ETH zürich

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Open RISC-V Hardware in Space

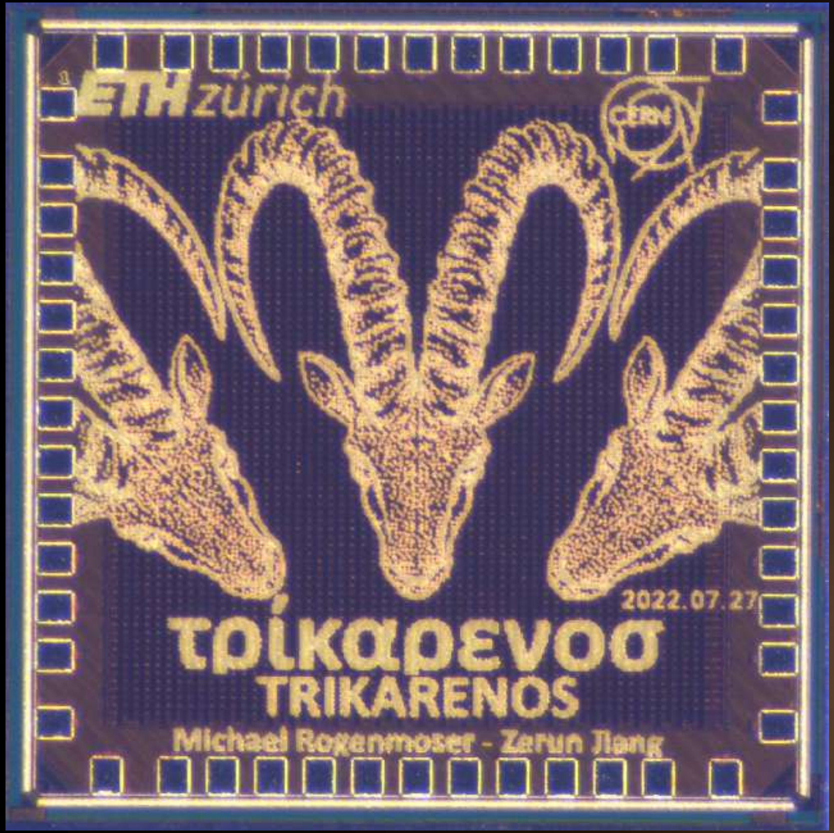
One small step for Space, one giant leap for PULP

Frank K. Gürkaynak kgf@iis.ee.ethz.ch

PULP Platform
Open Source Hardware, the way it should be!

pulp-platform.org
@pulp_platform
company/pulp-platform
youtube.com/pulp_platform

SpaceX Transporter-13 launch, March 15th 2025



Carried our first PULP chip
Trikarenos to space

Aboard the ALICE
experiment by ARIS

Let’s talk about how Open HW and RISC-V in Space

Challenges of Space Applications

- **One small processor in a galaxy far, far away..**
  - Problems that keep us busy: Reliability, communication (latency, bandwidth), need for more compute

Benefits of Open HW and RISC-V for Space Applications

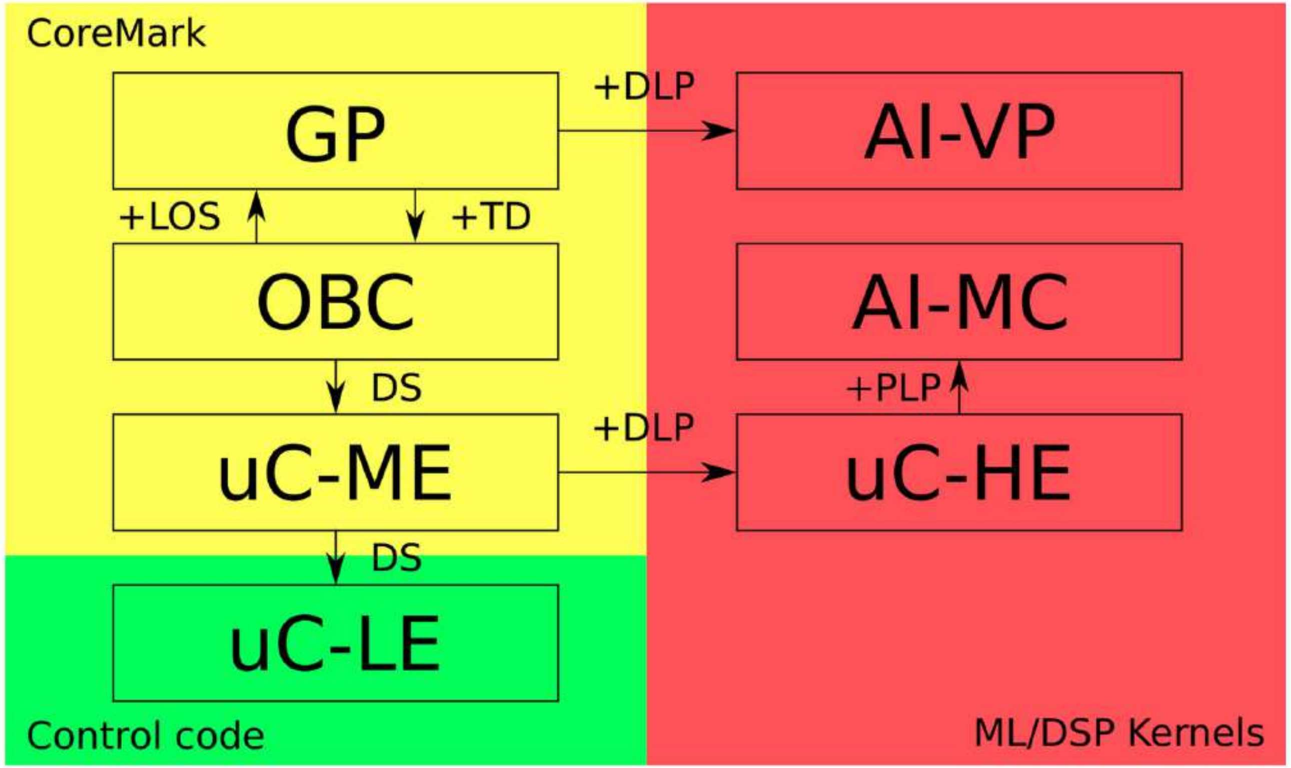
- **Some obvious, some which are not that much talked about**
  - Freedom: to collaborate, to change, to advance

How the PULP team is helping addressing these problems

- **Projects and results from our team**
  - Reliability, how to increase performance using heterogeneous many-core accelerators

Space missions will increasingly rely on processors

- **Not only On Board Computer**
  - Needed for real-time tasks
- **Different levels of microcontrollers**
  - Low/Mid/High end cores
  - For data acquisition/processing tasks
- **General Purpose processors**
  - To orchestrate complex systems
  - Running Linux-like Oses
  - More relaxed real-time tasks
- **And more and more ML/DSP cores**
  - For compute intensive AI applications



G. Furano, S. Di Mascio, A. Menicucci and C. Monteleone, "A European Roadmap to Leverage RISC-V in Space Applications," 2022 IEEE Aerospace Conference (AERO), Big Sky, MT, USA, 2022, pp. 1-7, doi: 10.1109/AERO53065.2022.9843361

Large variation of processor solutions are needed

Many known technical challenges

- **Exposed to radiation**
- **Large temperature variations**
- **Mechanical stresses of launch**
- **Operating in vacuum**
- **Long life time (much longer than commercial products)**
- **No repair possibilities**

Needs many experts solving these problems

- **Low volume, costly supply chain, need for diverse public funding sources**

Calls for efficient ways of collaborating with fewer restrictions

And several new challenges..

- **More and more data to analyze**
  - Can not afford to send all the data back home all the time
    - Limited data bandwidth
    - Long latency

Autonomous systems that can reason with data

- **New algorithms enable additional opportunities for operation**
  - Machine Learning and AI algorithms
  - On-board data processing and comprehension

Space missions can benefit from much more compute

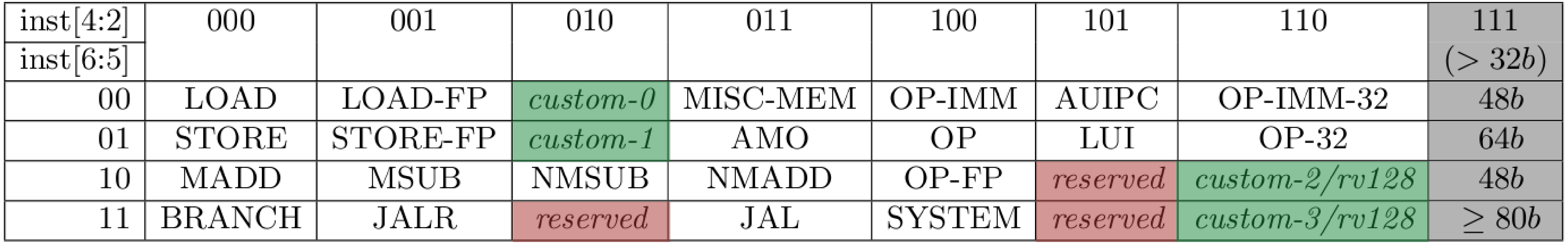
But how is open source HW and RISC-V going to help?

- **RISC-V the open ISA of choice**
  - Originally developed at UC Berkeley
  - Open ISA managed by RISC-V international since 2015
- **Simple Base ISA (RV32 / RV64 / RV128)**
  - Extensions to cover many aspects (vector, matrix..)
- **Open development**
  - Technical working groups where members discuss and propose new extensions
  - Public review and comments, ratified by the Board of Directors
- **Allows processors to be designed and extended easily**
  - While allowing a common SW infrastructure to be built around it.

The ISA is open, implementations can be open or proprietary

Key aspect of RISC-V: space for ISA Extensions

- **RISC-V has Reserved opcodes for standard extensions**
- **Rest of opcodes free for custom implementations**
- **Custom extensions can be standardized**
  - Standard extensions will be frozen/not change in the future



inst[4:2]	000	001	010	011	100	101	110	111
inst[6:5]								$(> 32b)$
00	LOAD	LOAD-FP	*custom-0*	MISC-MEM	OP-IMM	AUIPC	OP-IMM-32	$48b$
01	STORE	STORE-FP	*custom-1*	AMO	OP	LUI	OP-32	$64b$
10	MADD	MSUB	NMSUB	NMADD	OP-FP	*reserved*	*custom-2/rv128*	$48b$
11	BRANCH	JALR	*reserved*	JAL	SYSTEM	*reserved*	*custom-3/rv128*	$\geq 80b$

Extensibility is fundamental in the RISC-V ISA!

Extensions at work: Achieving ~100% **dotp** Unit Utilization

8-bit Convolution

HW Loop

LD/ST with post increment

8-bit SIMD sdotp

8-bit sdotp + LD

RV32IMC

N

```
addi   a0,a0,1
addi   t1,t1,1
addi   t3,t3,1
addi   t4,t4,1
lbu    a7,-1(a0)
lbu    a6,-1(t4)
lbu    a5,-1(t3)
lbu    t5,-1(t1)
mul    s1,a7,a6
mul    a7,a7,a5
add    s0,s0,s1
mul    a6,a6,t5
add    t0,t0,a7
mul    a5,a5,t5
add    t2,t2,a6
add    t6,t6,a5
bne    s5,a0,1c000bc
```

RV32IMCXpulp

N/4

```
lp.setup
p.lw  w1, 4(a0!)
p.lw  w2, 4(a1!)
p.lw   x1, 4(a2!)
p.lw   x2, 4(a3!)
pv.sdotsp.b   s1, w1, x1
pv.sdotsp.b   s2, w1, x2
pv.sdotsp.b   s3, w2, x1
pv.sdotsp.b   s4, w2, x2
end
```

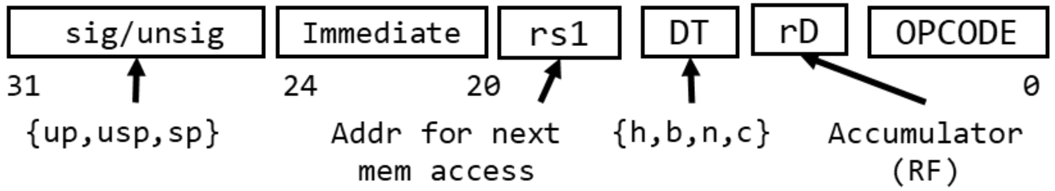
can we remove?

