

Designing Reliable & Efficient Integrated Systems for Embodied AI

An Open Platform Approach

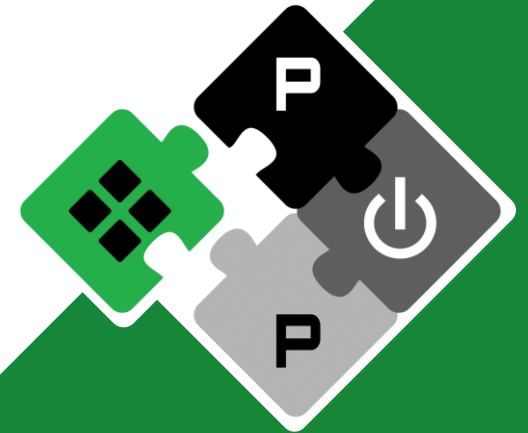
Luca Benini

lbenini@iis.ee.ethz.ch

luca.benini@unibo.it

PULP Platform

Open Source Hardware, the way it should be!



@pulp_platform 

pulp-platform.org 

youtube.com/pulp_platform 

Physical AI: Artificial Intelligence Everywhere



Humanoid



[Unitree26]

Interceptor Drone



[GeneralCherry26]

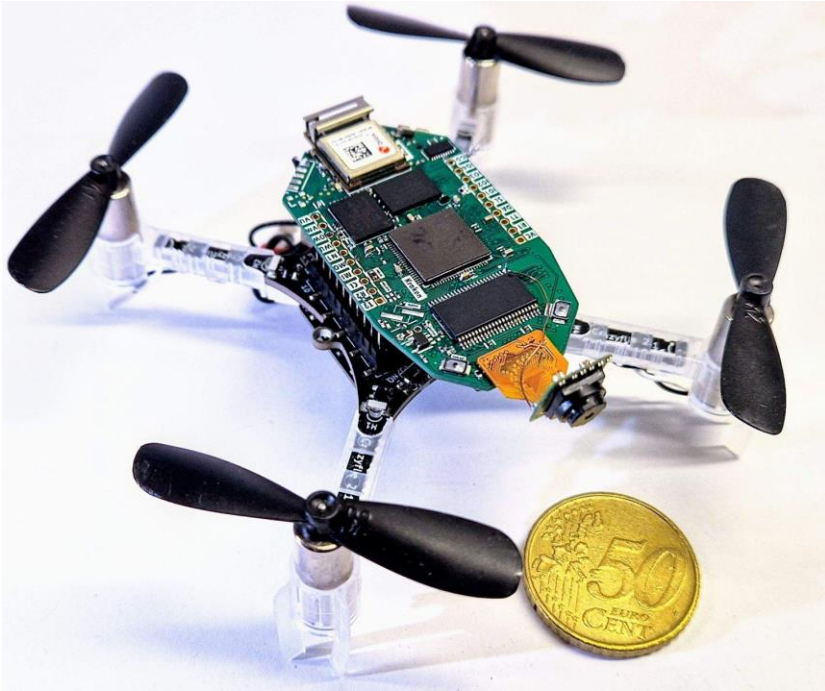
**Humanoid,
Interceptor Drone**
 $P_{\text{avg}} < 200 \text{ W}^*$

*[research.mobiusriskgroup.com/p/the-energy-diet-of-humanoid-robots]

