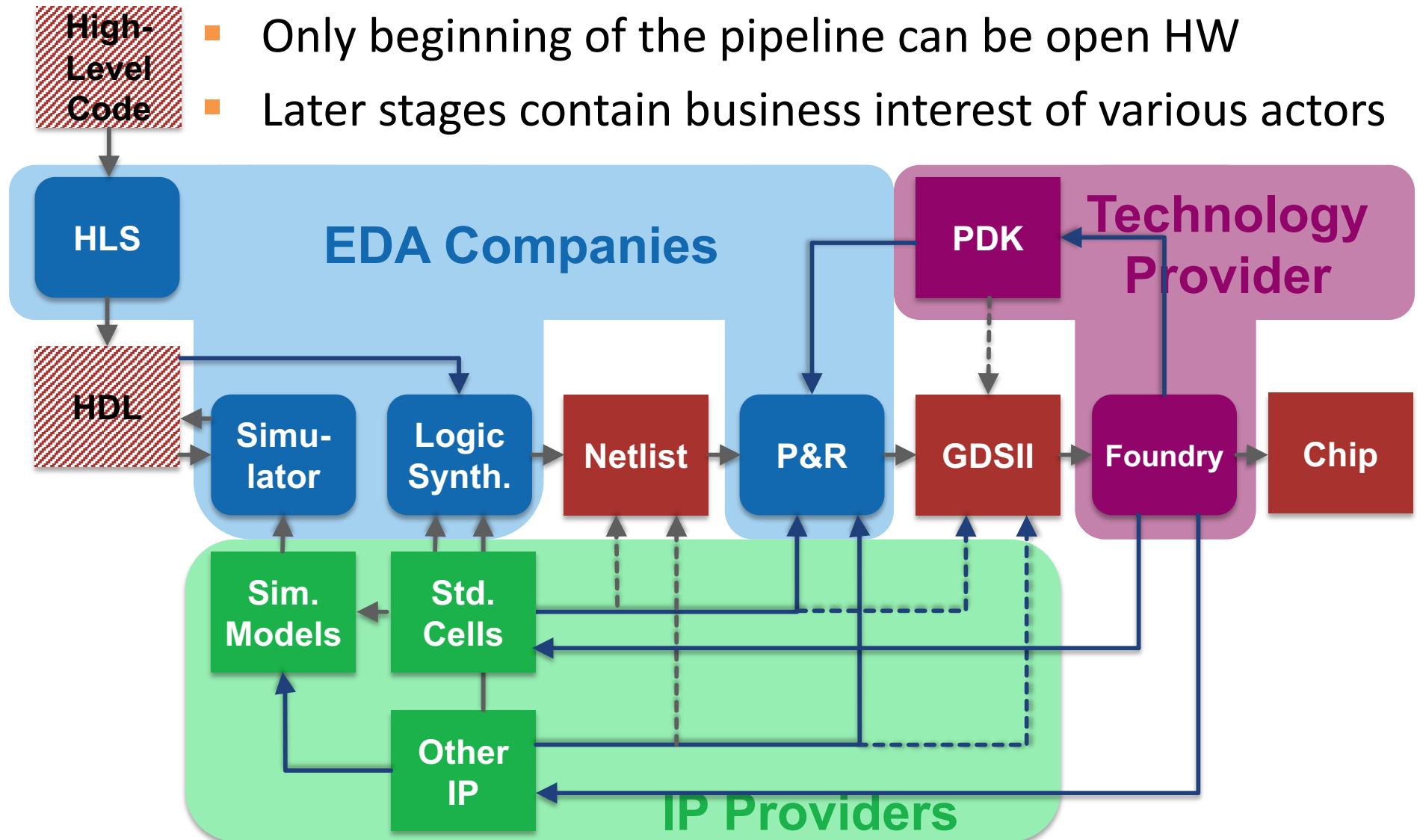
Hardware design flows for PCB/FPGA/ASIC are different

- **The differences complicate things**
 - Not a uniform way to discuss the issue, depends on the design flow
 - This talk about ASIC flow (the most complex one).
- **Different actors, different revenue streams**
 - Not every actor in the current design flow, earns money the same way
 - EDA companies are long complaining that they are out of the loop
Their income is not based on the amount of ICs produced.
 - **Not everybody will be happy** if open hardware will be more common
 - Important to understand the relationships
- **Interestingly most intermediate file formats are open, readable**
 - Verilog, Liberty, SPICE, EDIF, CIF, LEF, DEF, OA (open access)