Yes! `dotp+ld`

N/4

```
Init NN-RF (outside of the loop)
lp.setup
pv.nnsdotup.h  s0,ax1,9
pv.nnsdotsp.b  s1, aw2, 0
pv.nnsdotsp.b  s2, aw4, 2
pv.nnsdotsp.b  s3, aw3, 4
pv.nnsdotsp.b  s4, ax1, 14
end
```

`pv.nnsdot{up,usp,sp}.{h,b,n,c} rD, rs1, Imm`



9x less instructions than RV32IMC

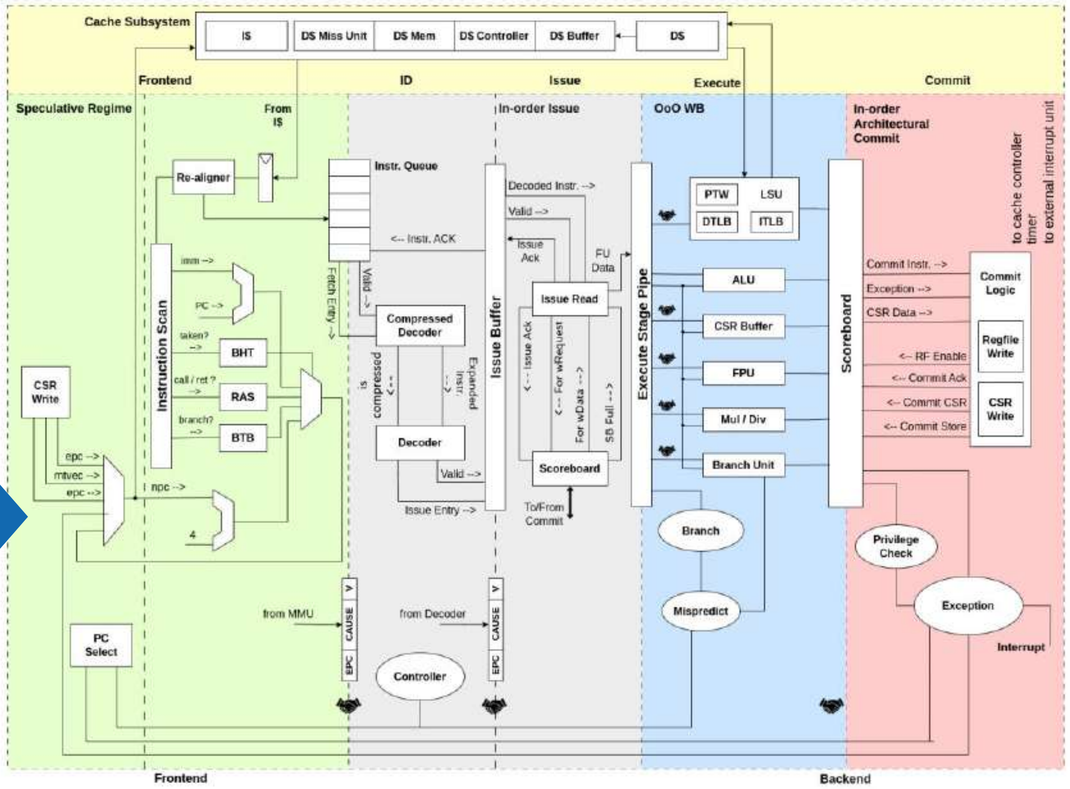
14.5x less instructions at an extra 3% area cost (~600GEs)

But there are more important aspects for RISC-V

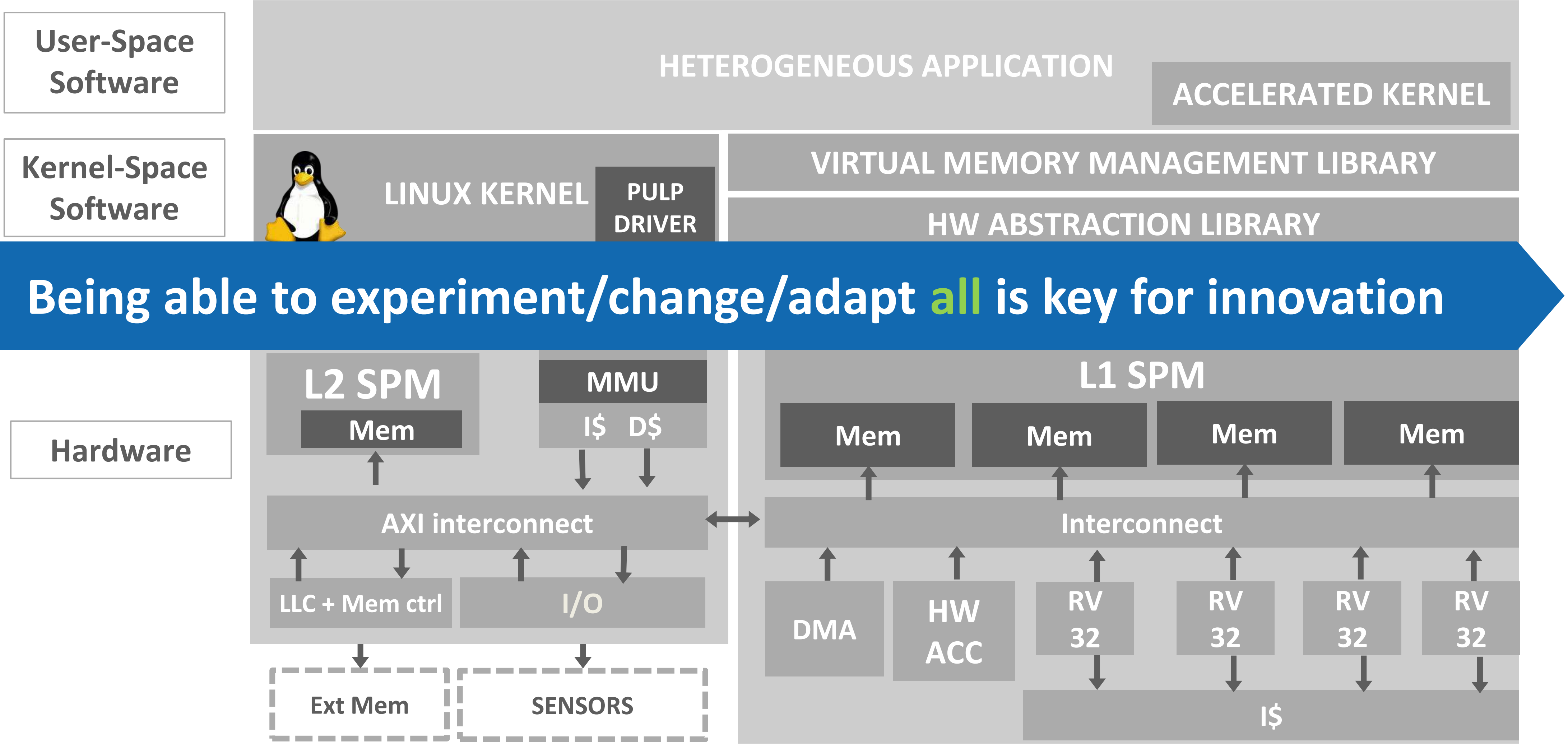
- **Modern processors are complex**
  - Make use of many tricks: Caches, reorder buffers, branch predictors, load/store queues
- **Proprietary&traditional designs are optimized**
  - Proven to work well and are robust
- **They are what they are, you take them as is**
  - Unless your use case has a large enough market you will be limited to what is available

This is also true for proprietary RISC-V

- **The real interesting bit is the freedom**
  - Do things differently, evaluate if they are good/bad
  - Share these ideas, allow good ones to be picked up by companies



A processor core is but one part of a modern SoC



Being able to experiment/change/adapt all is key for innovation

As an example for space, it is not performance at any cost

- **Earlier talk by Yvan:**
  - Astral: a Mixed-Criticality RISC-V SoC Architecture for Satellite Onboard AI
- **Such work is only possible**
  - If you can have deep reaching access to the entire SoC to make changes that are necessary
  - Not possible with proprietary designs (RISC-V or not) unless the company is part of it

Open source HW implementations are necessary for independence

- **Such work is relevant if**
  - You can demonstrate that it works on higher TRL levels (6+ for ESA)
  - Not easy to do for any individual group

Silicon proven SoC templates allowing easier collaboration is essential

The PULP project at ETH Zürich and University of Bologna

- **Research on open-source energy-efficient computing**

How is our work addressing problems in space applications?

Our research focus: cluster-based many-core accelerators

Multiple Scales of acceleration

Extensions to processor cores

- Explore new extensions
- Efficient implementations

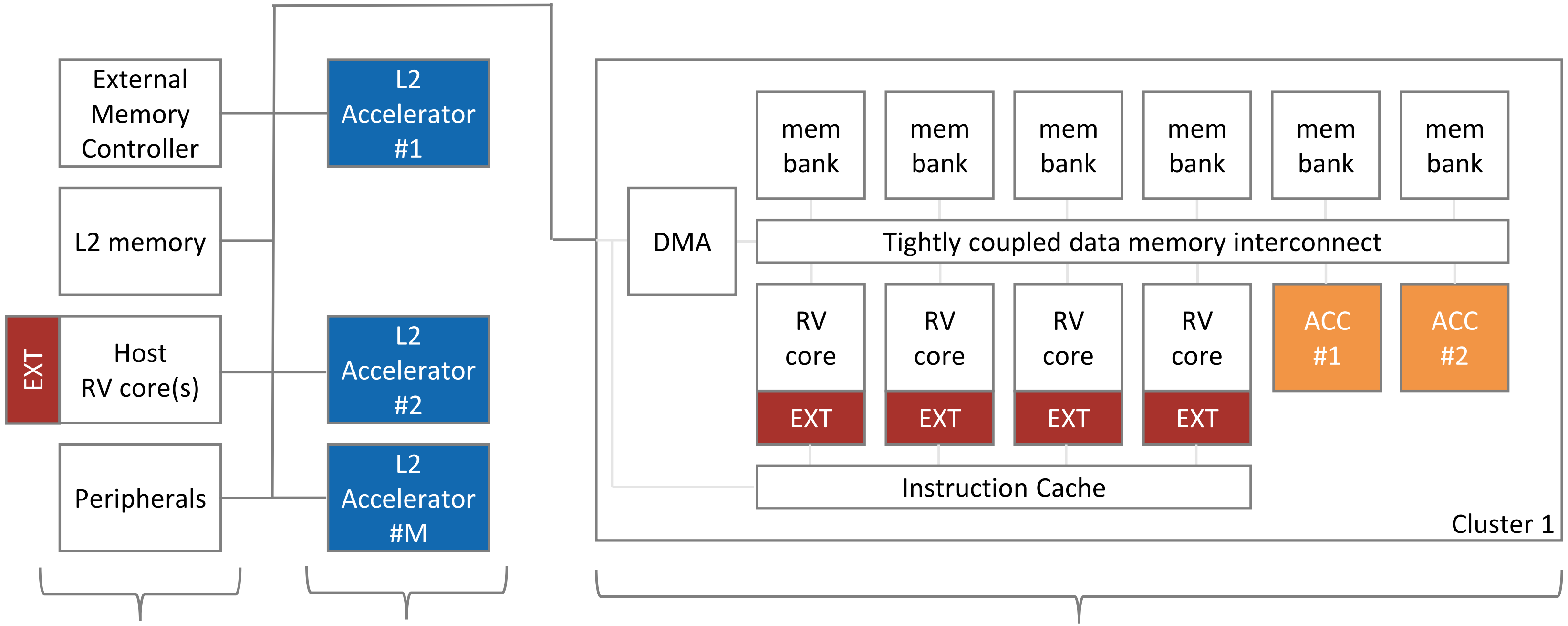
Shared-memory Accelerators

- Domain specific
- Local memory

Multiple Decoupled Accelerators

- Communication
- Synchronization

High-speed on-chip interconnect (NoC, AXI, other..)