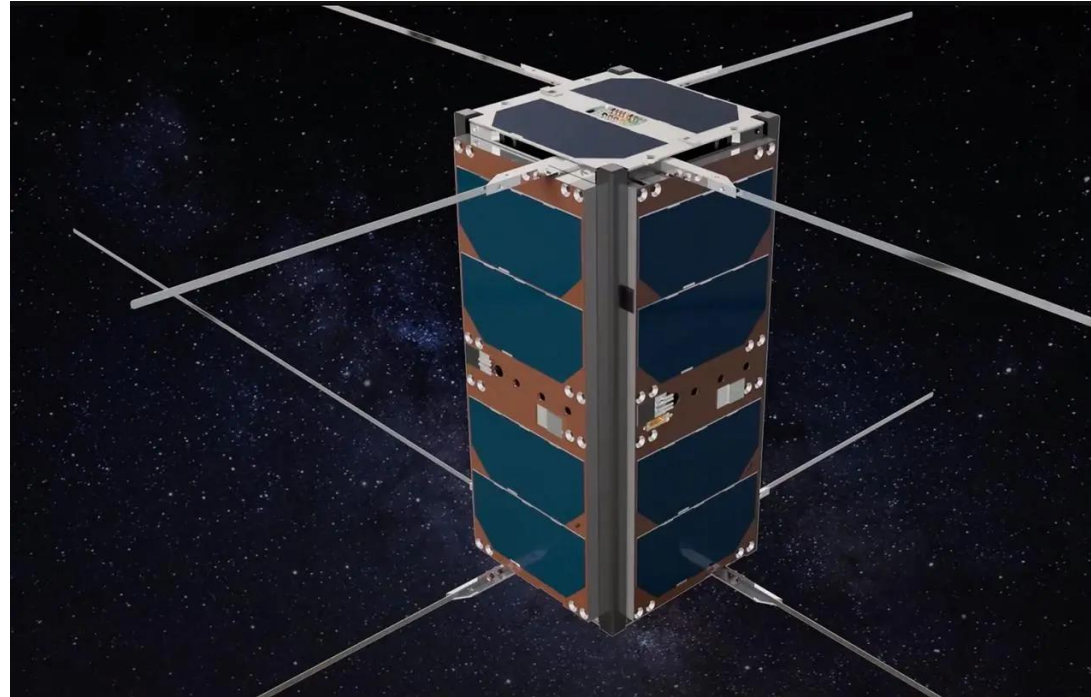
Physical AI: Artificial Intelligence Everywhere