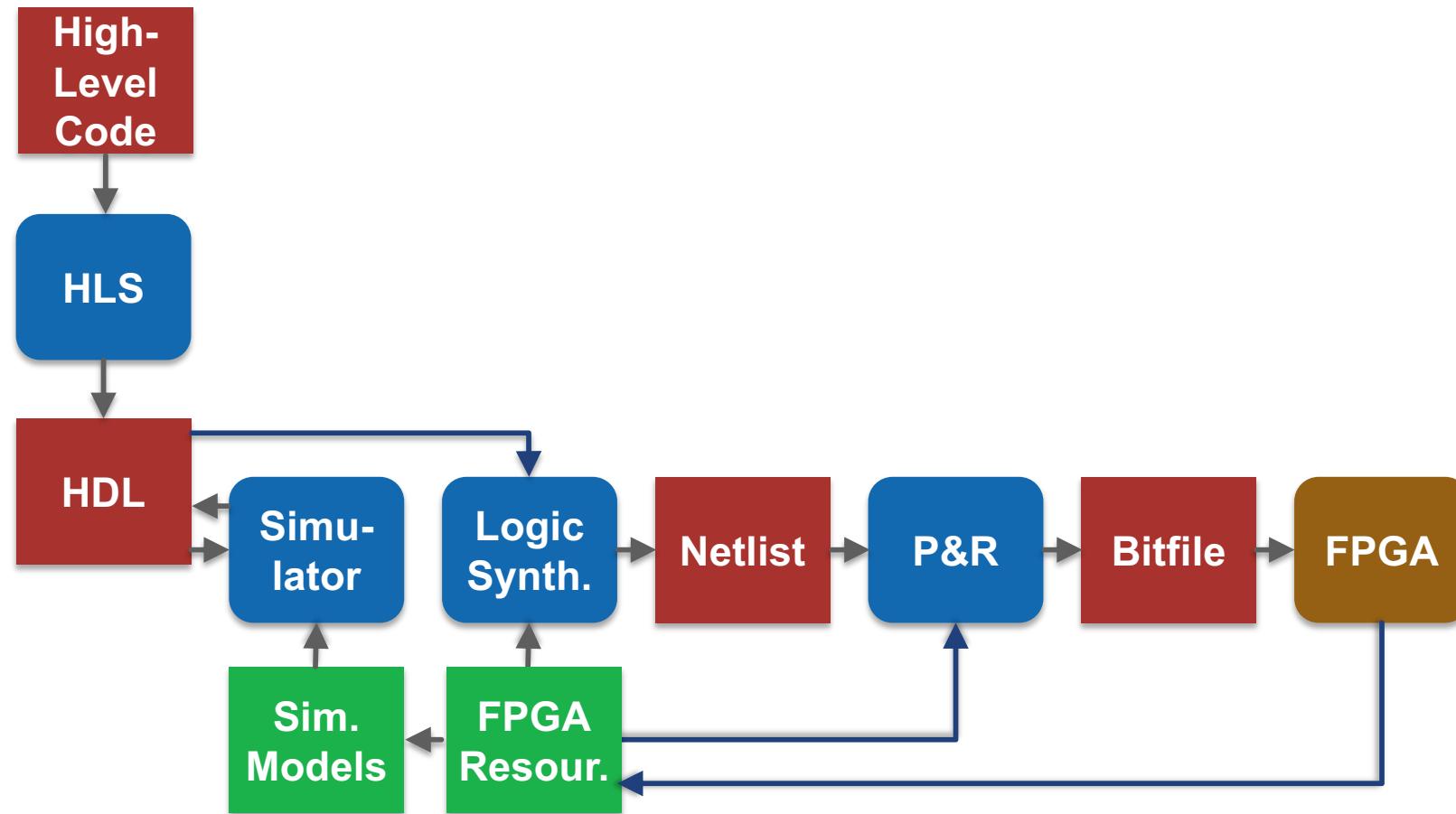
ASIC Design Flow and Main Actors



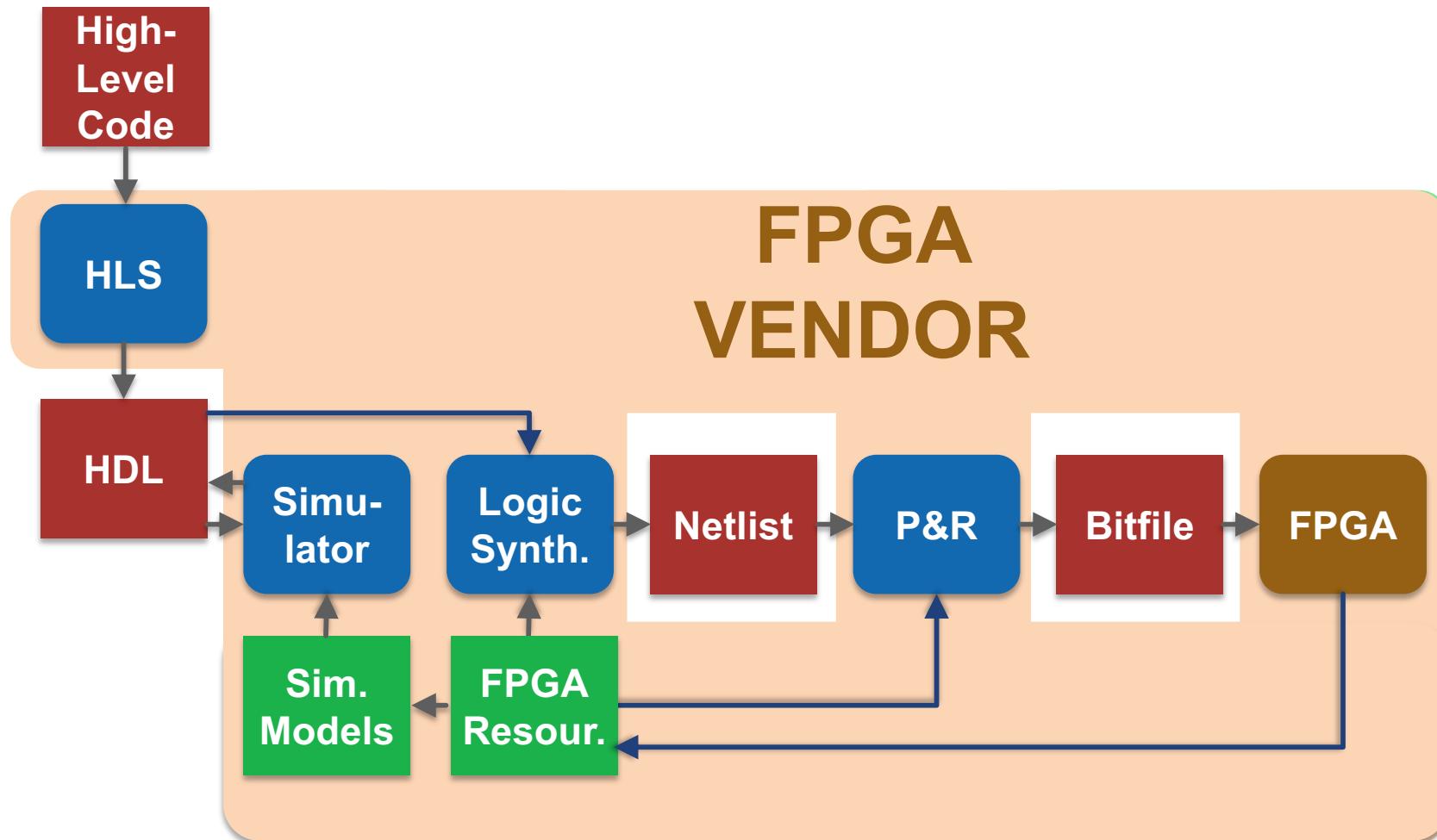
ASIC Design Flow and Main Actors



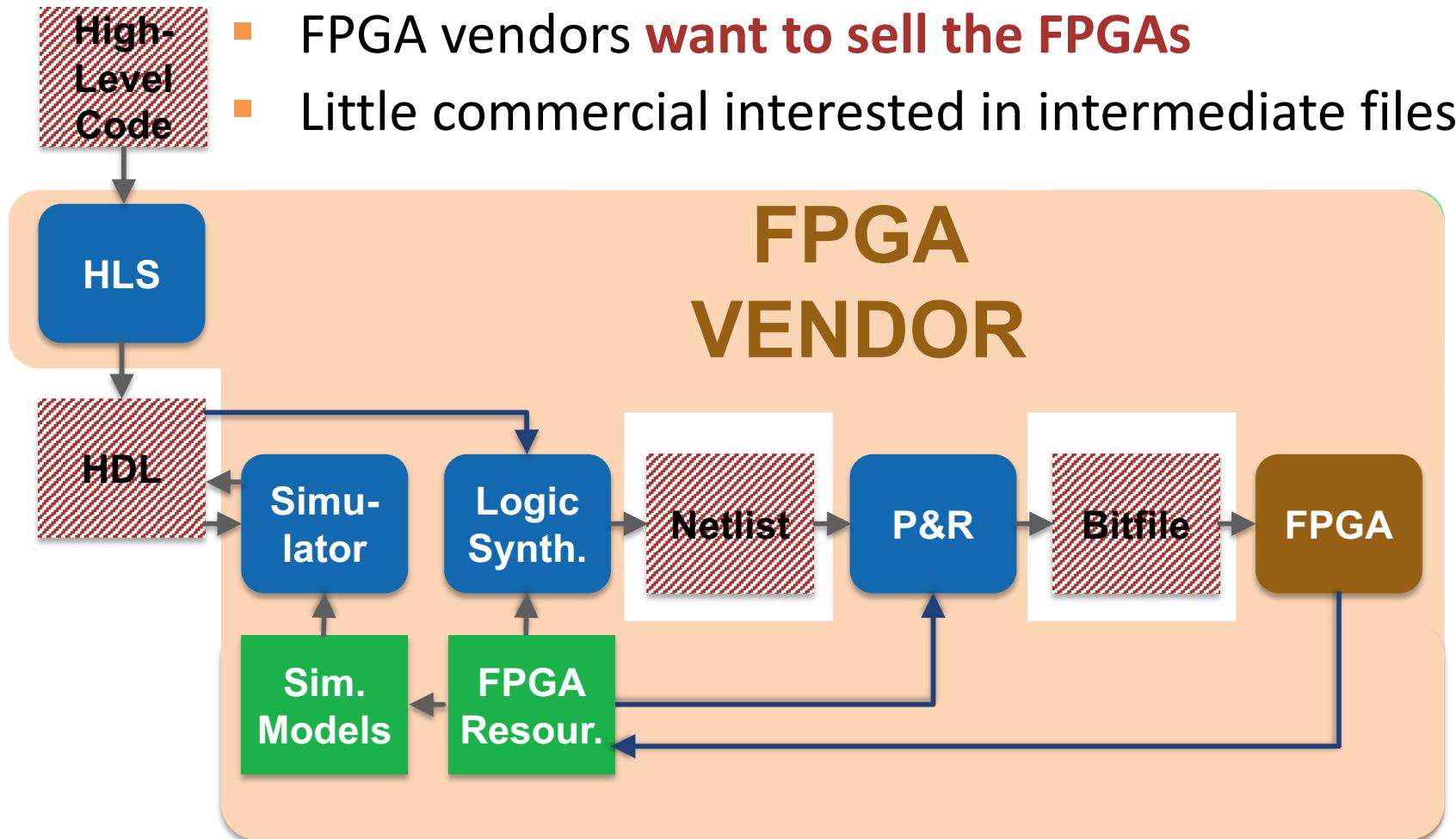
FPGA Design Flow and Main Actors



FPGA Design Flow and Main Actors



FPGA Design Flow and Main Actors



- Most vendors allow bitfiles to be published -> will sell more FPGAs

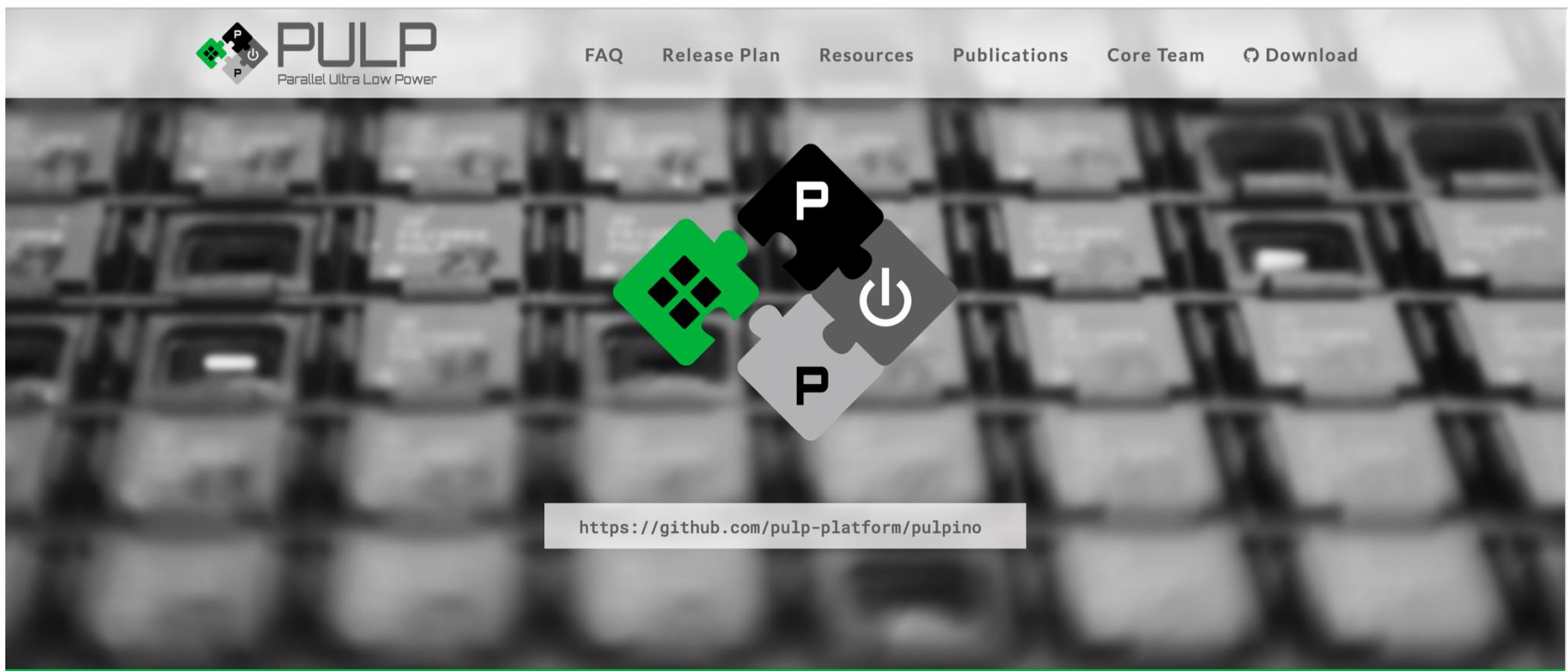
At the moment, open HW can (mostly/only) be HDL code

- **The following are ok:**
 - RTL code written in HDL, or a high-level language for HLS flow
 - Testbenches in HDL and associated makefiles, golden models
- **How about support scripts for different tools?**
 - Synthesis scripts, tool startup files, configurations
- **And these are currently no go :**
 - Netlists mapped to standard cell libraries
 - Placement information (DEF)
 - Actual Physical Layout (GDSII)

Pros and Cons of Open Source Hardware

- **Pros as we see it**
- **Develops larger community**
 - Many volunteers
 - Smaller projects
- **Access to ‘modern’ systems**
 - Will benefit SMEs
 - Direct academic contributions
- **You know what is inside**
 - Open systems vital for security
- **Benchmarking**
 - Replicate results easily
- **Cons as we are told**
- **Licensing issues**
 - Is GPL/LGPL possible
 - Copyleft/Copyright discussions
- **Technology specific parts**
 - Analog blocks
 - Several IPs (IOs, Memories)
- **Quality/Verification**
- **Tool support**
 - Big gap to commercial SW

We firmly believe in Open Source movement



- First launched in February 2016 (github)
 - HDL code, testbenches, testcases, SW, scripts, debug support.
 - We use solderpad (v0.51) license (Apache like license adapted for HW)

SOLDERPad
<http://www.solderpad.org/licenses/>



Multitherman



PULP Open-Source Releases and External Contributions

Releases

1

February 2016

First release of **PULPino**, our single-core microcontroller

2

May 2016

Toolchain and compiler for our RISC-V implementation (**RI5CY**), DSP extensions

3

August 2017

PULPino updates, new cores Zero-riscy and Micro-riscy, **FPU**, toolchain updates

4

February 2018

PULPissimo, ARIANE, PULP

5

A bit later in 2018

PULP, HERO



Multitherman



Community Contributions



1

June 2017

Porting of **Verilator** and **BEEBS**

benchmarks to PULPino

<https://github.com/embecosm/ri5cy>



2

September 2017

Porting of **ARM CMSIS** to PULPino

<https://github.com/misaleh/CMSIS-DSP-PULPino>



3

November 2017

Numerous **Bug fixes** to RI5CY in PULPino

<https://github.com/pulp-platform/riscv>



4

December 2017

STING: Open-Source Verification

Environment for PULPino

<http://valtrix.in/programming/running-sting-on-pulpino>

We try to leverage open source as much as possible

Programming Model



Virtualization Layer



Compiler Infrastructure



Processor & Hardware IPs



Low-Power Silicon Technology



Multitherman



Silicon and Open Hardware fuel PULP success

- Many companies (we know of) are actively using PULP
 - They value that it is **silicon proven**
 - They like that it uses a **permissive open source license**

Companies with announced products, business Companies that use PULP internally or for training Companies exploring opportunities	Companies that are using/evaluating PULP	Research Centers/Universities using PULP
	<ul style="list-style-type: none">■ GreenWaves Technologies■ Dolphin■ IQ Analog (14nm chips)■ Embecosm■ lowRISC■ Mentor Graphics■ Cadence Design Systems■ ST Microelectronics (IT,F)■ Micron■ SIAE Microelectronica■ Advanced Circuit Pursuit■ NXP■ Shanghai Xidian Technology■ SCS Zurich■ IMT technologies■ Google■ Microsemi■ Arduino■ RacyICs	<ul style="list-style-type: none">■ Stanford■ Cambridge■ UCLA■ CEA/LETI■ EPFL■ National Chia Tung University■ Politecnico di Milano■ Politecnico di Torino■ Universita Roma I■ Instituto Superior Tecnico – U. de Lisboa■ Fondazione Bruno Kessler■ Zagreb HER■ Universita di Genova■ Istanbul Technical U.■ RWTH Aachen■ Lund■ USI – Lugano■ Bar-Ilan■ TU-Kaiserslautern■ TU-Graz■ UC San Diego■ CSEM■ IBM Research

The PULP family explained

RISC-V Cores

Peripherals

Interconnect

Platforms

Accelerators

We have developed several optimized RISC-V cores

RISC-V Cores

RI5CY

32b

**Micro
riscy**

32b

**Zero
riscy**

32b

Ariane

64b

We have also been working on hardware accelerators

RISC-V Cores

RI5CY

32b

**Micro
riscy**

32b

**Zero
riscy**

32b

Ariane

64b

Accelerators

HWCE
(convolution)

Neurostream
(ML)

HWCrypt
(crypto)

PULPO
(1st order opt)

We have our own peripherals and interconnect solutions

RISC-V Cores

RI5CY

32b

**Micro
riscy**

32b

**Zero
riscy**

32b

Ariane

64b

Peripherals

JTAG

SPI

UART

I2S

DMA

GPIO

Interconnect

Logarithmic interconnect

APB – Peripheral Bus

AXI4 – Interconnect

Accelerators

**HWCE
(convolution)**

**Neurostream
(ML)**

**HWCrypt
(crypto)**

**PULPO
(1st order opt)**

By combining these components we get PULP platforms

RISC-V Cores

RI5CY

32b

Micro riscy

32b

Zero riscy

32b

Ariane

64b

Peripherals

JTAG

SPI

UART

I2S

DMA

GPIO

Interconnect

Logarithmic interconnect

APB – Peripheral Bus

AXI4 – Interconnect

Platforms



Single Core

- PULPino
- PULPissimo

Accelerators

HWCE
(convolution)

Neurostream
(ML)

HWCrypt
(crypto)

PULPO
(1st order opt)

Our main research is on Near-Threshold Multi-Core Systems

RISC-V Cores

RI5CY

32b

Micro riscy

32b

Zero riscy

32b

Ariane

64b

Peripherals

JTAG

UART

DMA

SPI

I2S

GPIO

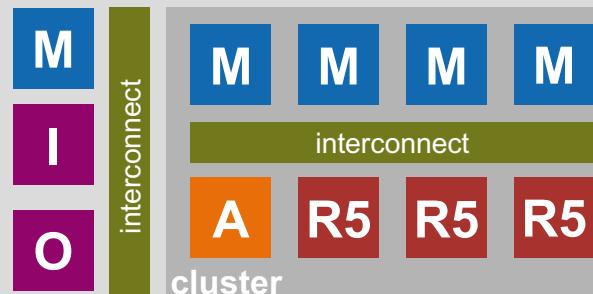
Interconnect

Logarithmic interconnect

APB – Peripheral Bus

AXI4 – Interconnect

Platforms



Single Core

- PULPino
- PULPissimo

Multi-core

- Fulmine
- Mr. Wolf

Accelerators

HWCE
(convolution)

Neurostream
(ML)

HWCrypt
(crypto)

PULPO
(1st order opt)