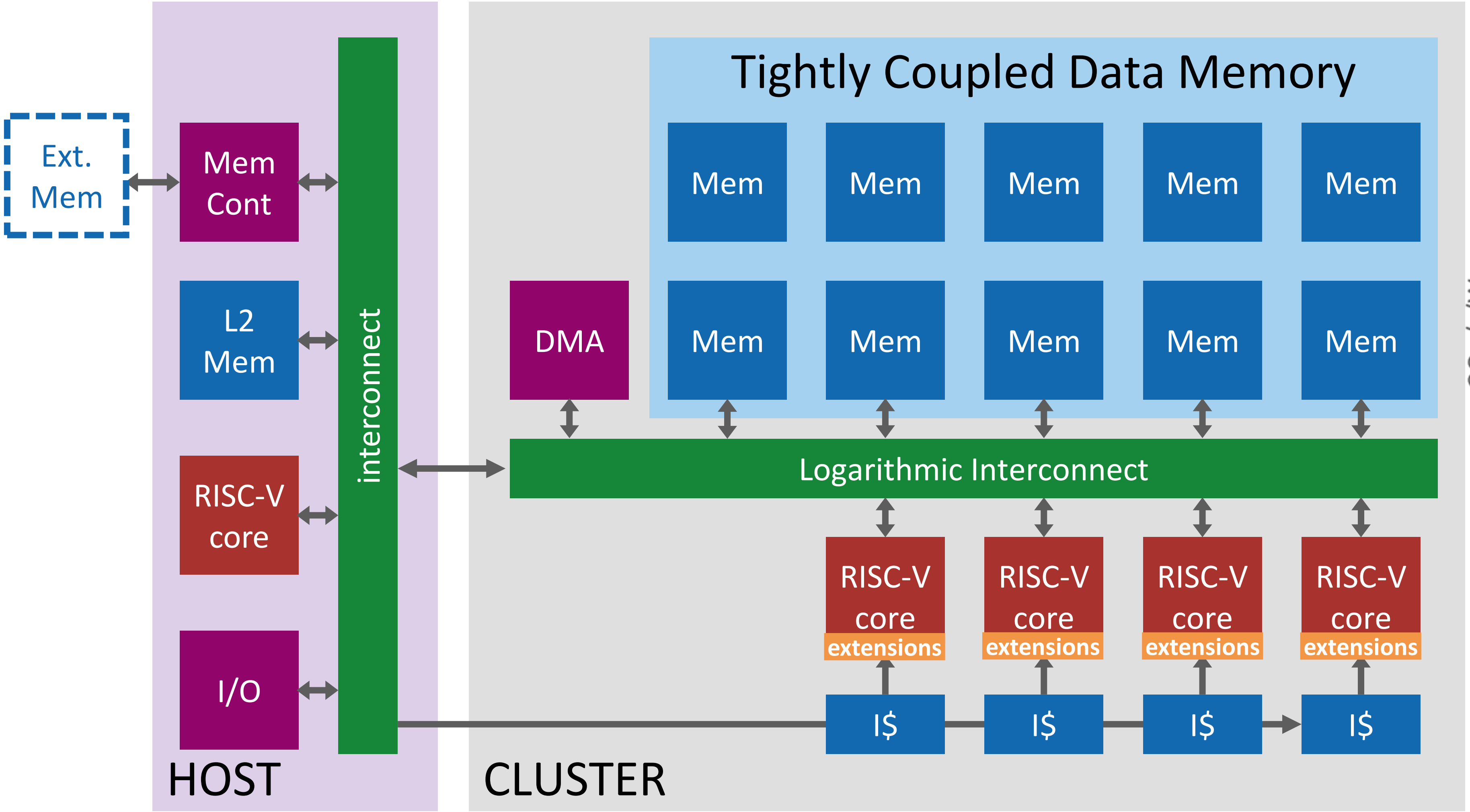
Host, L2, L3 IOs

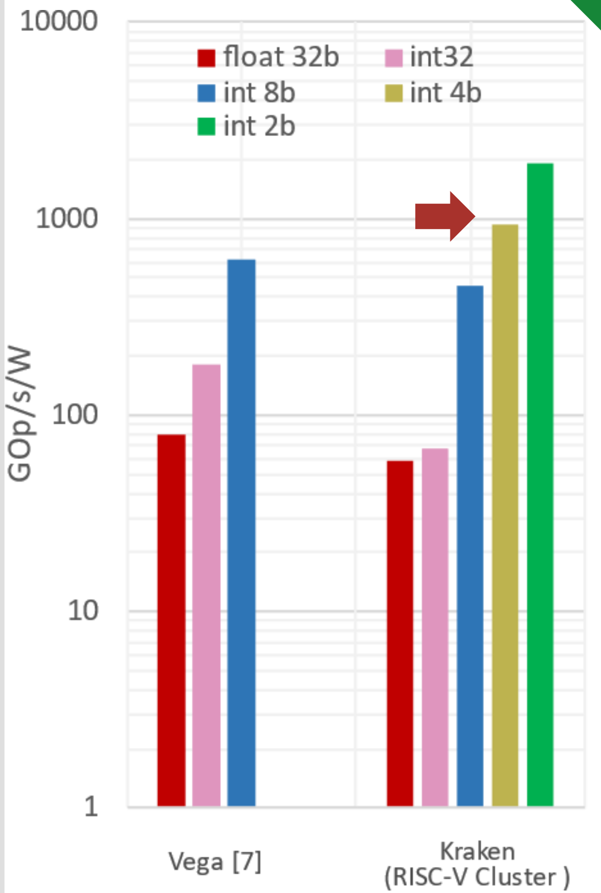
Decoupled accelerators

Computing cluster with tightly coupled accelerators

RISC-V is a key enabler → max agility, enabling SW build-up, without vendor lock-in

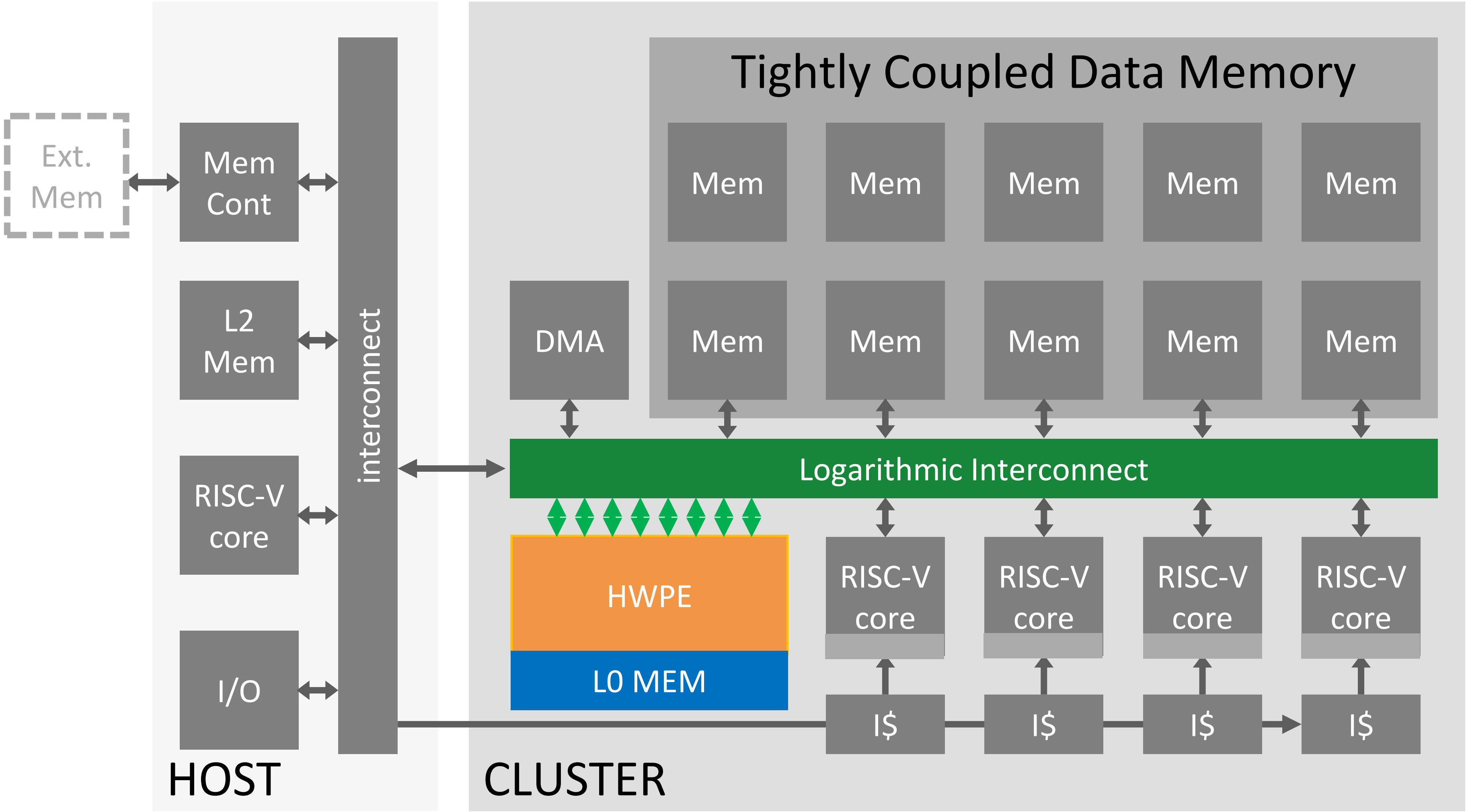
PULP Paradigm: A PE cluster accelerates a host system





1TOP/s/W
2b/4b OPS

Tightly-coupled accelerators support the cluster



Specialization in perspective

Kraken: Using 22FDX tech, NT@0.6V, High utilization, minimal IO & overhead

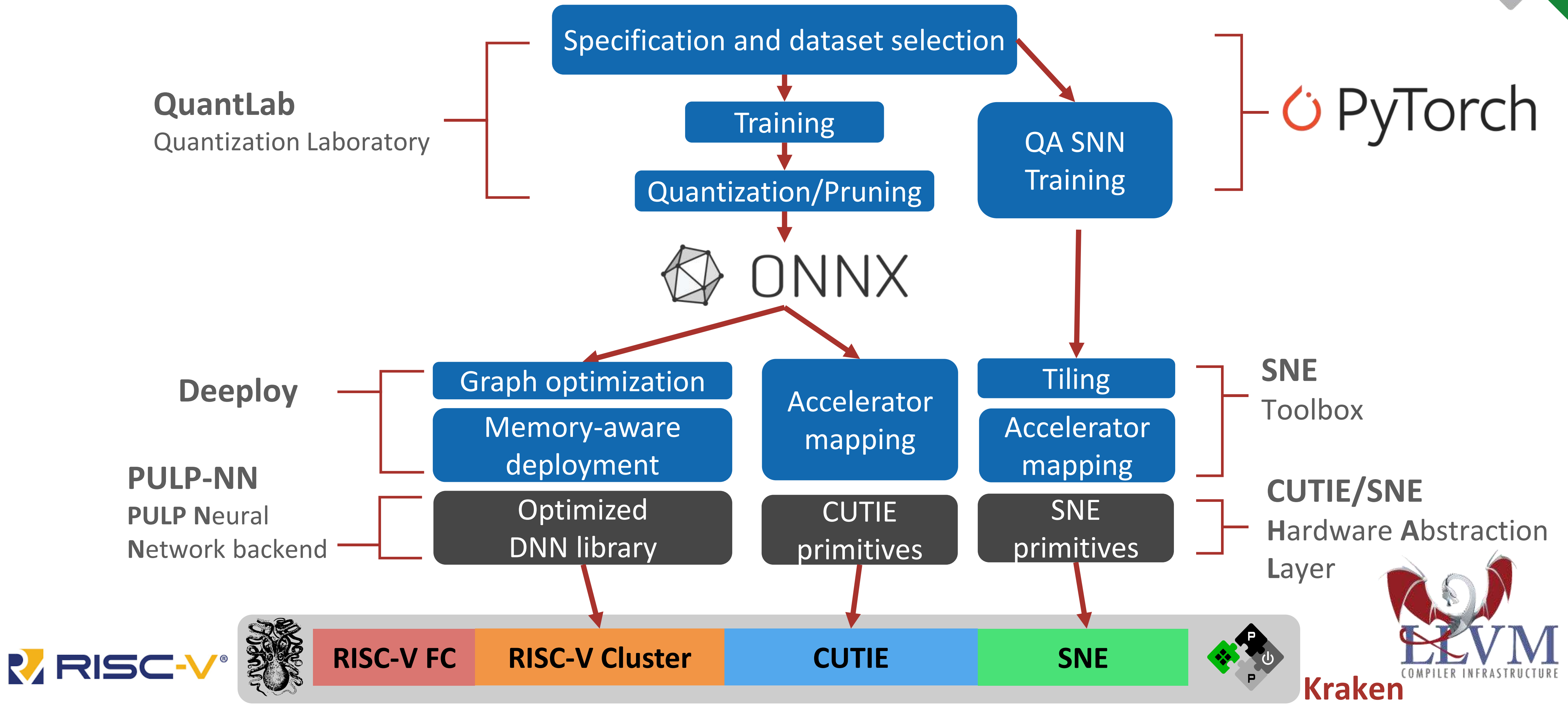
Energy-Efficient RISC-V Core → **20pJ (8bit)**

ISA-based 10-20x →**1pJ (4bit)** → **XPULP**

Configurable DP 10-20x → **100fJ (4bit)** → **RBE**

Highly specialized DP 100x →**1fJ (ternary)** → **CUTIE, SNN**

Fully Open-Source Deployment Flow!