Nano-Drone



Cubesat

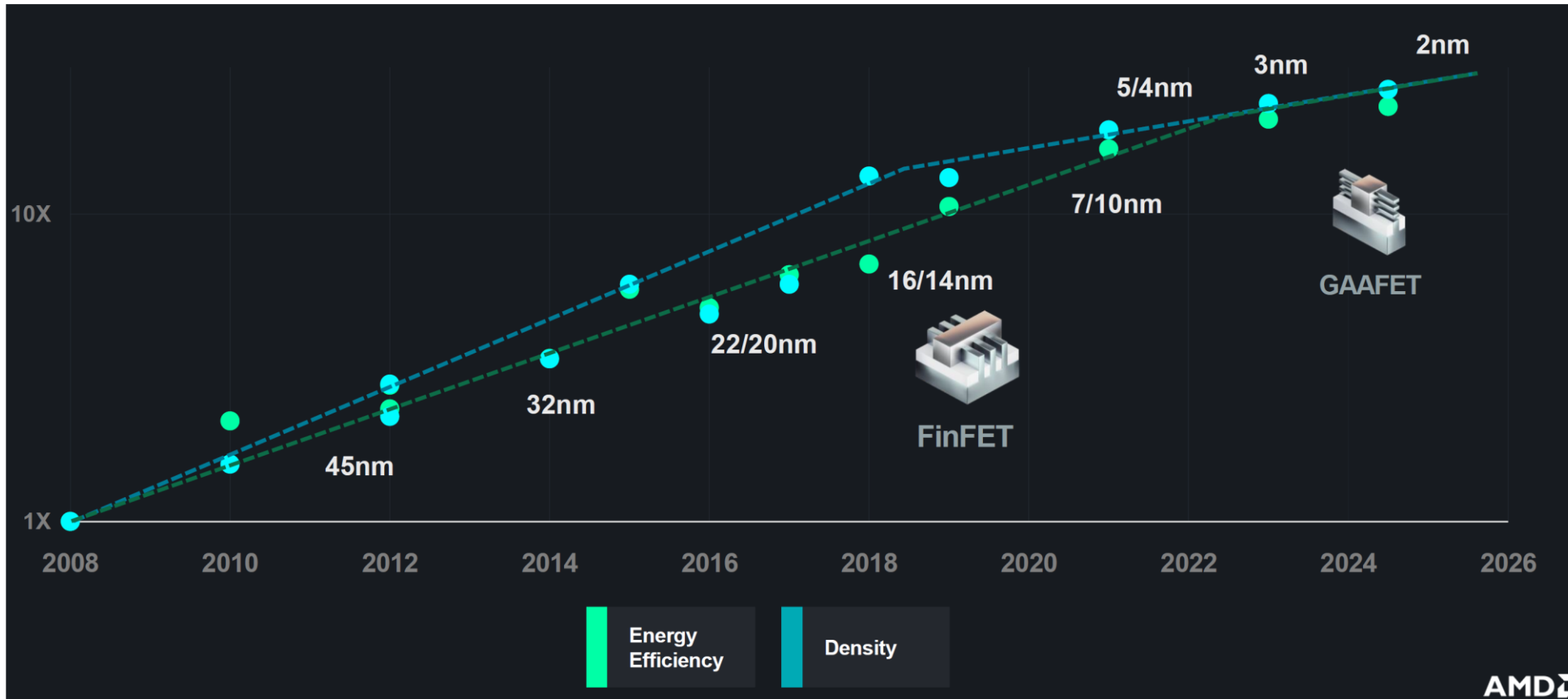


Cubesat
 $P_{\text{avg}} < 1.5 \text{ W}$

Nanodrone
 $P_{\text{avg}} < 150 \text{ mW}$

Efficient and Reliable operation key for Phy.AI!

Efficiency Challenge



Model complexity
10x every **~2.5** years

Moore's Law
10x every **12** years!



[AMD HotChips24]

Algorithm, Architecture, Design

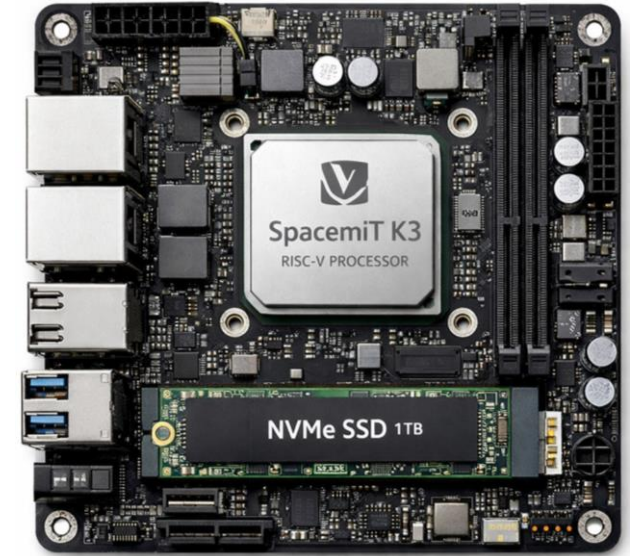
Innovation beyond “NVIDIA Gravity” is Challenging!

It's the software → **flexibility** key for fast evolution!