Finally for HPC applications we have multi-cluster systems

RISC-V Cores

RI5CY

32b

Micro riscy

32b

Zero riscy

32b

Ariane

64b

Peripherals

JTAG

UART

DMA

SPI

I2S

GPIO

Interconnect

Logarithmic interconnect

APB – Peripheral Bus

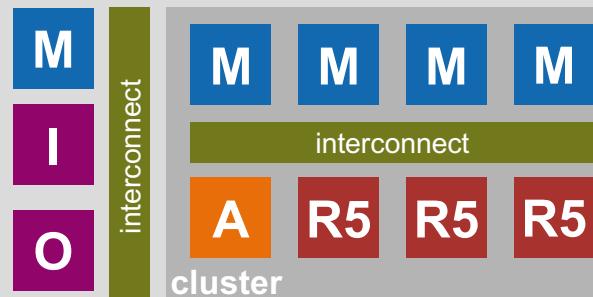
AXI4 – Interconnect

Platforms



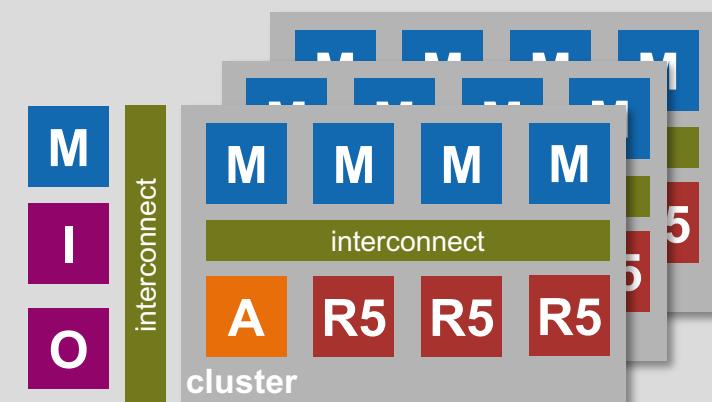
Single Core

- PULPino
- PULPissimo



Multi-core

- Fulmine
- Mr. Wolf



Multi-cluster

- Hero

IOT

HPC

Accelerators

HWCE
(convolution)

Neurostream
(ML)

HWCrypt
(crypto)

PULPO
(1st order opt)

Eventually we plan to release ALL we did on PULP

RISC-V Cores



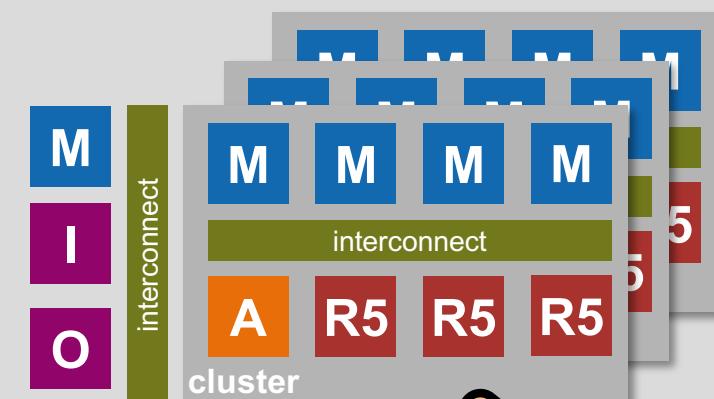
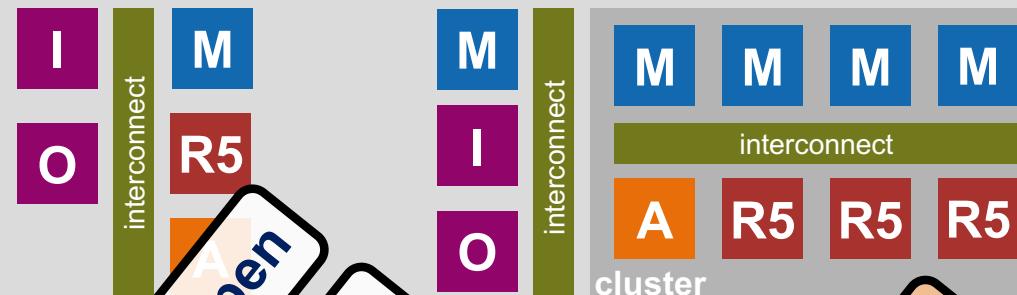
Peripherals



Interconnect



Platforms



Multi-cluster

- Hero

Accelerators

HWCE
(convolution)

Neurostream
(ML)

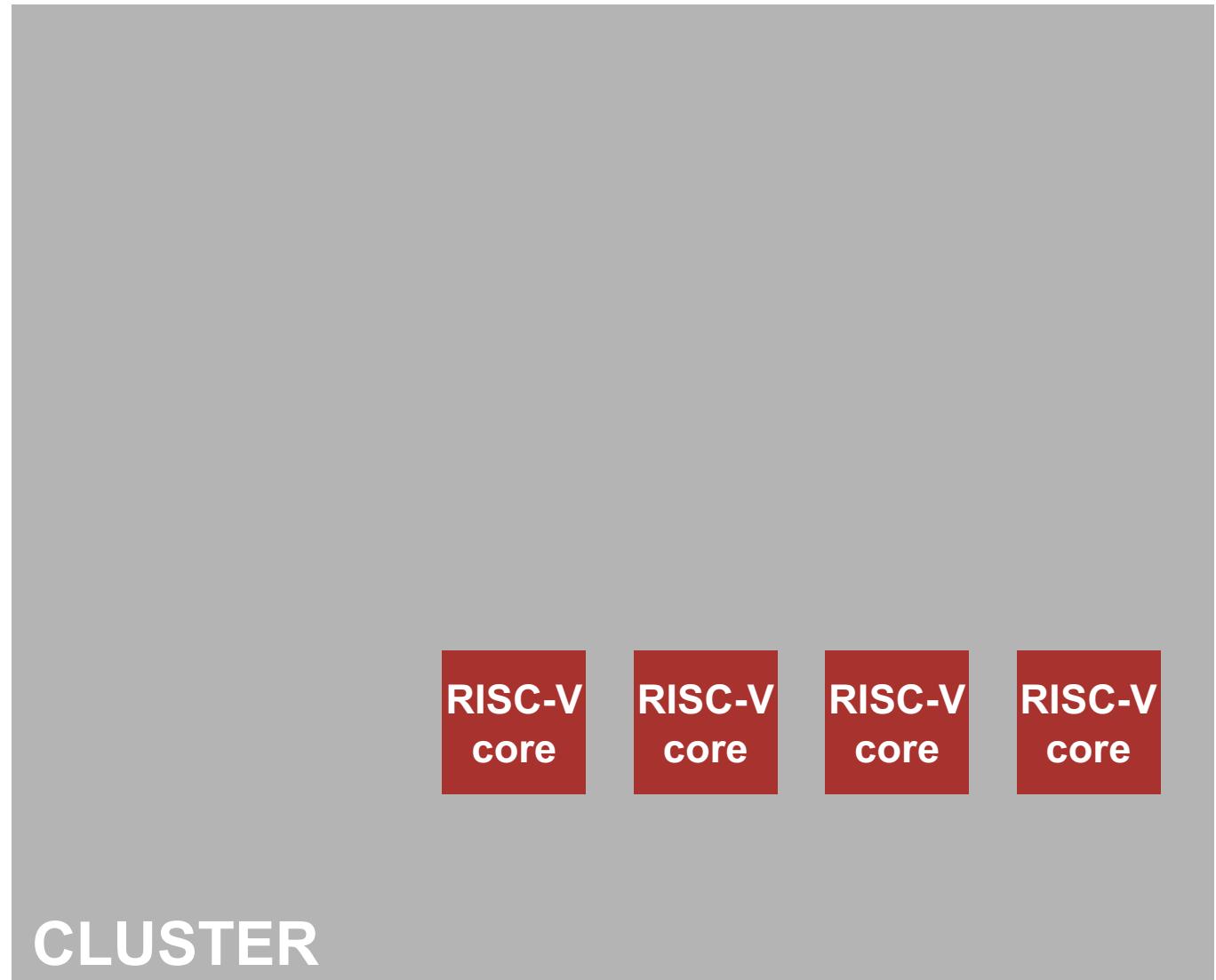
HWCrypt
(crypto)

PULPO
(1st order opt)

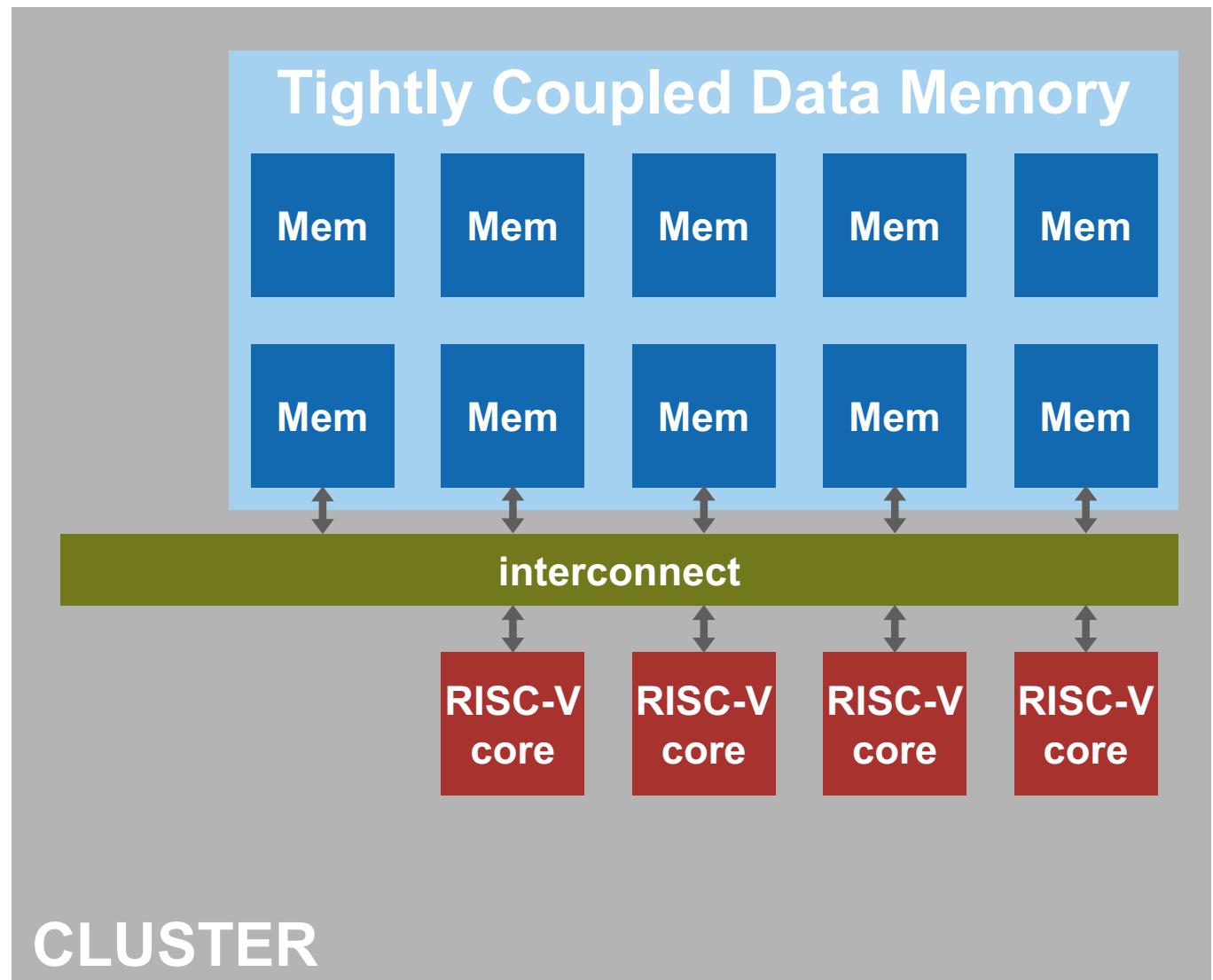
The main components of a PULP cluster

- **Multiple RISC-V cores**
 - Individual cores can be started/stopped with little overhead
 - DSP extensions in cores
- **Multi-banked scratchpad memory (TCDM)**
 - **Not a cache**, there is no L1 data cache in our systems
- **Logarithmic Interconnect allowing all cores to access all banks**
 - Cores will be stalled during contention, includes arbitration
- **DMA engine to copy data to and from TCDM**
 - Data in TCDM managed by software
 - Multiple channels, allows pipelined operation
- **Hardware accelerators with direct access to TCDM**
 - No data copies necessary between cores and accelerators.