PULP has developed a toolbox to make efficient SoCs

RISC-V Cores and Vector Units

RI5CY	Zero R	Snitch	Spatz	Ariane	ARA
CV32E	*Ibex*			*CVA6*	
RV32	RV32	RV32	RVV	RV64	RVV

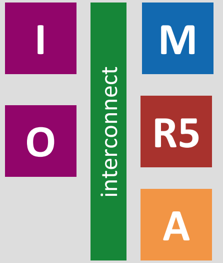
Peripherals

- JTAG
- SPI
- UART
- I2S
- DMA
- GPIO

Interconnects

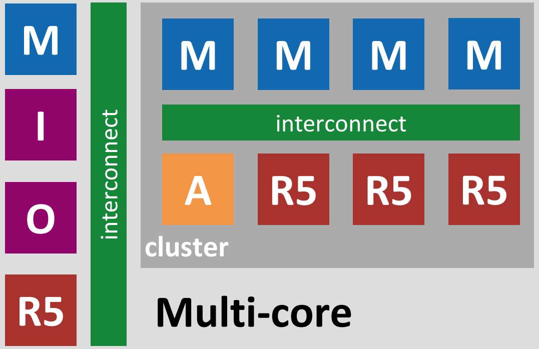
- LIC
- HCI
- APB
- FlooNoC
- AXI4

Platforms



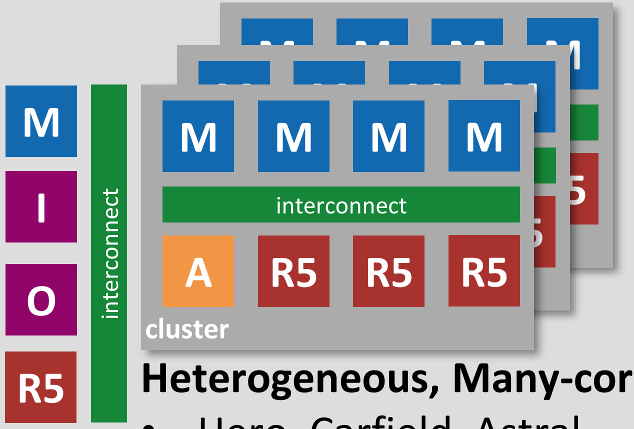
Single core

- PULPino, PULPissimo
- Cheshire



Multi-core

- OpenPULP
- ControlPULP



Heterogeneous, Many-core

- Hero, Carfield, Astral
- Occamy, Mempool

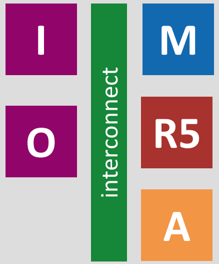
IOT → HPC

Accelerators and ISA extensions

- XpulpNN, XpulpTNN
- ITA (Transformers)
- RBE, NEUREKA (QNNs)
- FFT (DSP)
- REDMULE (FP-Tensor)

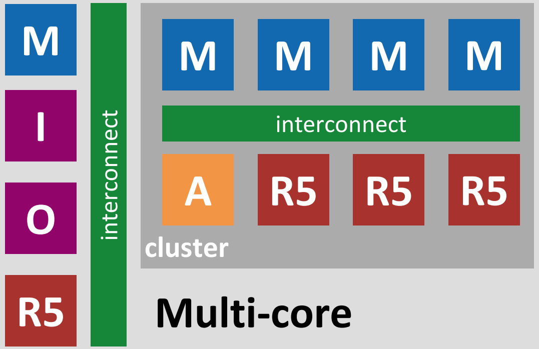
Which have been demonstrated in more than 60 ICs

Platforms



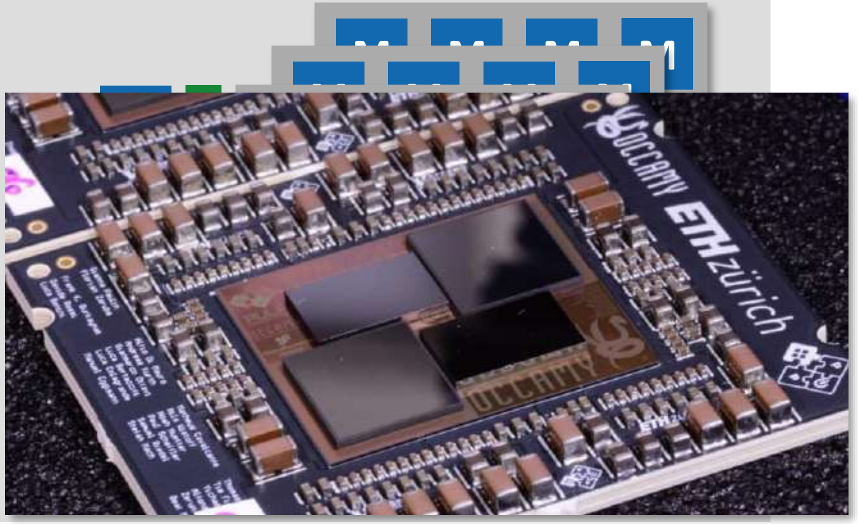
Single core

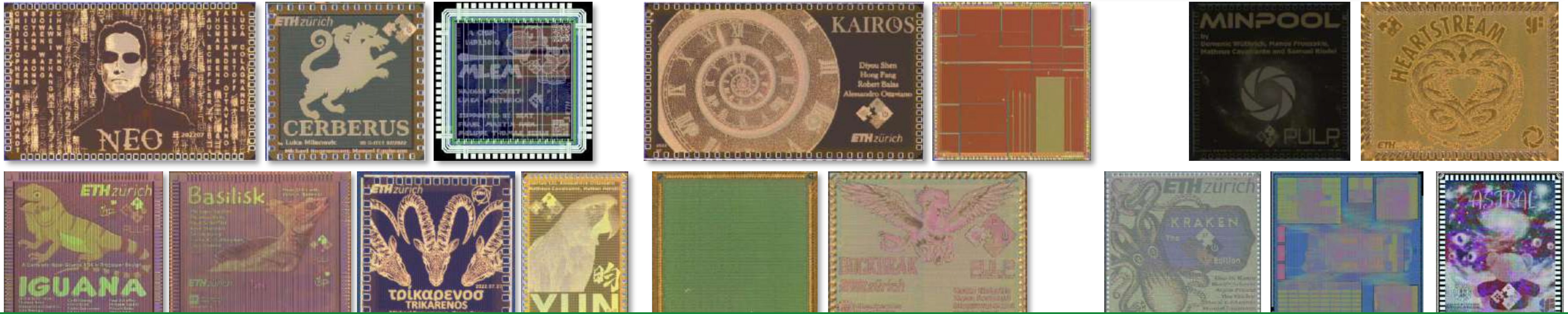
- PULPino, PULPissimo
- Cheshire



Multi-core

- OpenPULP
- ControlPULP





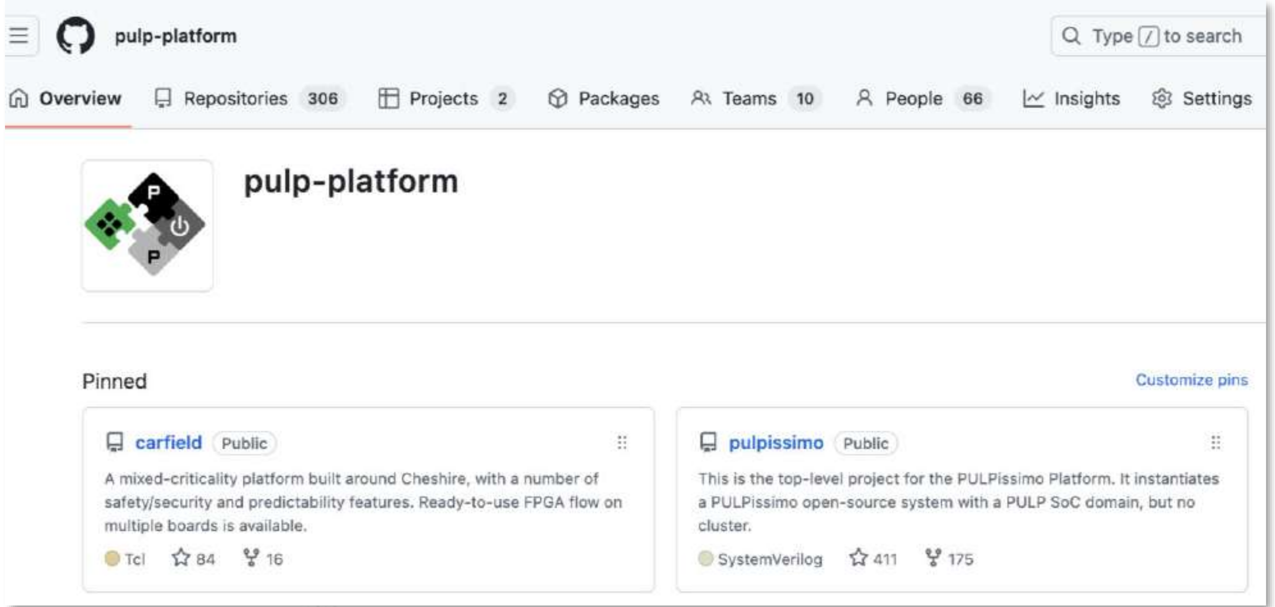
See our chip gallery for all our chips: http://asic.ethz.ch

All of our designs are open-source hardware

- **All our development is on GitHub using a permissive license**
  - HDL source code, testbenches, software development kit, virtual platform

https://github.com/pulp-platform

- Allows anyone to use, change, and make products without restrictions.



pulp-platform

Type / to search

Overview | Repositories 306 | Projects 2 | Packages | Teams 10 | People 66 | Insights | Settings

pulp-platform

Pinned

Customize pins

carfield Public

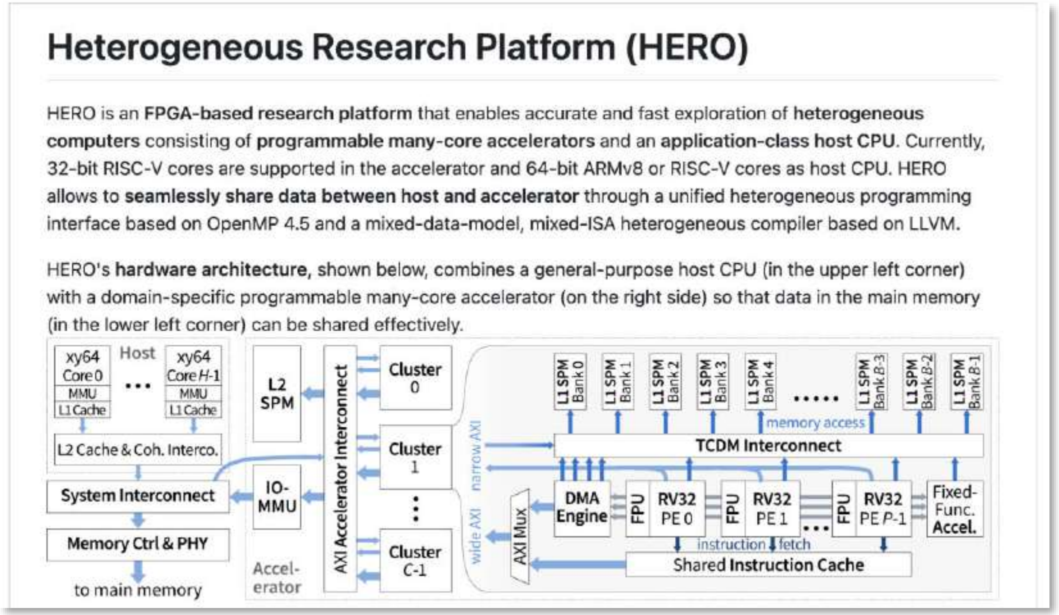
A mixed-criticality platform built around Cheshire, with a number of safety/security and predictability features. Ready-to-use FPGA flow on multiple boards is available.