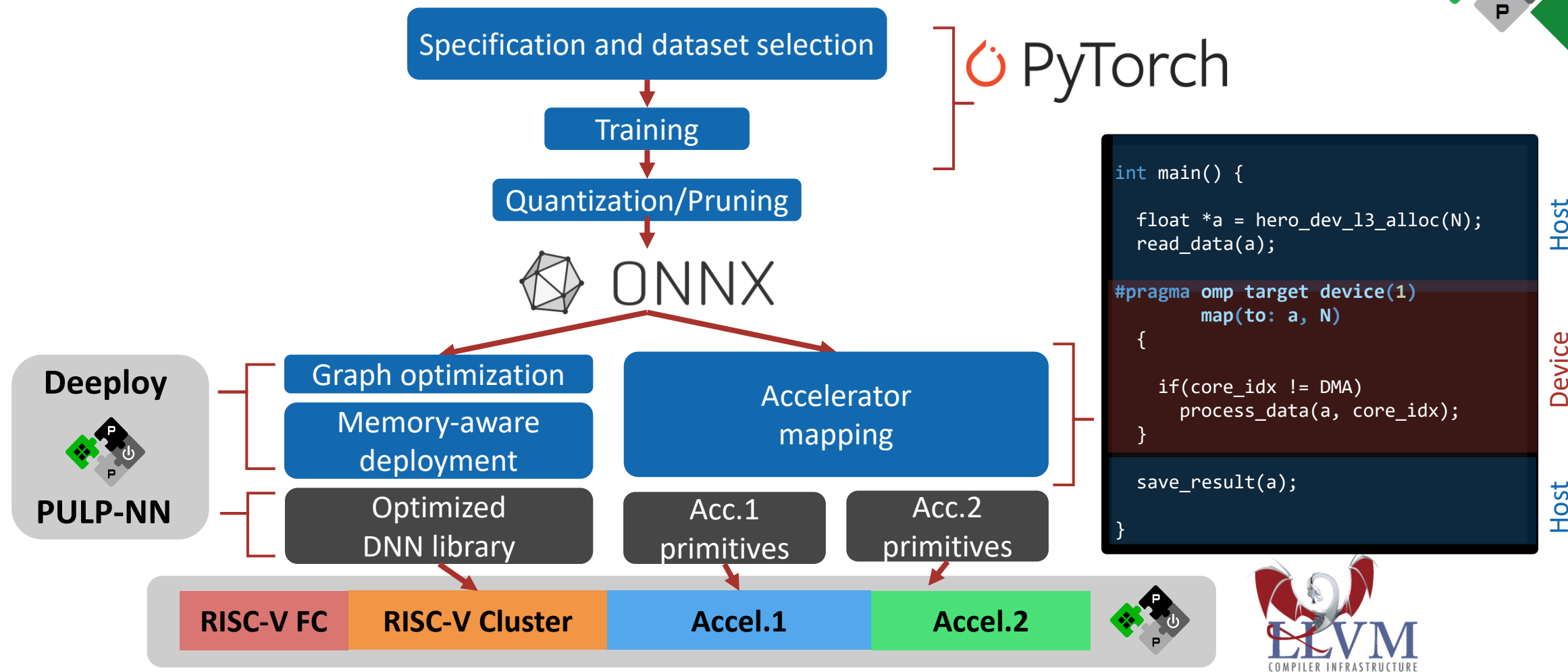
Need an **open standard** to counter a monopoly



RISC-V: The Free and Open RISC
Instruction Set Architecture



Let's not Forget Software....



Open HW-SW platforms & standards are key for open innovation vs “AI Monopoly”

Efficiency via Heterogeneity: Multi-Domain Specialization

Brain-inspired: Multiple areas, different structure different function!



1 Higher Mental Functions

- Concentration
- Planning
- Judgment
- Emotional expression
- Creativity
- Inhibition - Ability to control self

2 Motor Function Area

- Eye movement and placement of eyes

3 Broca's Area

- Ability to talk
- Ability to write

4 Motor Function Area

- Ability to move muscles

5 Association Area

- Short-term memory
- Emotion

6 Sensory Area

- Touching and feeling

7 Auditory Area

- Hearing

8 Wernicke's Area

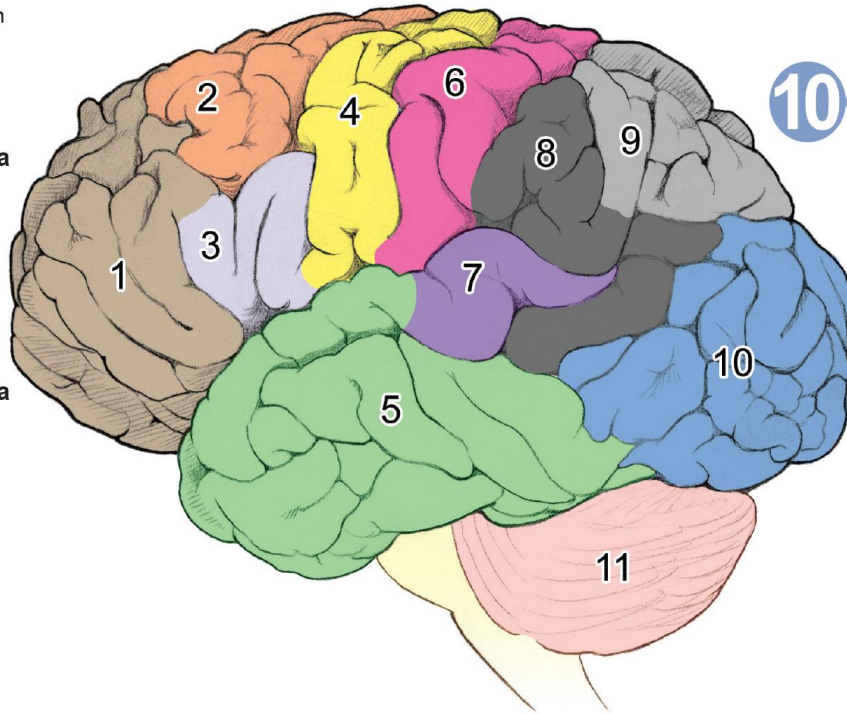
- Written and spoken language understanding