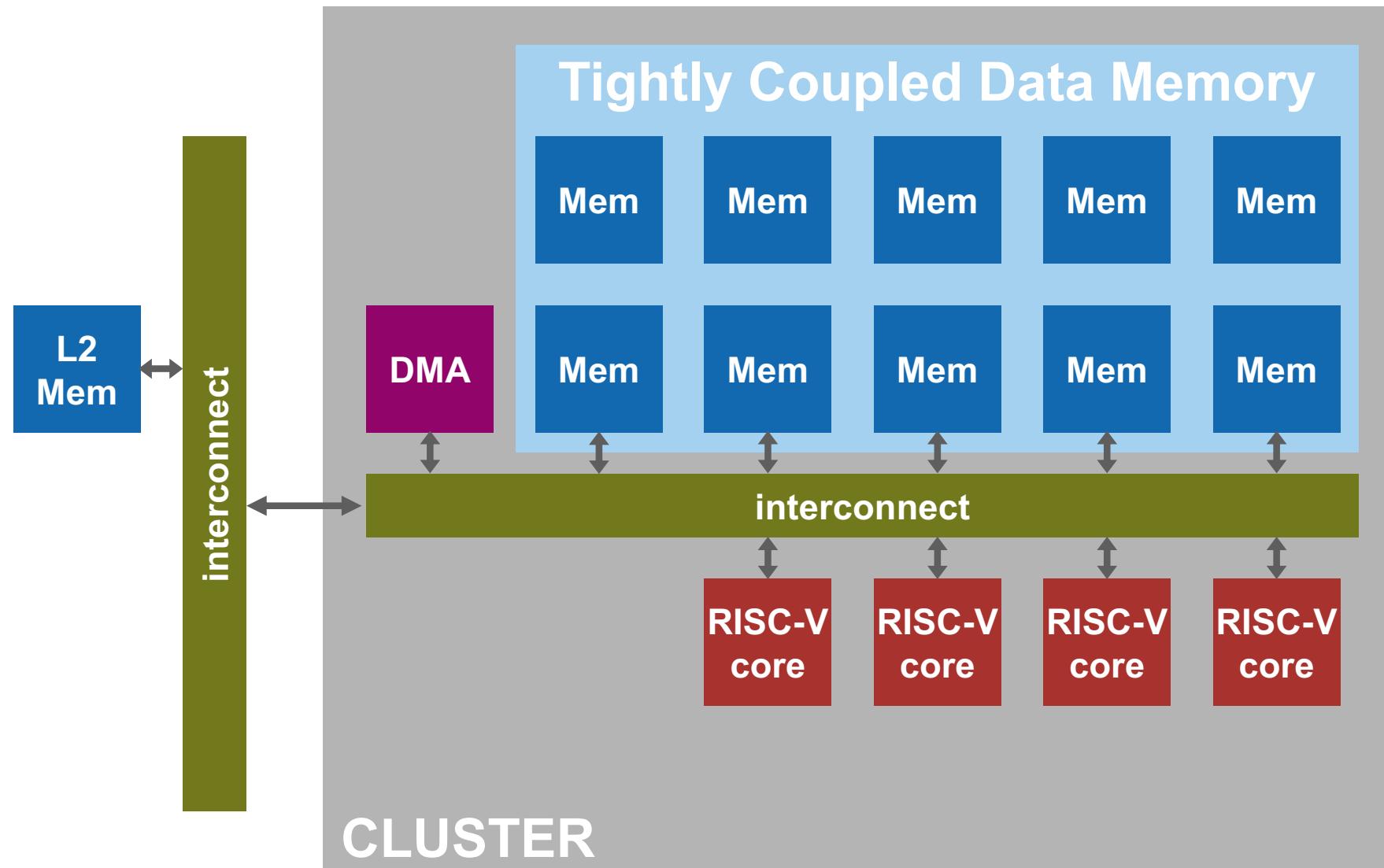
PULP cluster contains multiple RISC-V cores



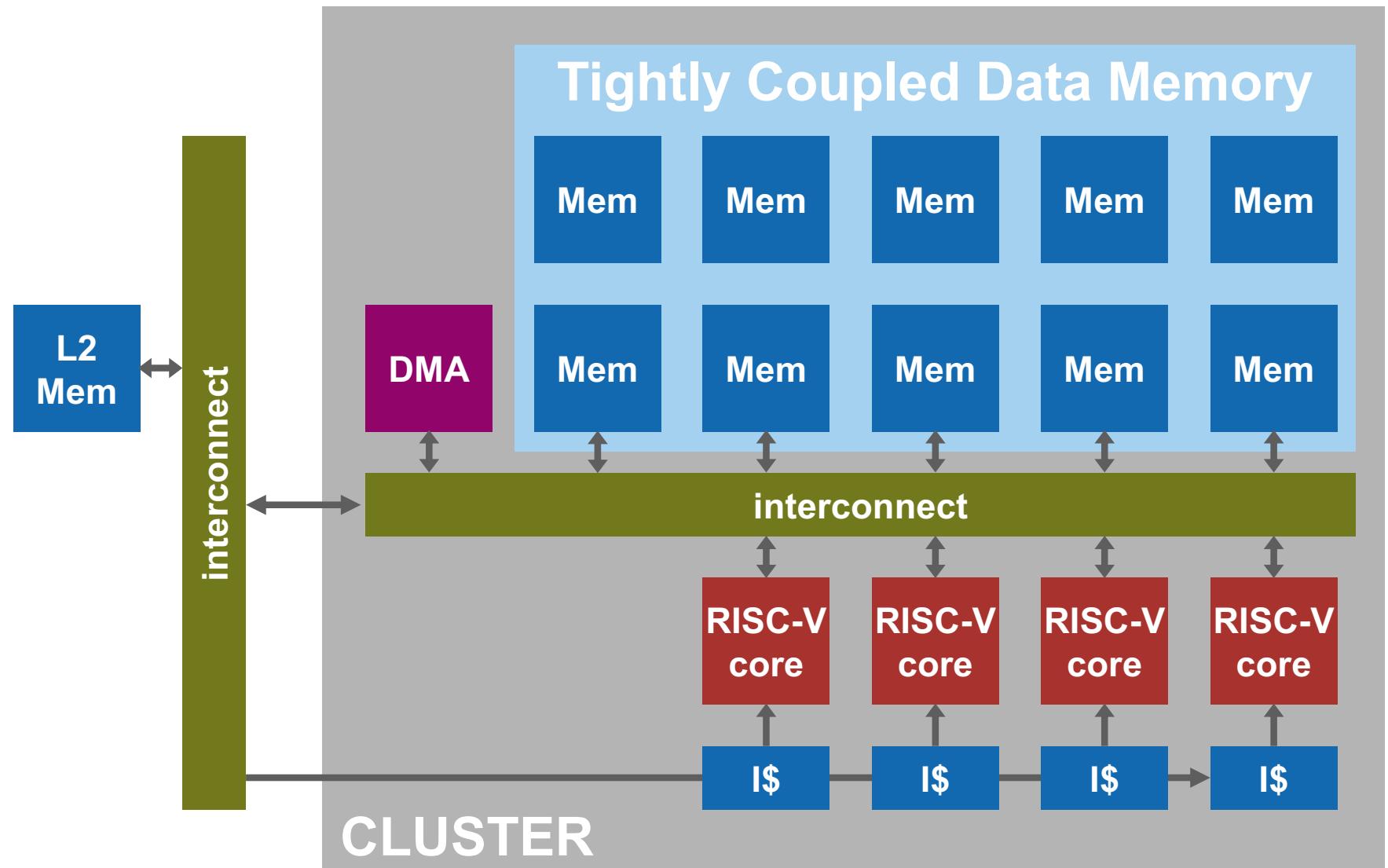
All cores can access all memory banks in the cluster



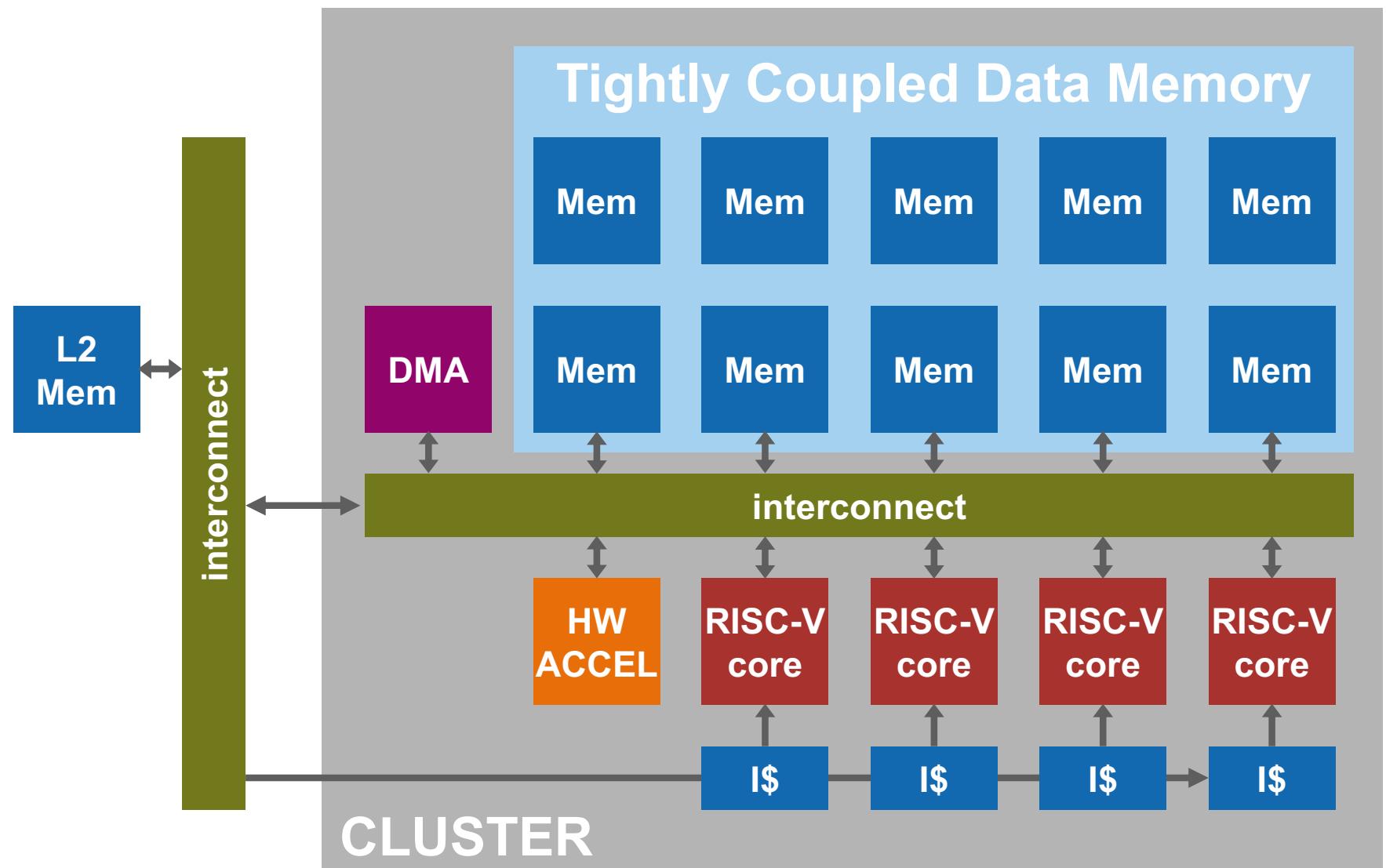
Data is copied from a higher level through DMA



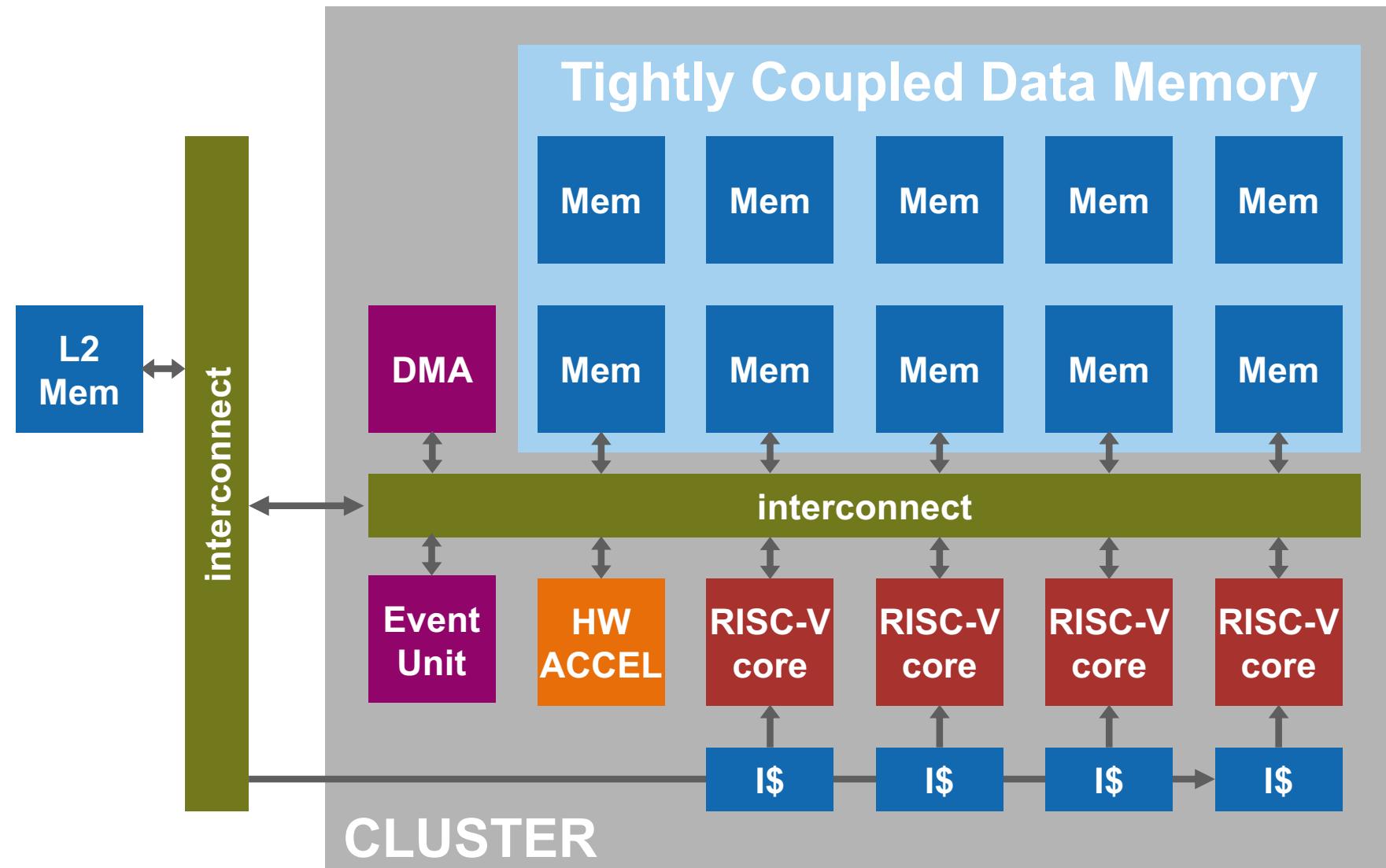
There is a (shared) instruction cache that fetches from L2