Tcl ☆ 84 16

pulpissimo Public

This is the top-level project for the PULPissimo Platform. It instantiates a PULPissimo open-source system with a PULP SoC domain, but no cluster.

SystemVerilog ☆ 411 175



Heterogeneous Research Platform (HERO)

HERO is an **FPGA-based research platform** that enables accurate and fast exploration of **heterogeneous computers** consisting of **programmable many-core accelerators** and an **application-class host CPU**. Currently, 32-bit RISC-V cores are supported in the accelerator and 64-bit ARMv8 or RISC-V cores as host CPU. HERO allows to **seamlessly share data between host and accelerator** through a unified heterogeneous programming interface based on OpenMP 4.5 and a mixed-data-model, mixed-ISA heterogeneous compiler based on LLVM.

HERO's **hardware architecture,** shown below, combines a general-purpose host CPU (in the upper left corner) with a domain-specific programmable many-core accelerator (on the right side) so that data in the main memory (in the lower left corner) can be shared effectively.

Current research focus on reliable, scalable architectures

Reliable, safe and secure architectures

- Supporting mixed-criticality systems, trade-off between performance and reliability
- Better/faster virtualization support: vCLIC, cache partitioning
- Efficient implementations of RISC-V extensions:
  Zicfiss: Control-Flow Integrity Shadow Stack, **Zicfilp**: Landing Pads

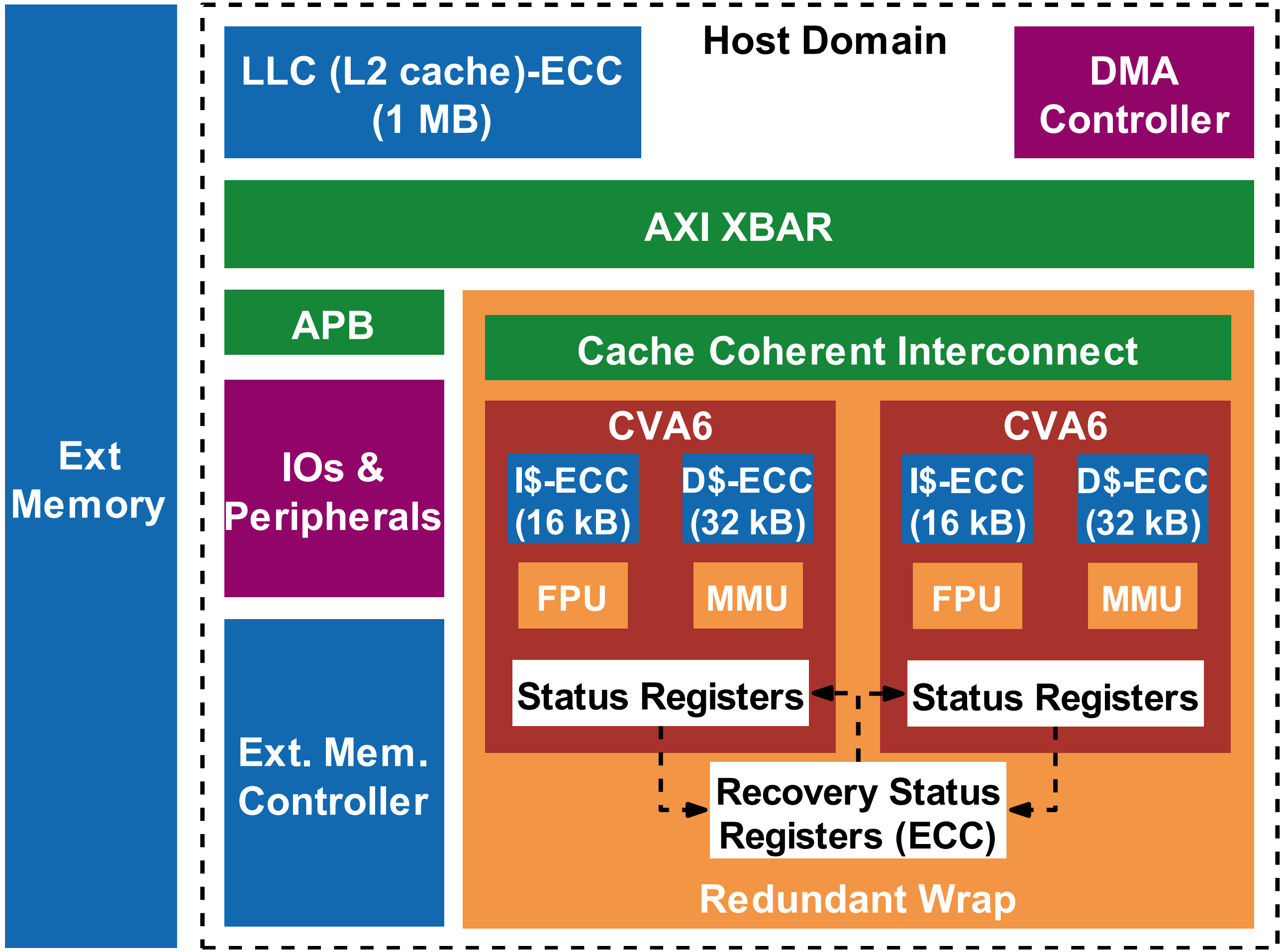
More efficient computing

- Support for various data types
- Vector units for different applications
- Heterogeneous computing, adding configurable accelerators

Scaling up compute to 100s and 1000s of cores

- Data transport solutions: NoC, working with sparse data, transforming data in transit

Reliability features for everyone



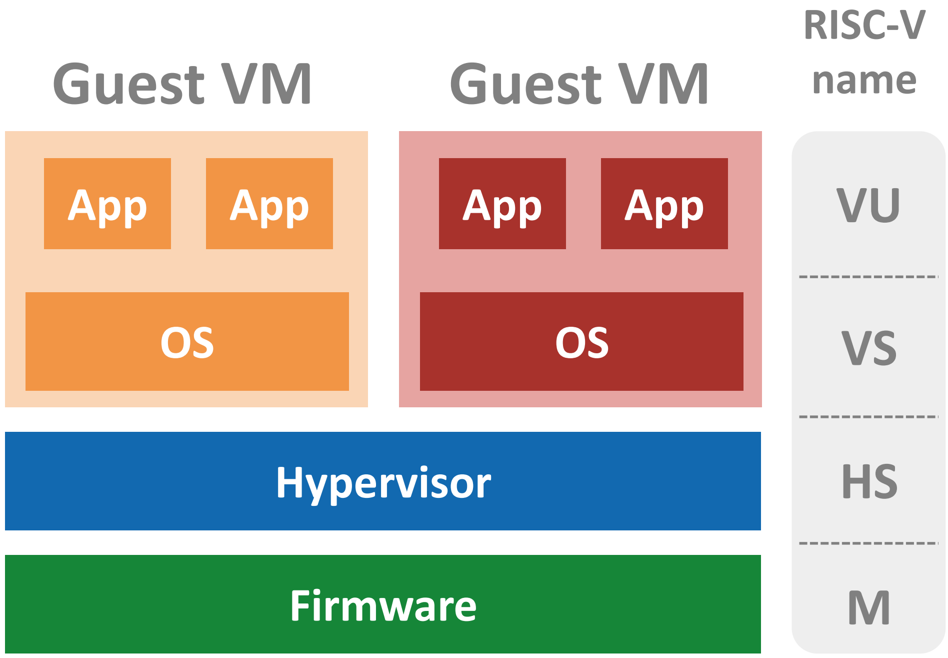
Goal: develop a toolbox of reliability features to harden cores against faults

Milestones:

- **Core-wide ECC protection of vulnerable elements** (I$, D$, MMU, Branch Predictor, etc.)
- **Rapid recovery mechanism** for execution rollback in lockstep mode
- **SW managed split/lock mode** on reset
- Support for fast virtual interrupts

RISC-V mixed-criticality systems

- **Integrate applications with different criticality levels**
  - Trade-off performance vs reliability
  - **Challenges**: scalability, cost, connectivity of 100s of components
- ***Virtualization* is key**
  - Supported in RISC-V through **Hypervisor extension**
  - Improves efficiency by sharing hardware resources
  - **Challenges**: isolation, **real-time responsiveness**
  - Needs HW support to reduce performance overhead
    - i.e. reduce hypervisor intervention by reducing interrupt latency

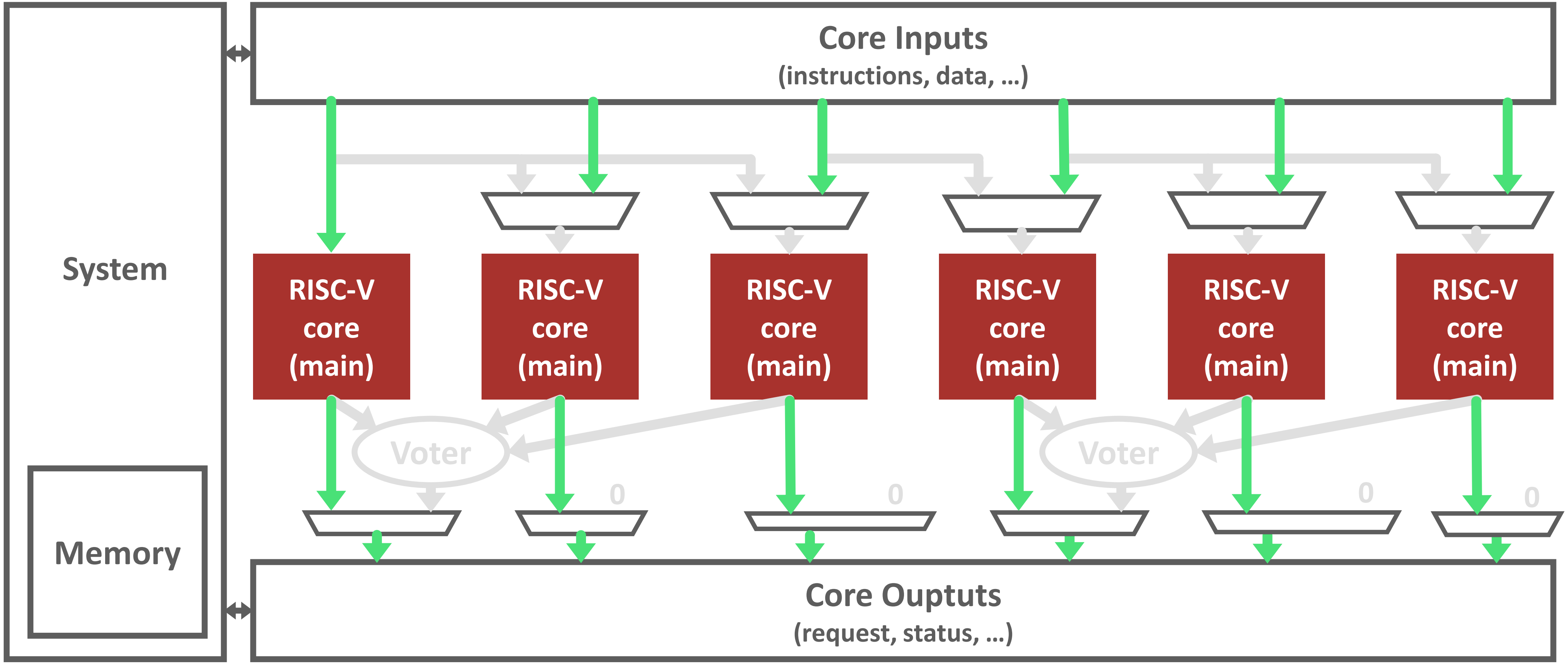


RISC-V® **Privilege Modes**

- **Research Question**
  - How to guarantee real-time responsiveness of safety-critical components in virtualized MCS?