9 Somatosensory Association Area

- Understanding of weight, texture, temperature, etc. for recognizing and comprehending an object

10 Visual Areas

- Sight
- Ability to recognize pictures
- Awareness of size and shape



FUNCTIONAL AREAS OF THE CEREBELLUM

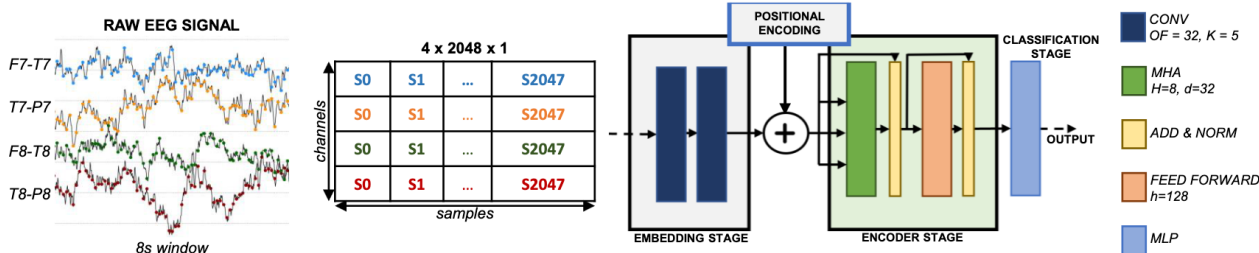
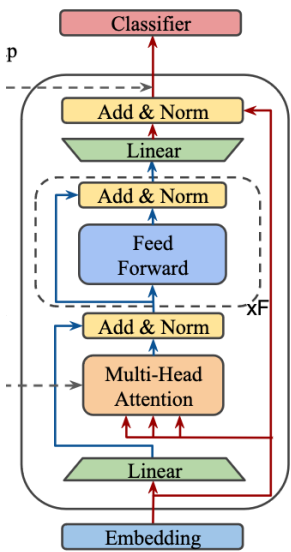
11 Motor Functions