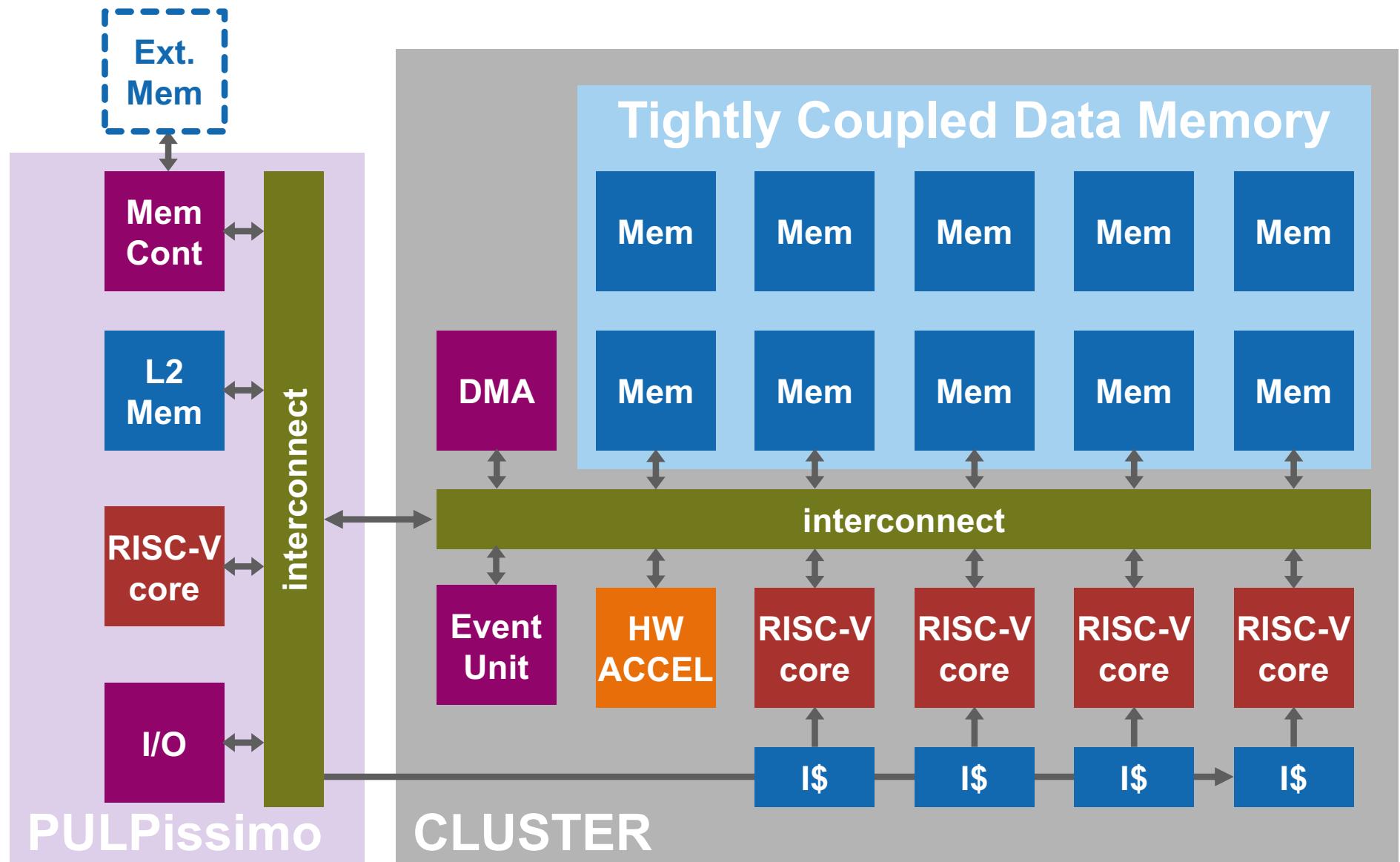
Hardware Accelerators can be added to the cluster



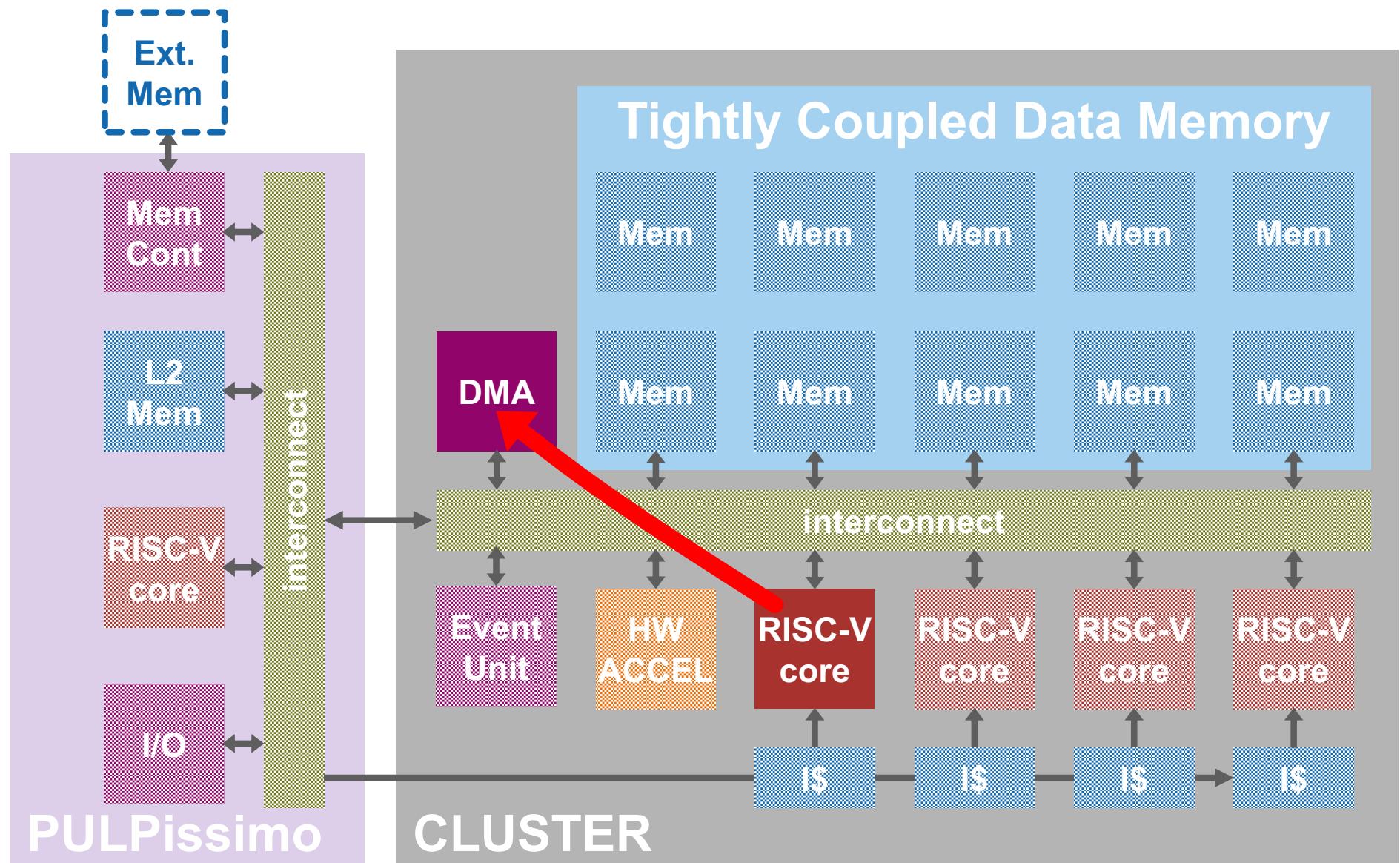
Event unit to manage resources (fast sleep/wakeup)



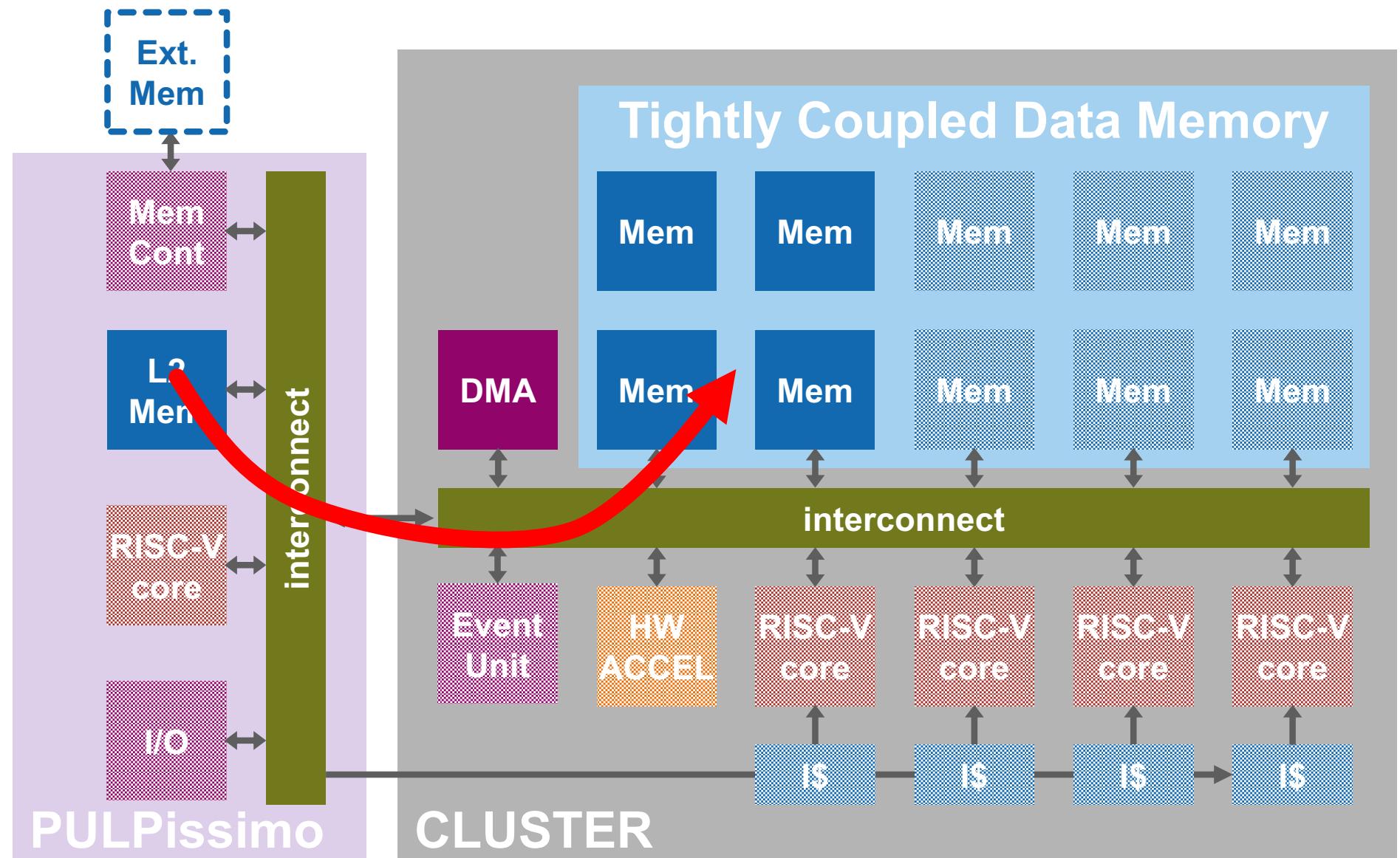
An additional microcontroller system (PULPissimo) for I/O