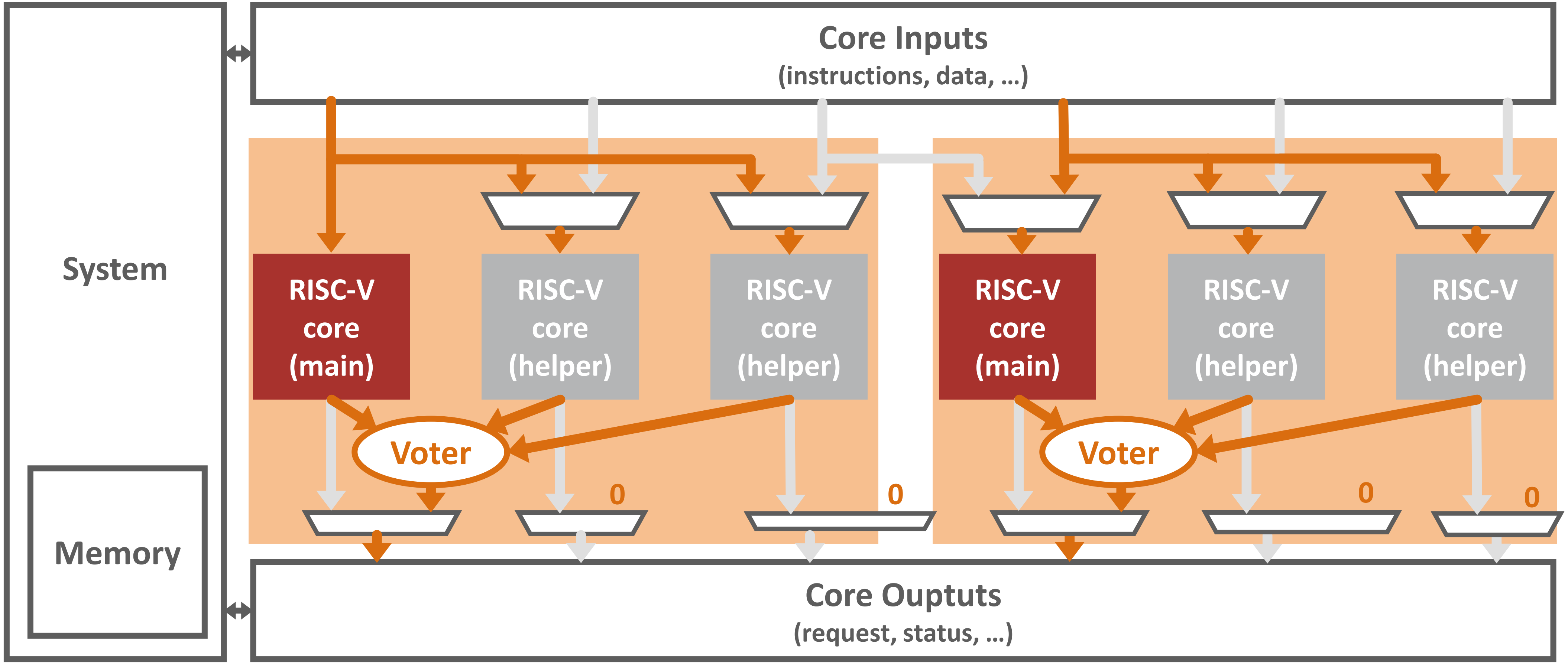
Hybrid Modular Redundancy (HMR): Reconfigurable

Independent Mode: high performance, no reliability



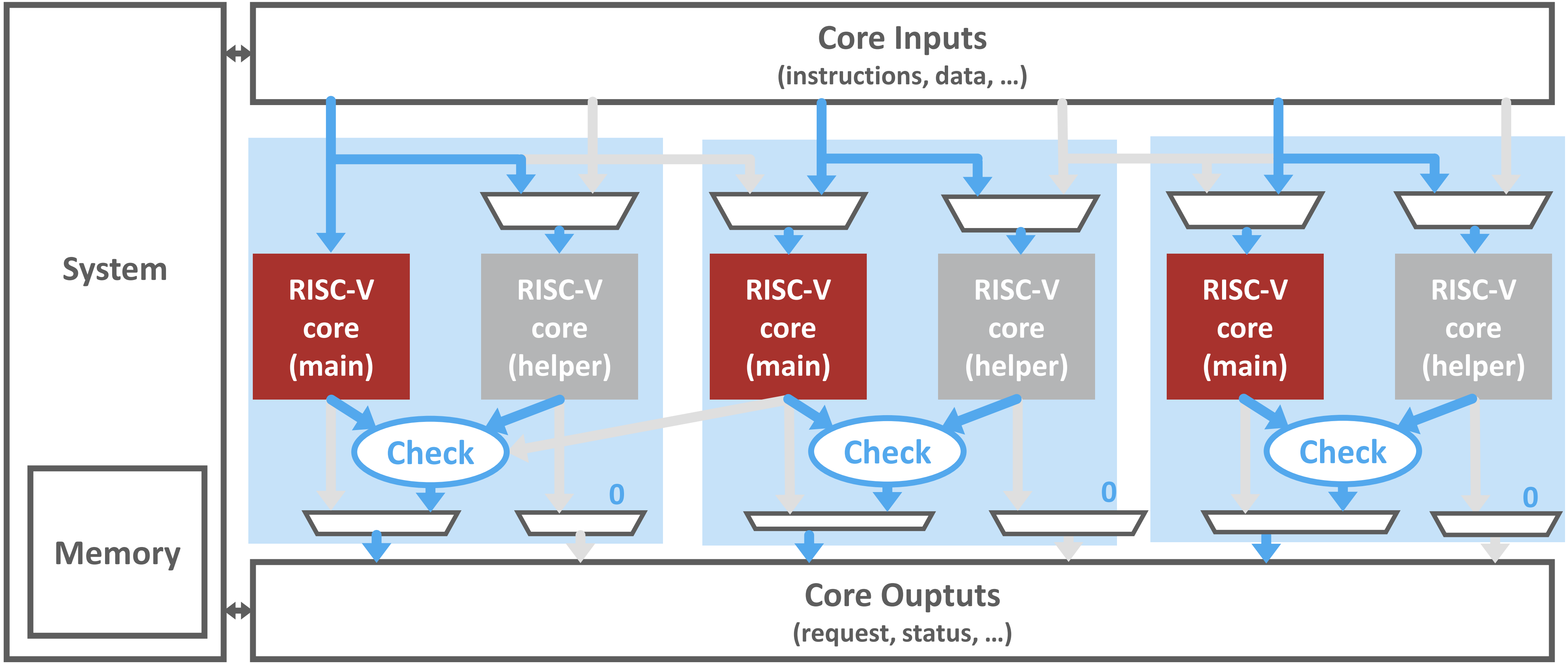
Hybrid Modular Redundancy (HMR): Reconfigurable

TMR Mode: low performance, high reliability, quick recovery



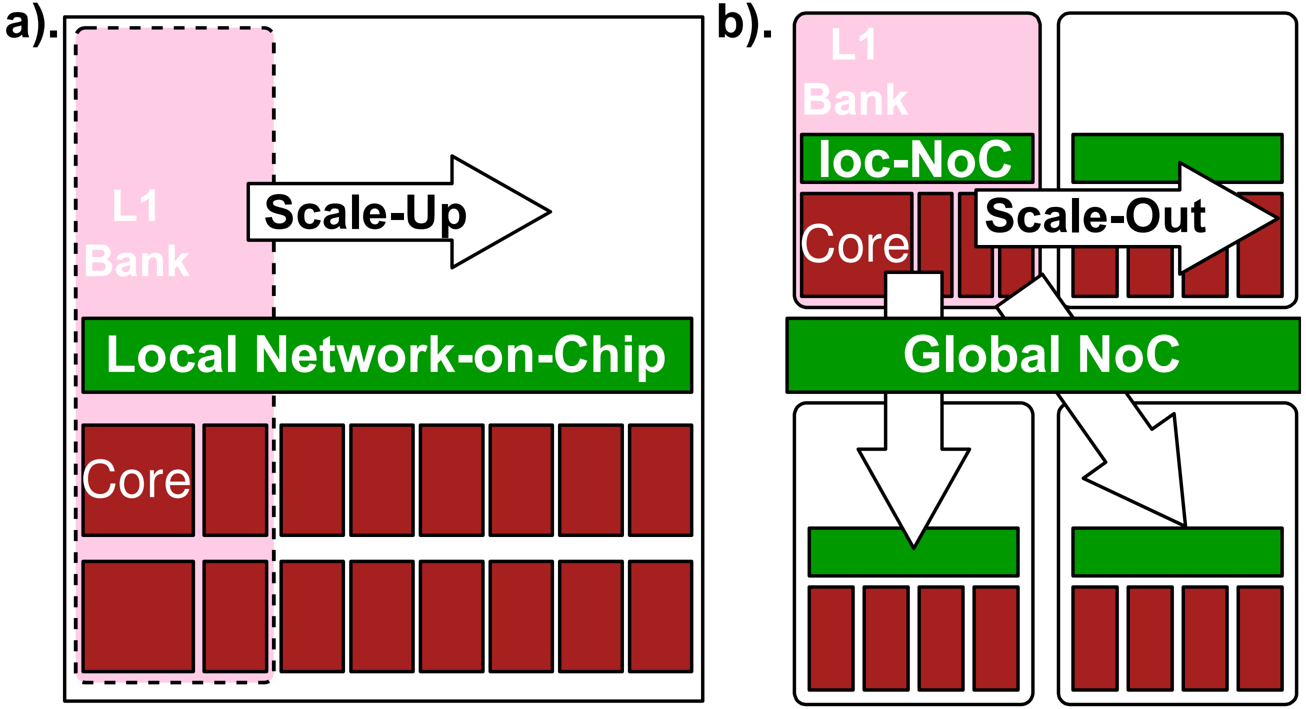
Hybrid Modular Redundancy (HMR): Reconfigurable

DMR Mode: good performance, good reliability, slow recovery



Scaling questions for many-core clusters

- **BIG workload -> MANY cores + BIG memory**
- **Do we scale-up or scale-out ?**

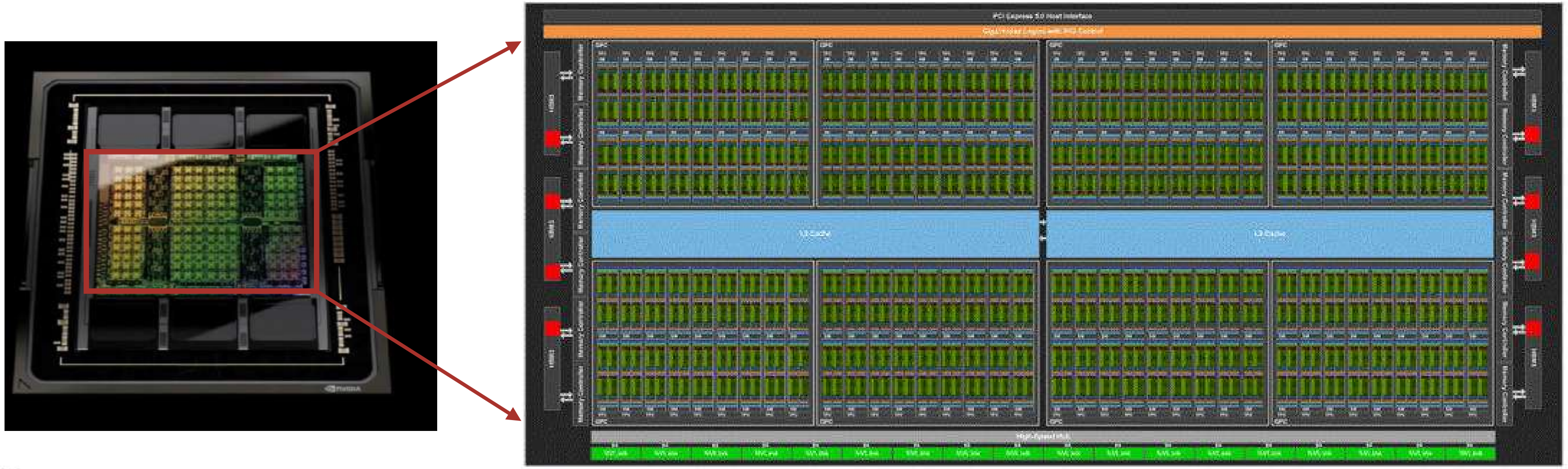


- **Challenges remain similar**
  - Low-latency memory access;
  - High computing utilization;
  - Assure physical feasibility
  - Scale to as many cores as possible
- **And most importantly**
  - Programmers need to be able to work with our innovative architectures!