- Coordination of movement
- Balance
- Posture

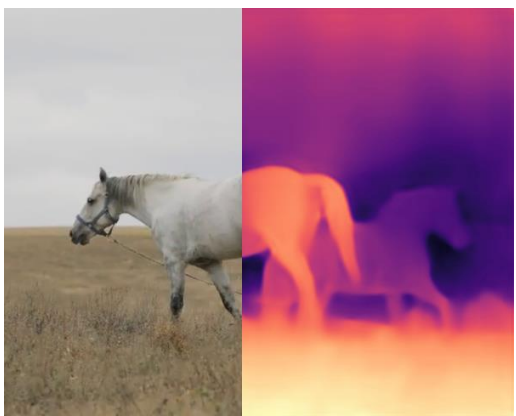
A Fast-Evolving Model Zoo



[Z. Sun et al.] MobileBERT
Encoder Transformer



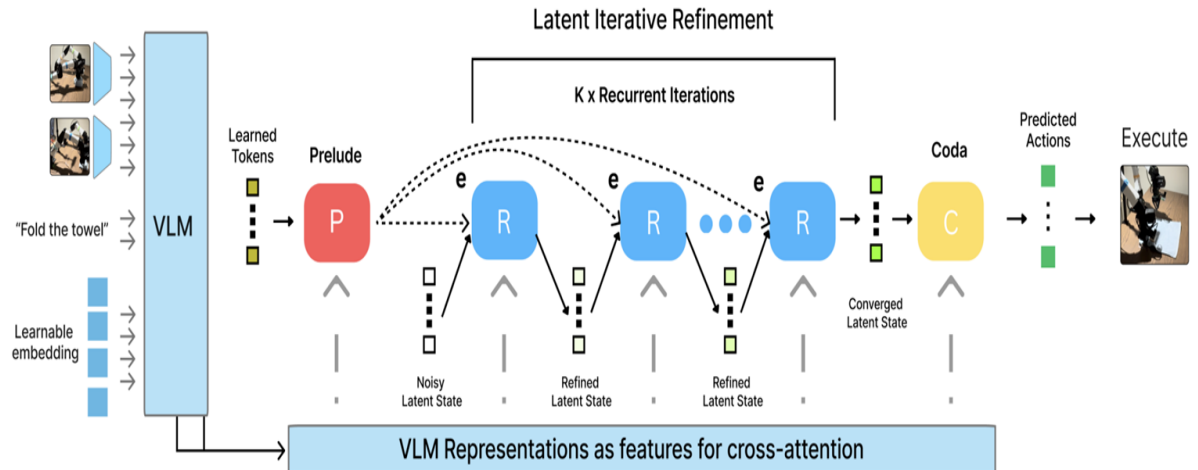
[P. Busia et al.] EEGFormer
Encoder Transformer



DINOv2: Learning Robust Visual Features without Supervision
[M. Oquab et al.]
Encoder



Auto-regressive



Recurrent

Domain-Specific ISA Extensions



Instruction set: open and extensible by construction (great!)