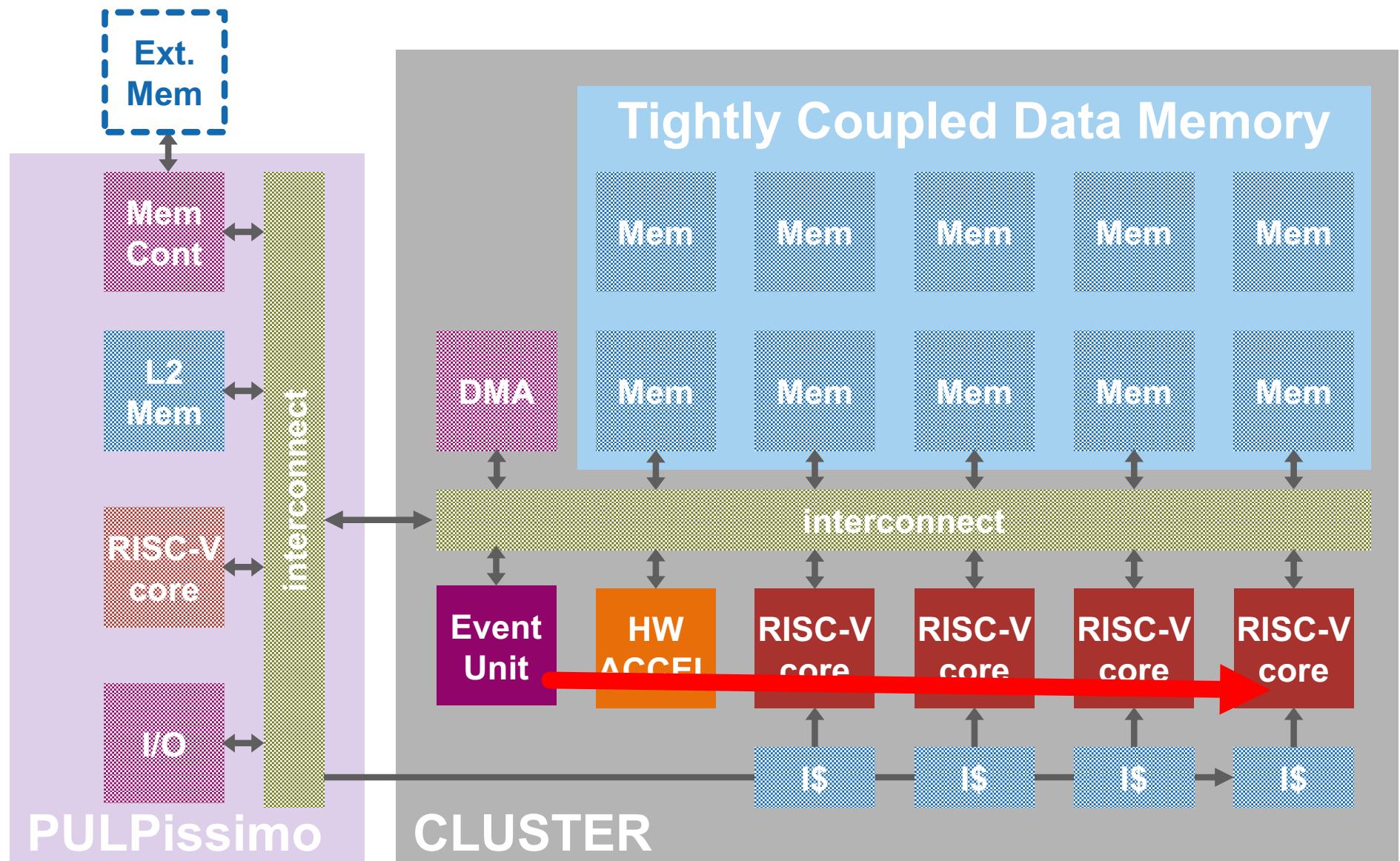
How do we work: Initiate a DMA transfer



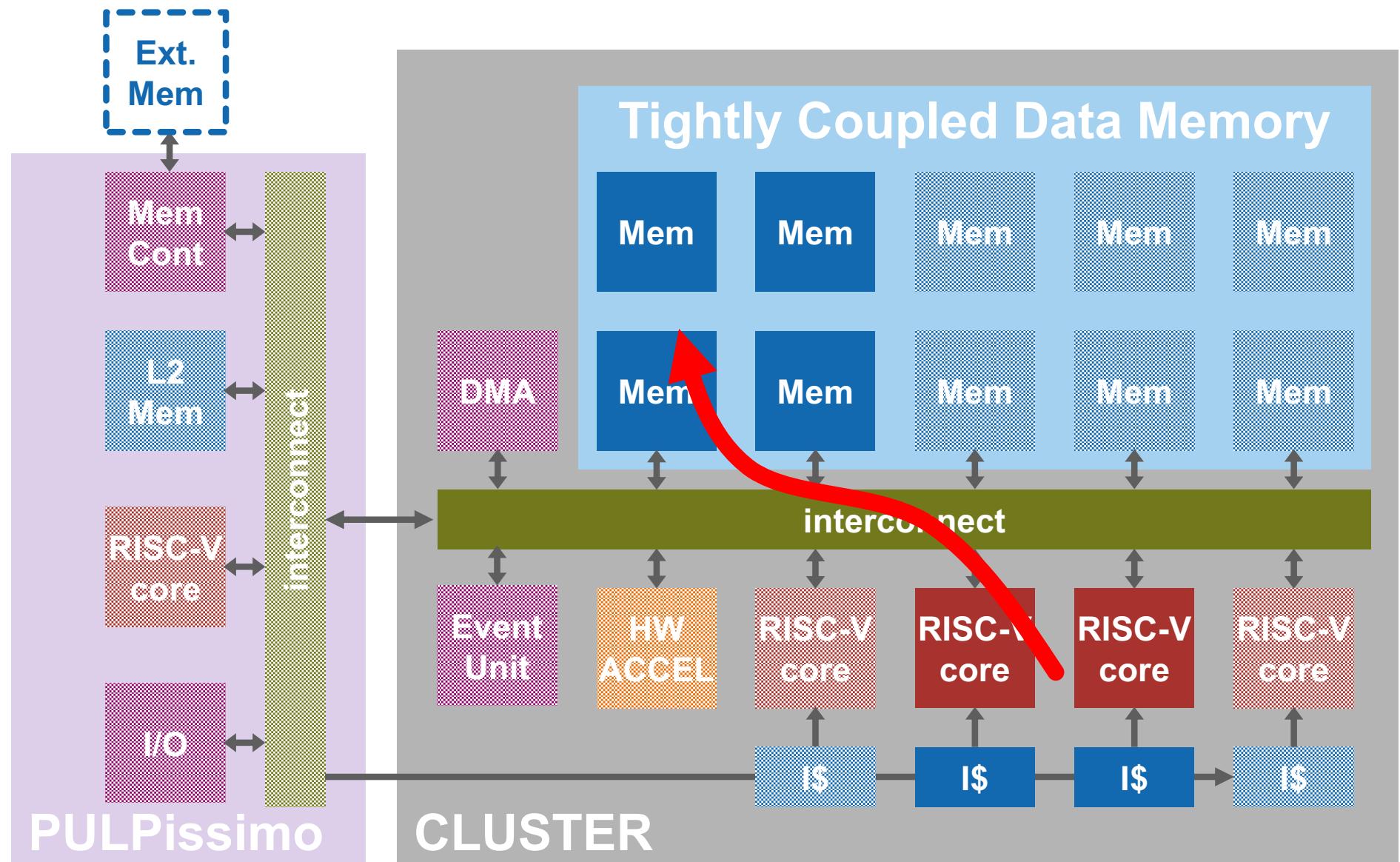
Data copied from L2 into TCDM



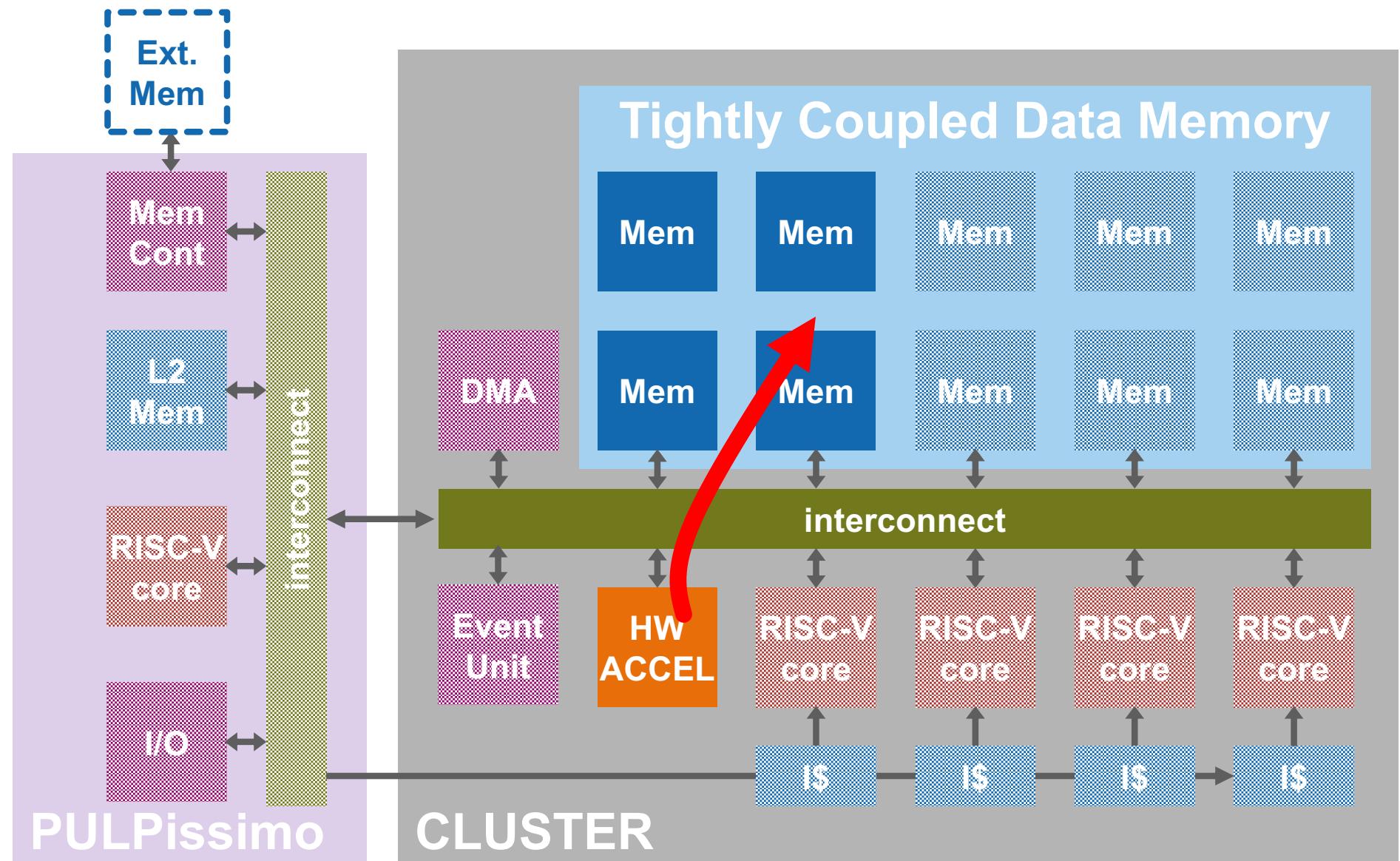
Once data is transferred, event unit notifies cores/accel