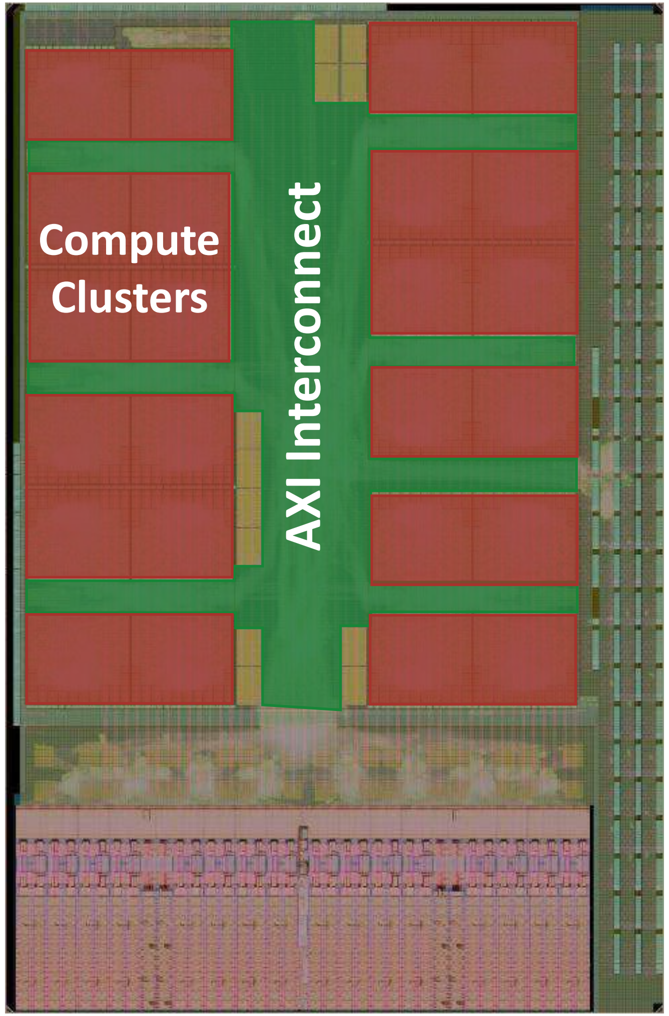
We work on various solutions to support scaling of many-core clusters

Scaling UP/OUT: Challenges in memory hierarchy

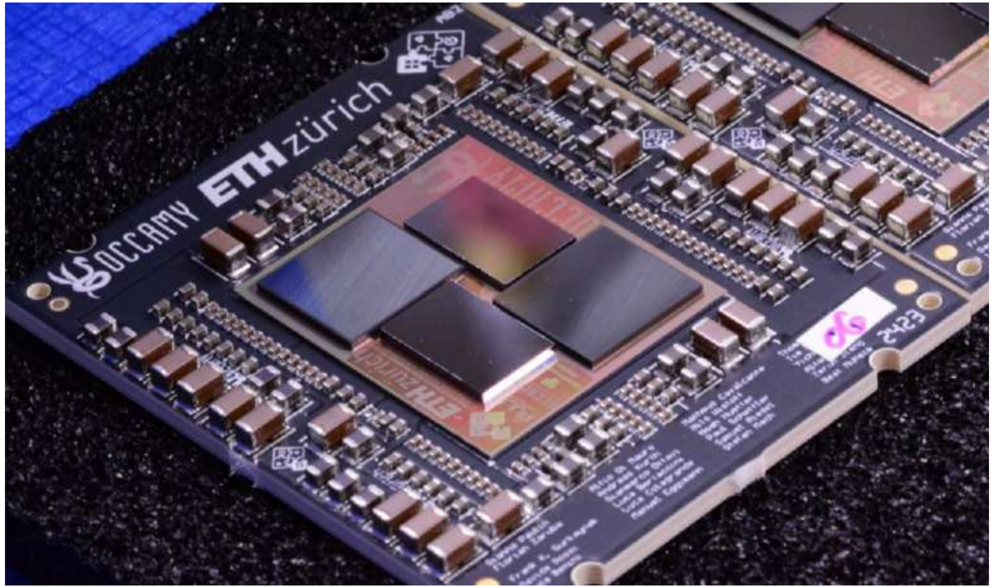
- **Instruction Memory Bottlenecks**
  - Demand for domain-specific ISAs to manage 1000s of operations with single instructions.
  - Amortizing instruction fetches reduces energy impact on the instruction memory.
- **Data Memory Bottlenecks (The "Memory Wall")**
  - High-bandwidth demands from PEs challenge traditional memory architectures.
  - Scaling KiloPEs requires novel solutions beyond classical cache hierarchies.


PCI Express 5.0 Host Interface
GigaThread Engine with MIG Control
GPC
L2 Cache
L2 Cache
HBM3
Memory Controller
High-Speed Hub

Addressing interconnect scalability



- **In our earlier Occamy chiplet design in GF12**

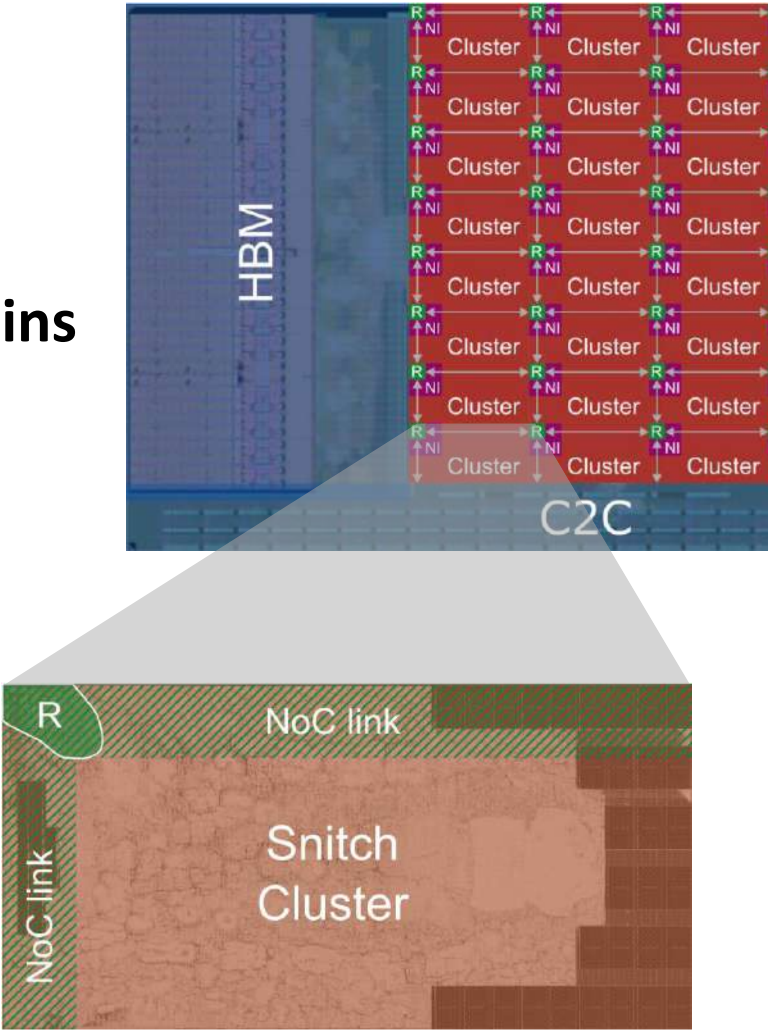


- **Fat-tree was very challenging during implementation**
  - AXI has severe scalability issues
  - Top-level Xbar had to be split up
  - Still, interconnect takes up almost **40%***

**HBM & C2C excluded*

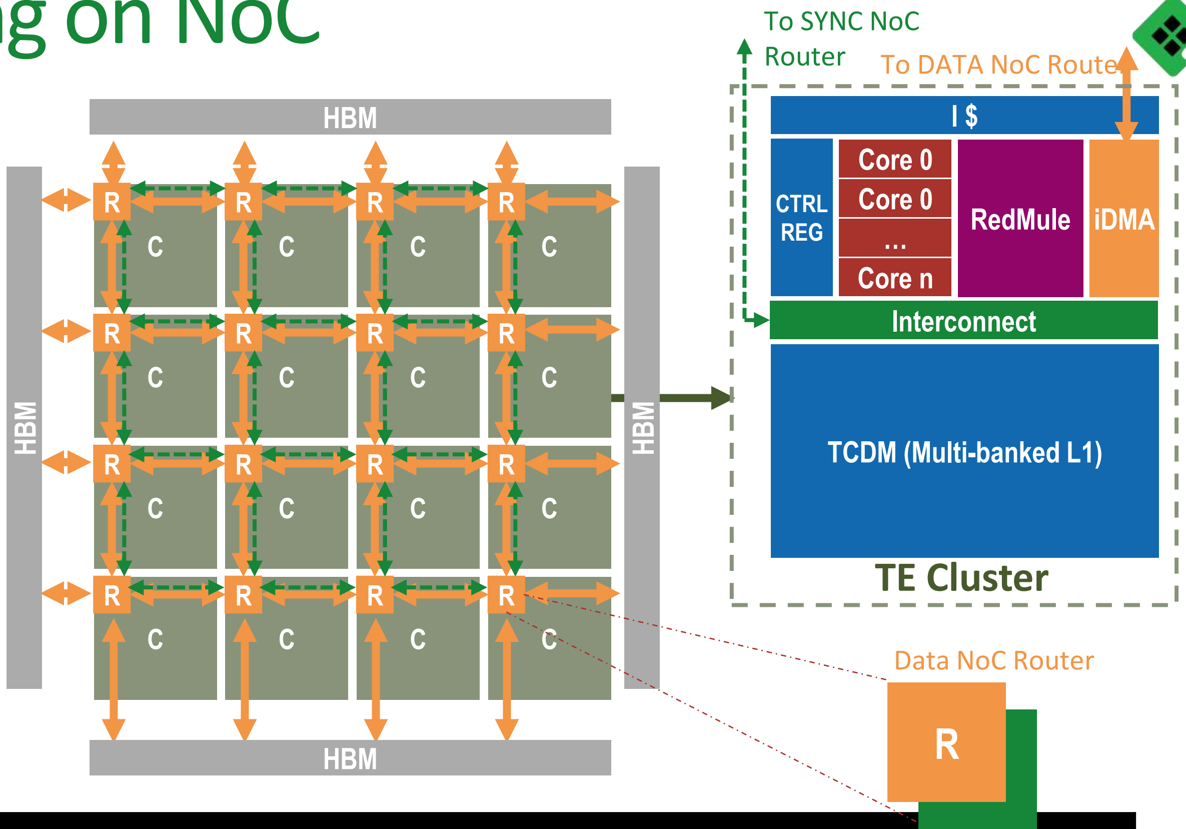
We need a NoC to transfer data between clusters

- **FlooNoC: Wide link NoC**
  - 1000+ parallel wires
  - Routing resources on the side of the clusters
- **Potential for big area/performance gains**
  - Only **~10%** interconnect area
  - **66%** more clusters, same floorplan
  - *High Bandwidth*: **629Gbps/link**
  - *High Energy-Efficiency*: **0.19pj/B/hop**



Multi Head Attention Mapping on NoC

- **Dataflow Schedule of MHA**
  - Multiple clusters in parallel
  - We leverage all-cluster L1 for single head attention
- **Gen.AI specialized NoC**
  - Matrix transpose engine for transposition of (K -> K^T)
  - Collective operations on NoC