8-bit Convolution

Vanilla

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
```

RISC-V
core

Specialized for AI → Mixed precision SIMD (16-2bit)

N/4

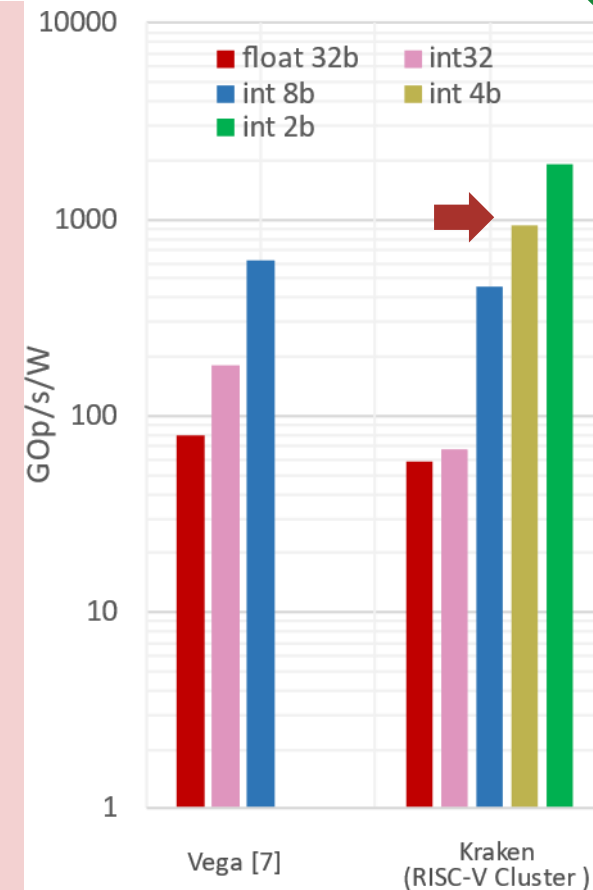
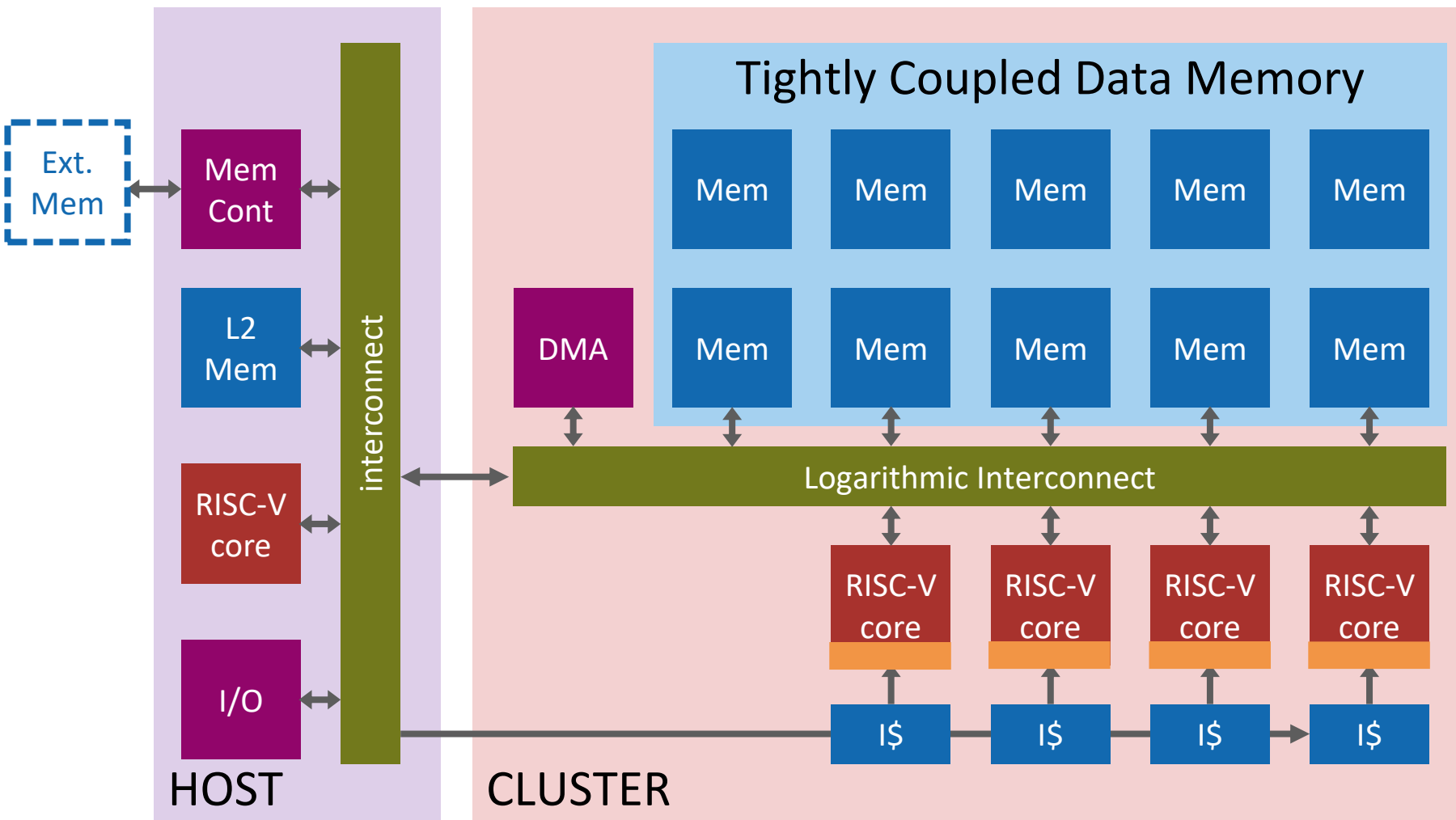
```
Init NN-RF (outside of the loop)
lp.setup
pv.nnsdotup.h s0,ax1,9
pv.nnsdotsp.b s1,aw2,0
pv.nnsdotsp.b s2,aw4,2
pv.nnsdotsp.b s3,aw3,4
pv.nnsdotsp.b s4,ax1,14
end
```

RISC-V
core

15x less instructions than Vanilla
90%+ ALU Utilization