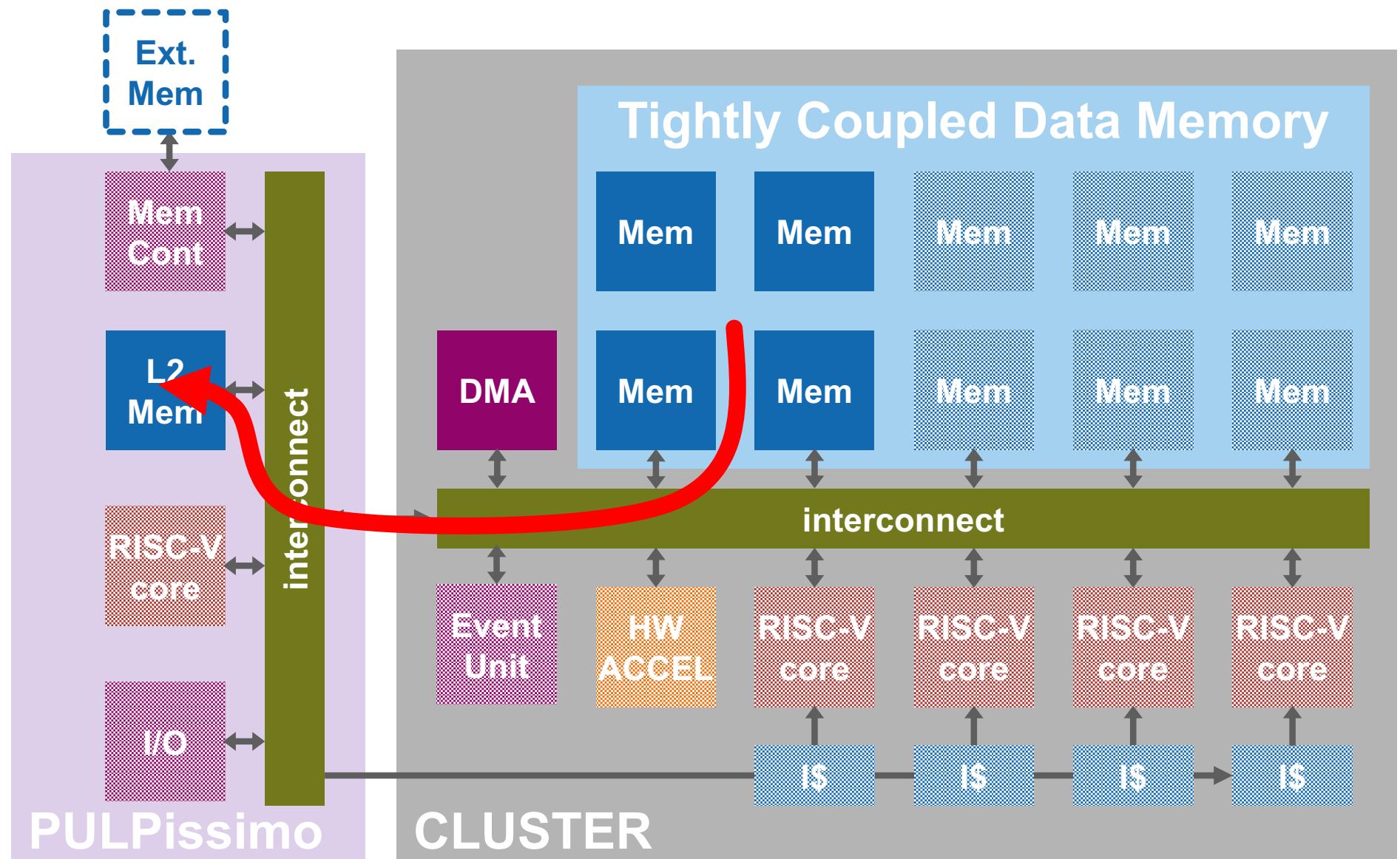
Cores can work on the data transferred



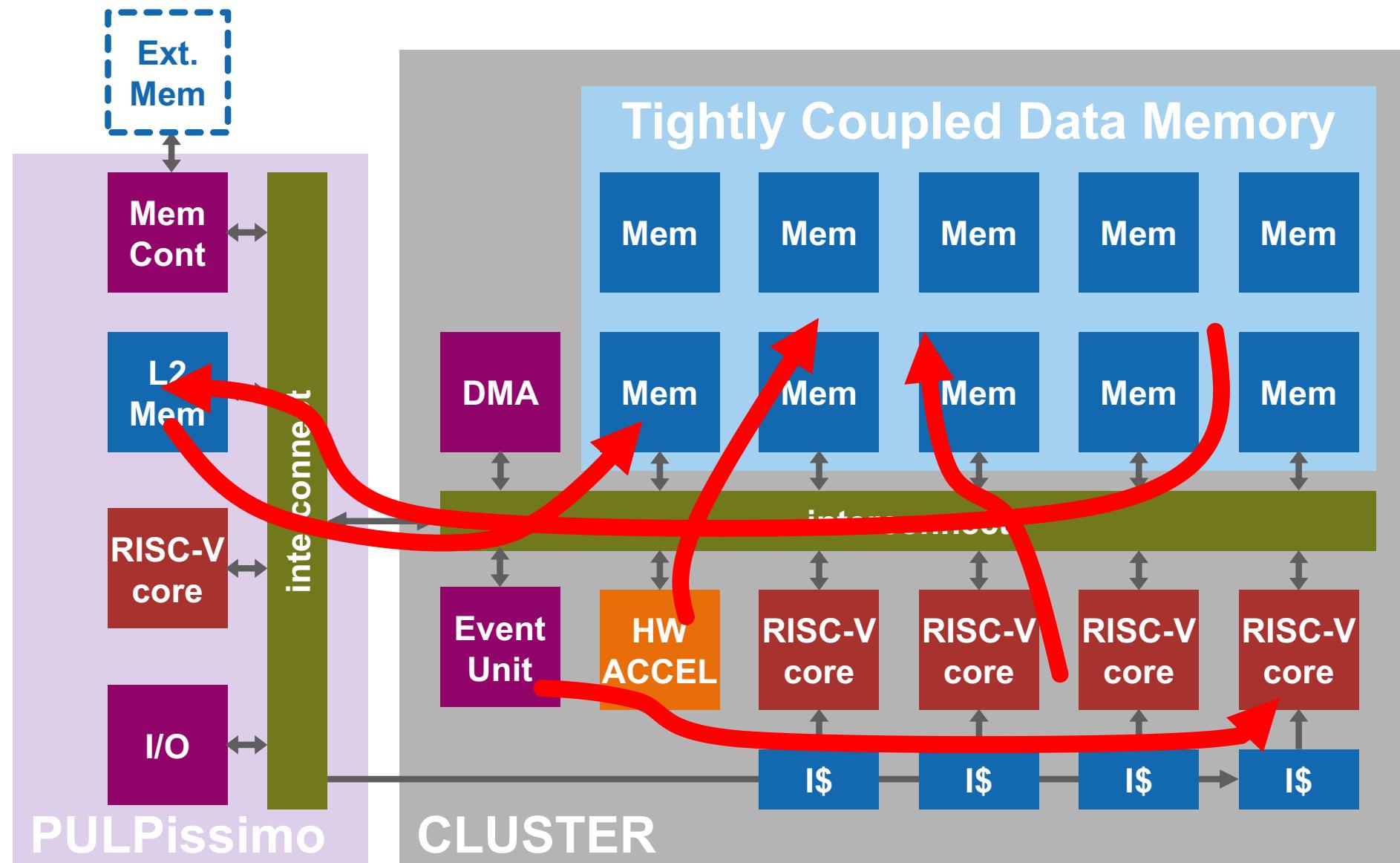
Or accelerators



Once our work is done, DMA copies data away

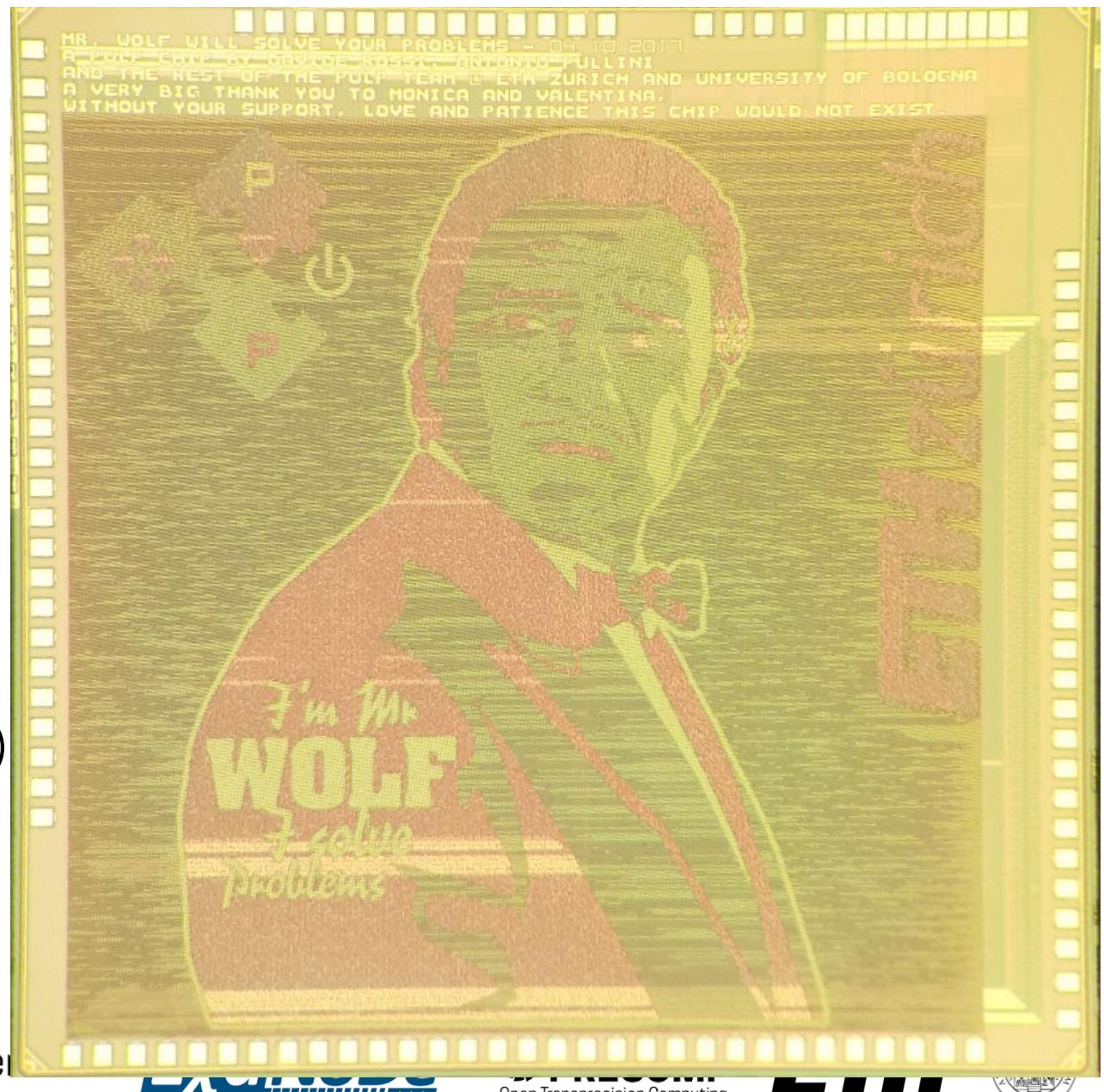


DMA data copies and processing actually work in parallel



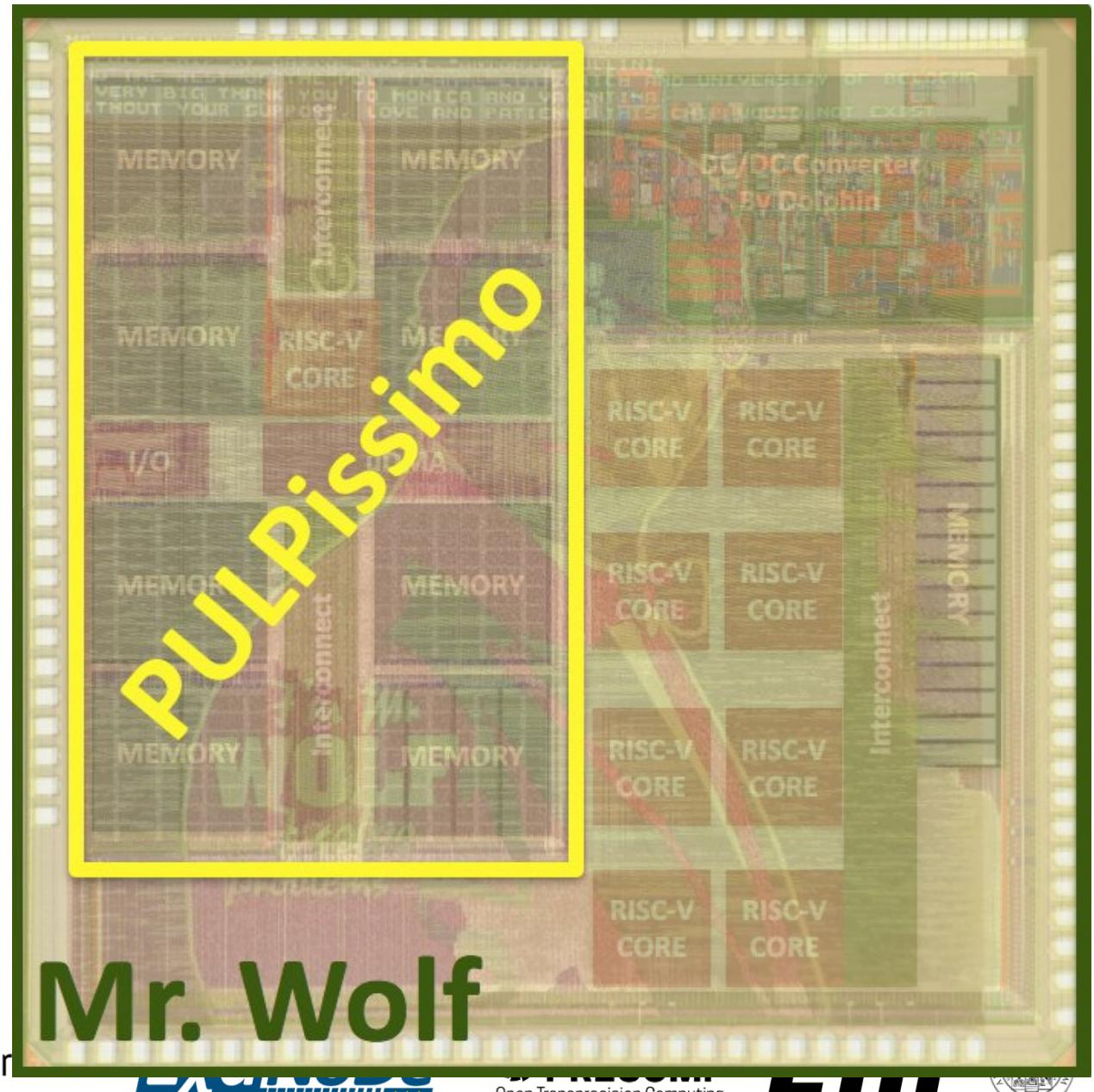
Mr. Wolf will (hopefully) solve problems