0.52x Die Size

0.66x HBM Stacks

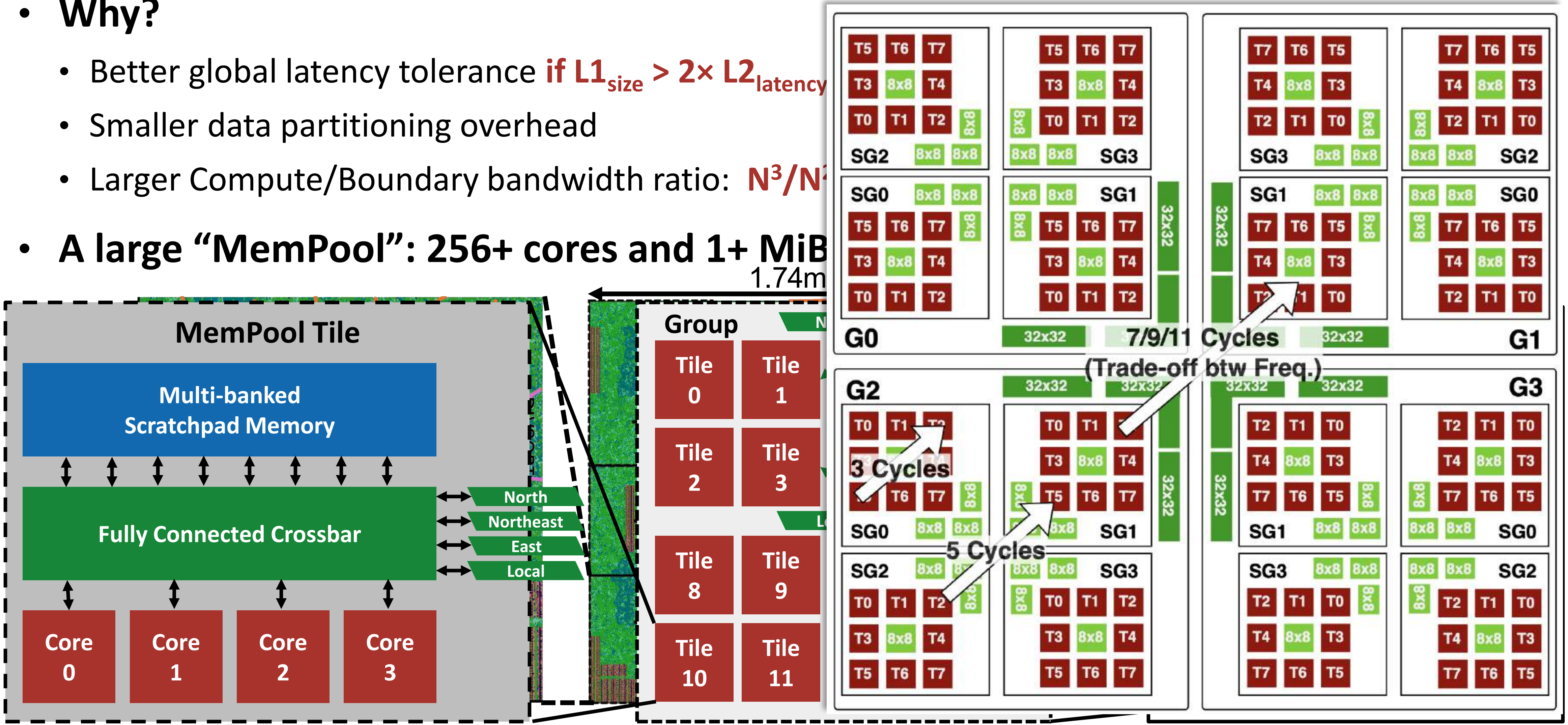
1.30x MHA perf speedup

BestArch vs Nvidia H100

Total Runtime(ms)	
Baseline: Flash Attention for Each Head on Each Cluster	14.4
Flatten Attention (w/o NoC collective)	17.7
Flatten Attention (w/ NoC collective)	4.6

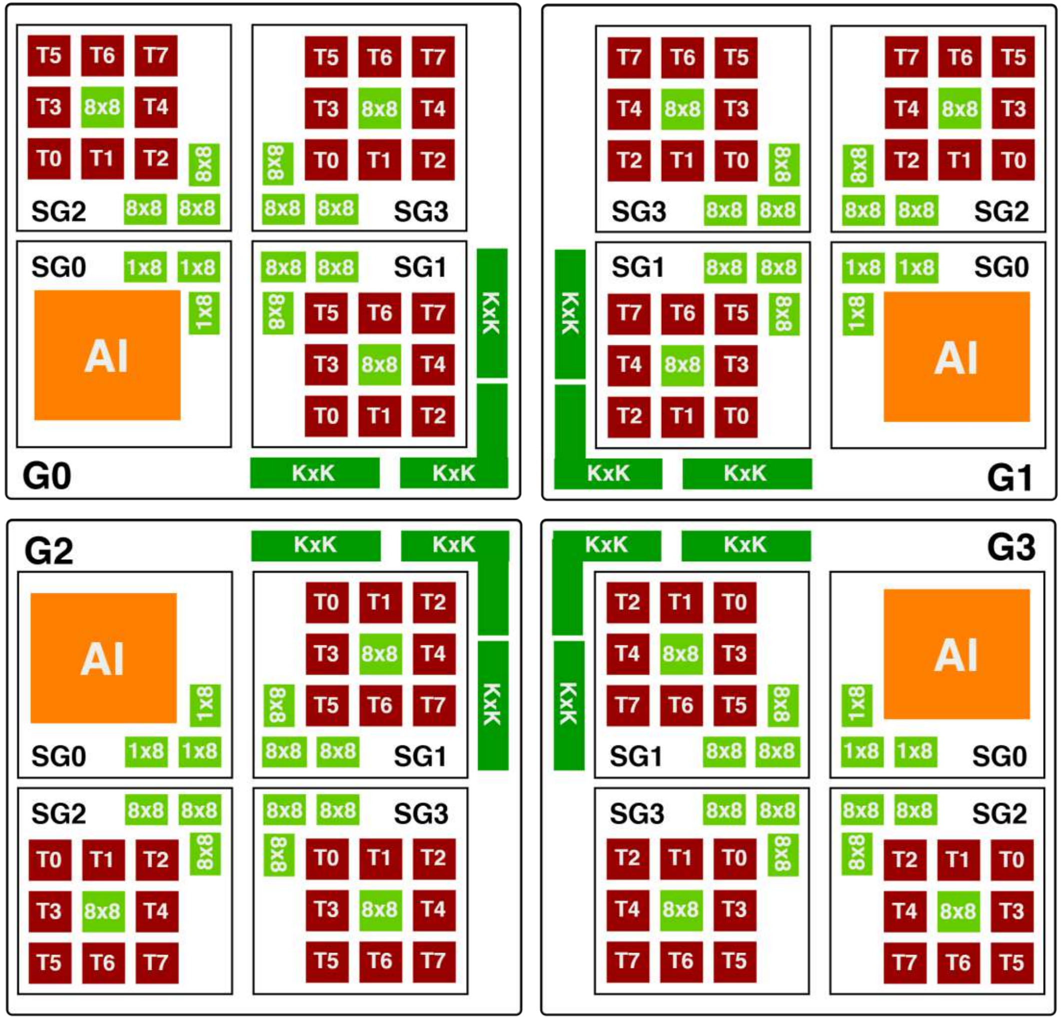
Scale UP: Matmul Benefits from Large Shared-L1 clusters

- **Why?**
  - Better global latency tolerance **if $L1_{size} > 2\times L2_{latency}$**
  - Smaller data partitioning overhead
  - Larger Compute/Boundary bandwidth ratio: **N^3/N^2**
- **A large "MemPool": 256+ cores and 1+ MiB**



MemPool + Integer Transformer Accelerator (ITA)

- **Tightly coupled Acceleration Engine**
  - Matmul & Softmax
  - Reduce pressure on memory and interconnect
    - Collaborative Execution
  - Cores prepare activations for the next attention head
  - Final head accumulation computed in cores
  - Nonlinearity in cores (PACE)
- **What is the correct mix of components**
  - How many accelerators
  - How many cores
  - Active area of research

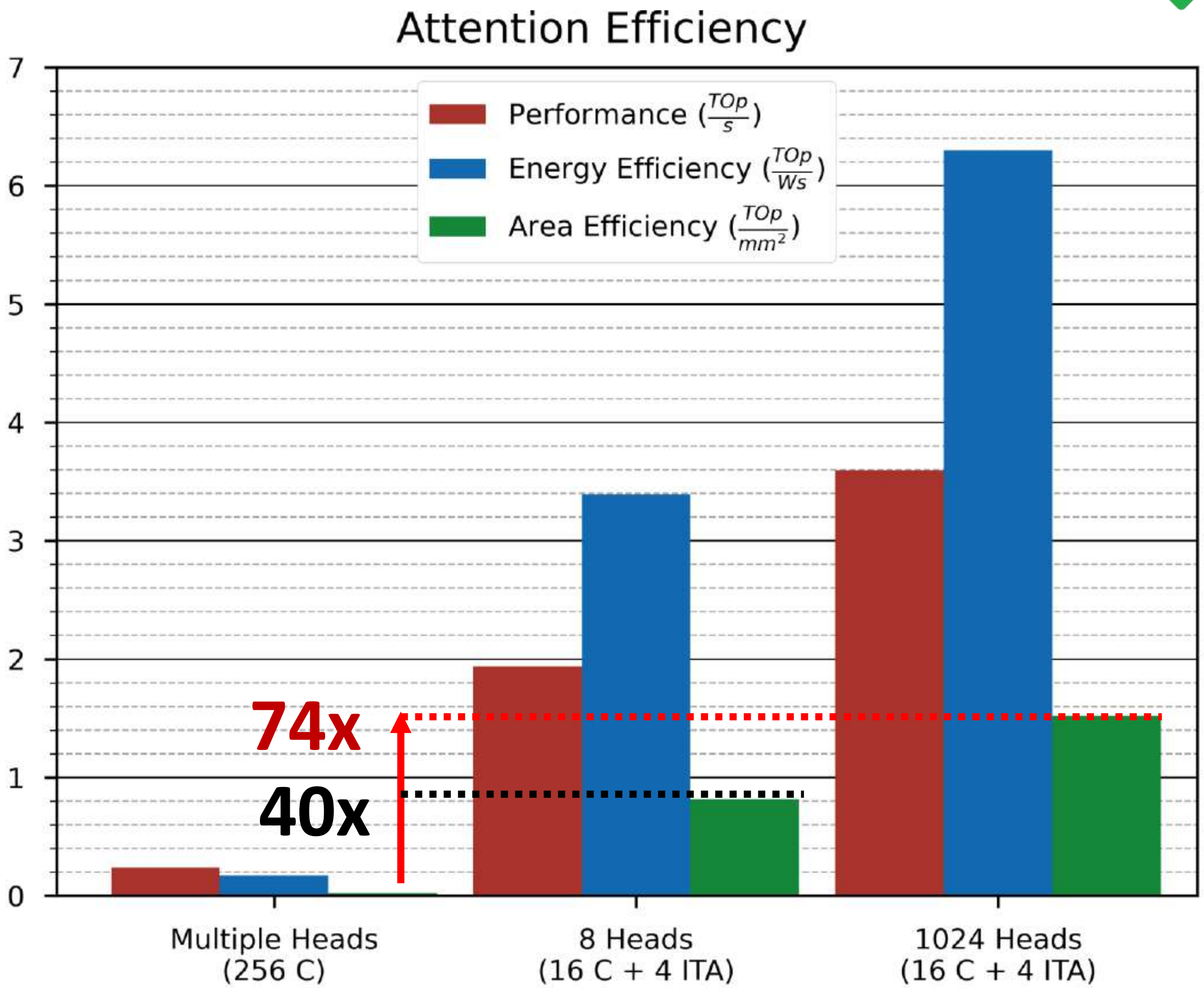


Attention on ITA

Performance
increase of **15x**

Energy Efficiency
increase of **36x**

Area Efficiency
increase of **74x**



Let’s talk about how Open HW and RISC-V in Space

Challenges of Space Applications

- Many interesting and challenging problems
- We need to find ways to **work together in Europe** to develop our own solutions

OpenHW and RISC-V hold the key to boost innovation

- It is not so much the ISA, but the **freedom to change/adapt and evaluate**
- Having access to **silicon proven** SoC templates and IP is crucial

PULP is working on reliable and high-performance open architectures

- Suitable for space applications (and beyond)
- Silicon proven, open for use, **open for collaboration**

http://pulp-platform.org @pulp_platform

Looking forward to more space missions

https://open-source-chips.eu/

Our WWW page contains a wide collection of talks/papers

PULP Platform Resources ▾ Projects ▾ About ▾ FAQ Privacy Policy Contact

PULP Platform

Open hardware, the way it should be!

Latency 6G-SDR won the Best PhD forum Award at VLSI-SoC 2024 in Tangier.

28 August 2024
Luca Benini received the 2024 TCMM Open Source Hardware Contribution Award for his work on the PULP platform. The award was presented at Hot Chips 2024

Most of our talks: https://pulp-platform.org/conferences.html

And most of our papers: https://pulp-platform.org/publications.html