- TSMC 40LP - 9mm²
- 64 pin QFN package
- Cluster with
 - 8x RI5CY
 - 2x shared FPUs
 - 64 kByte TCDM
- SoC domain with
 - micro-riscy core to control power modes
 - DC-DC converter (Dolphin) to regulate power
 - 512 kBytes L2



Mr. Wolf will (hopefully) solve problems

- TSMC 40LP - 9mm²
- 64 pin QFN package
- Cluster with
 - 8x RI5CY
 - 2x shared FPUs
 - 64 kByte TCDM
- SoC domain with
 - micro-riscy core to control power modes
 - DC-DC converter (Dolphin) to regulate power
 - 512 kBytes L2
- Chips already back



PULP



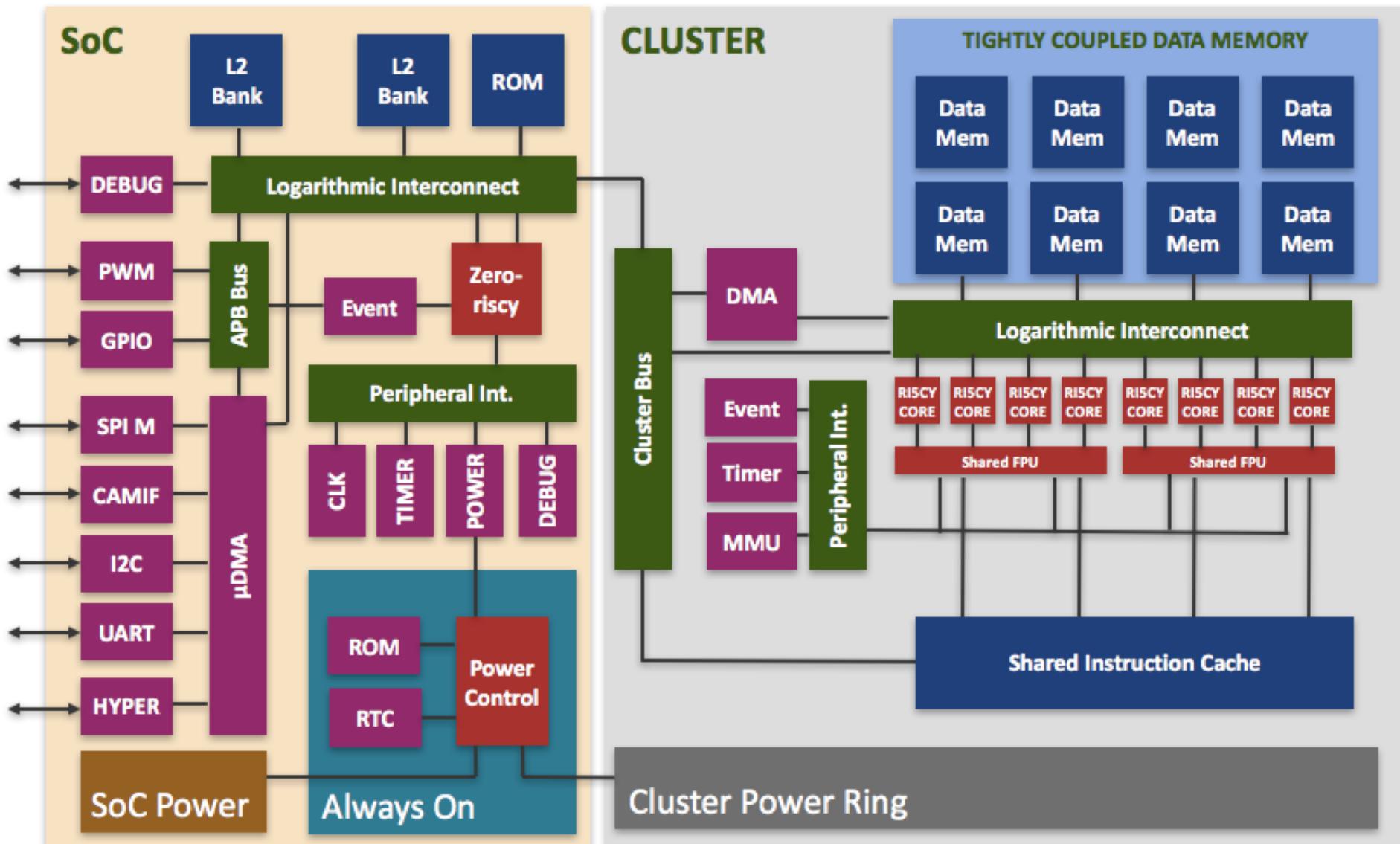
Multithreaded



Open Transprecision Computing

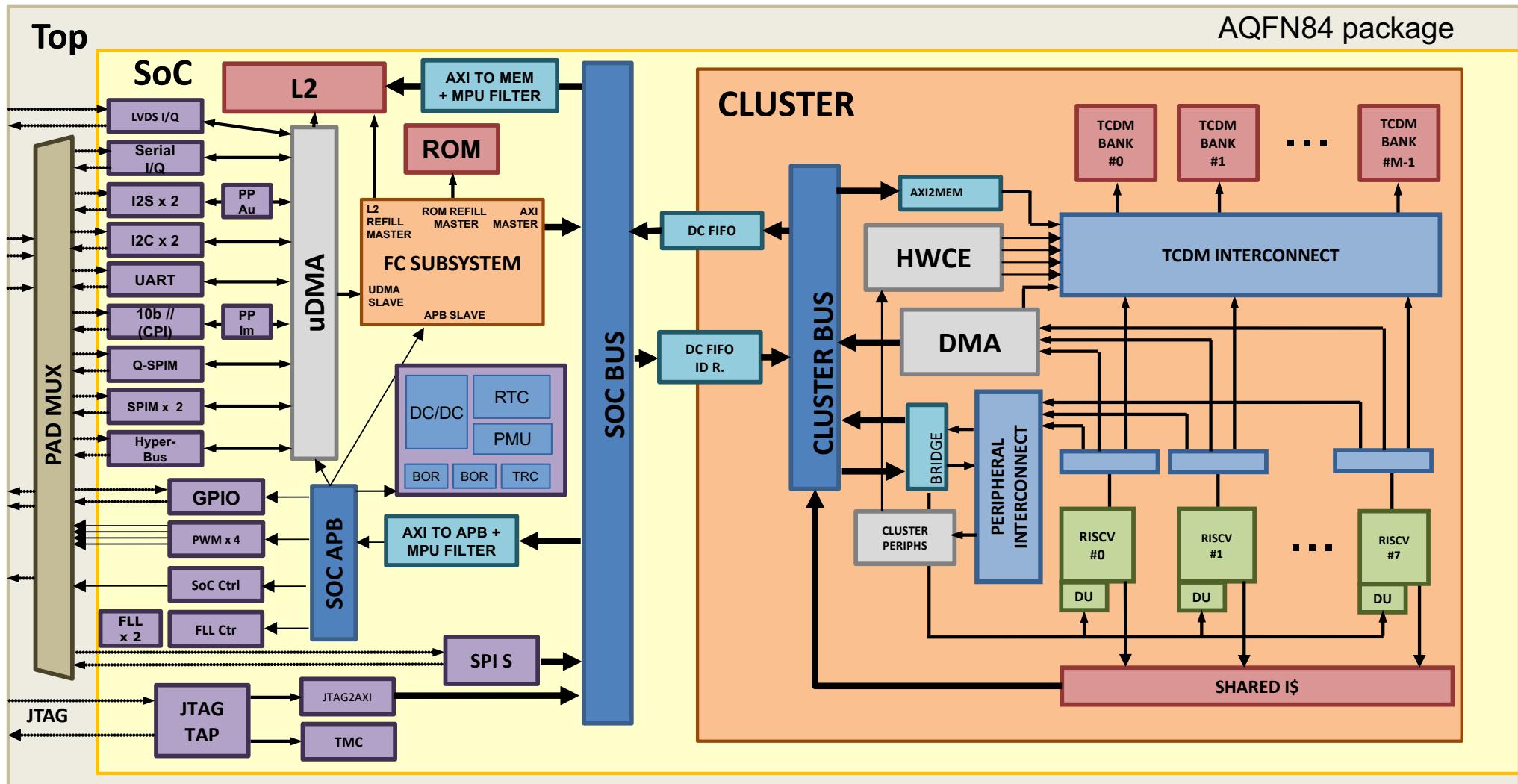


PULP in the IoT domain: Latest version Mr. Wolf



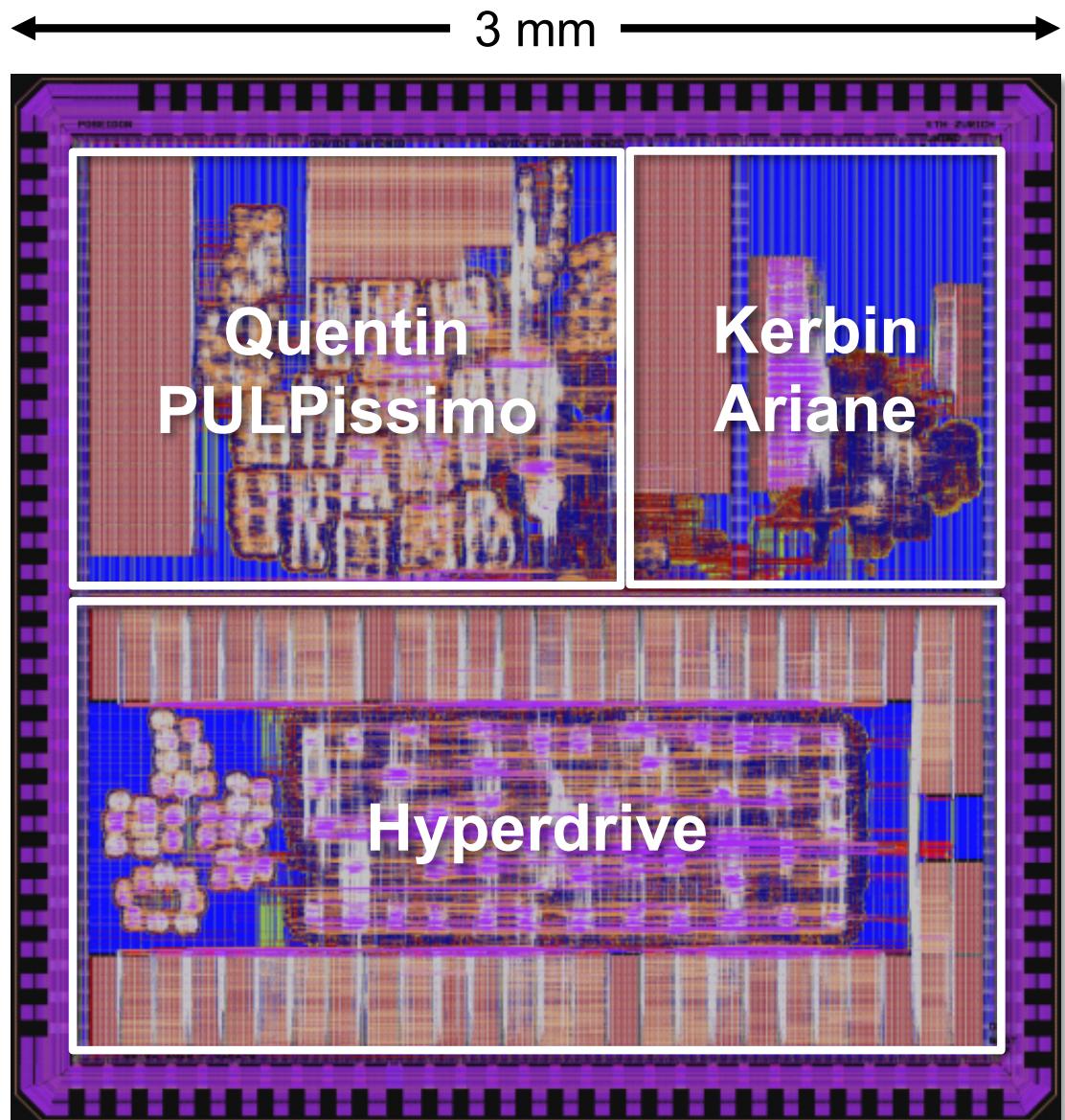
GAP8: commercial big brother of PULP from Greenwaves

Samples arrived, available 1Q2018



Poseidon: Our latest chip in GF 22FDX

- Taped out Jan 2018
- **Quentin**
 - PULPissimo implementation
 - RI5CY+ FPU + HWCE
 - 512 kByte RAM
- **Kerbin**
 - Ariane core + Caches
 - Uses the memory infrastructure of Quentin
- **Hyperdrive**
 - Binary CNN Accelerator



Structure of this workshop

- Open source hardware and our role (Frank)
- The PULP family tree (Frank)
- Our RISC-V cores: Ariane, RI5CY and friends (Florian)
- Break – Demos
- Accelerators in PULP (Francesco)
- Our Programmable Multi-Core Accelerator – HERO (Andreas)
- Programming PULP (Andreas)

Please interrupt at any time to ask questions

QUESTIONS?