Specialization Cost: Power, Area: 1.5x↑ Time 15x↓ → E = PT 10x ↓

A Cluster of Domain Specific Cores



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

Heterogeneous, Multiscale Domain-Specific Architecture



Multiple Scales of Specialization

Extensions to processor cores

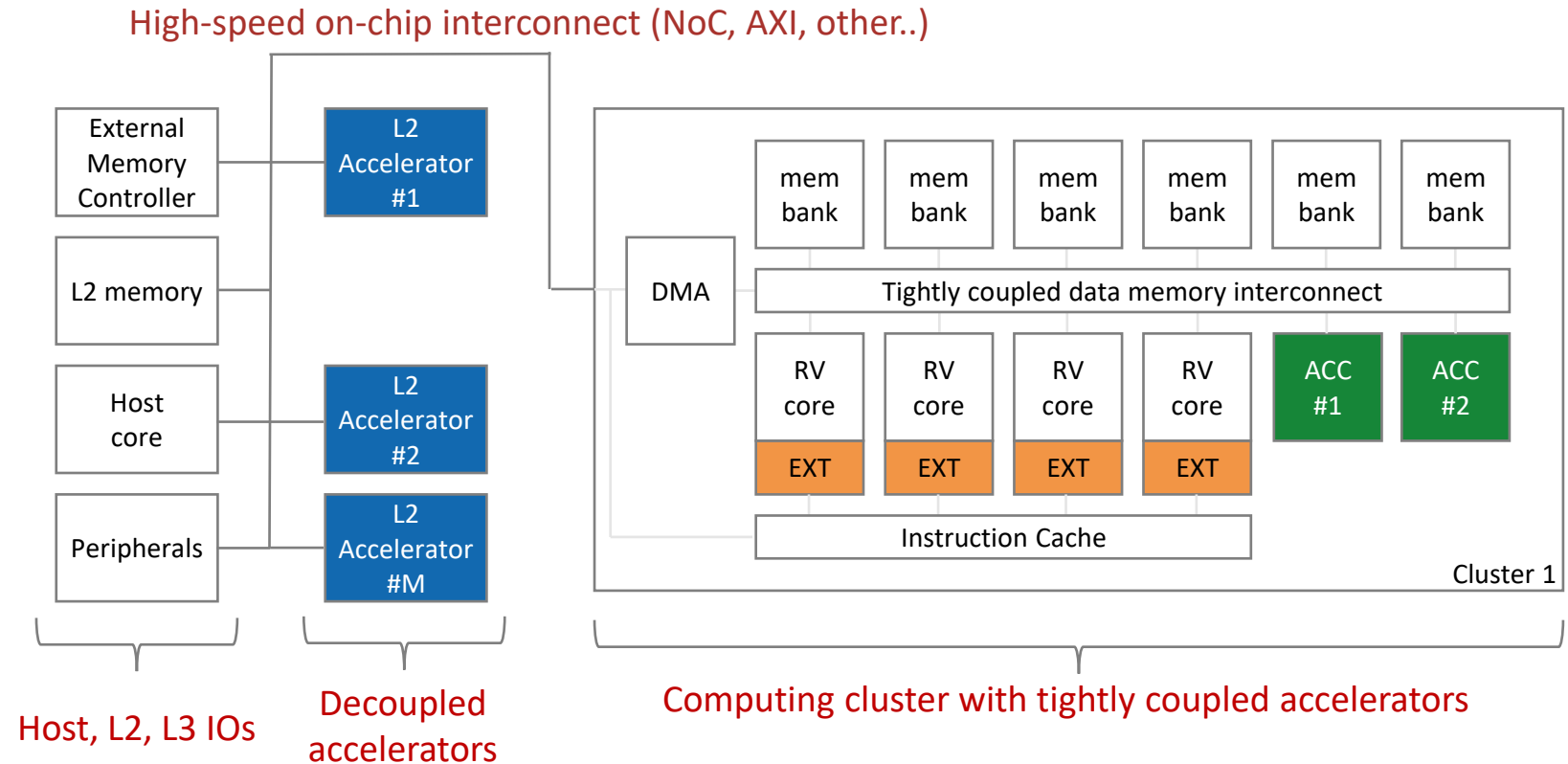
- Tightly-coupled, fine sychro
- Shared L0

Tightly coupled Accelerators

- Fast offload, coarse sychro
- Shared L1

Decoupled Accelerators

- Decoupled
- Shared L2 (or higher)

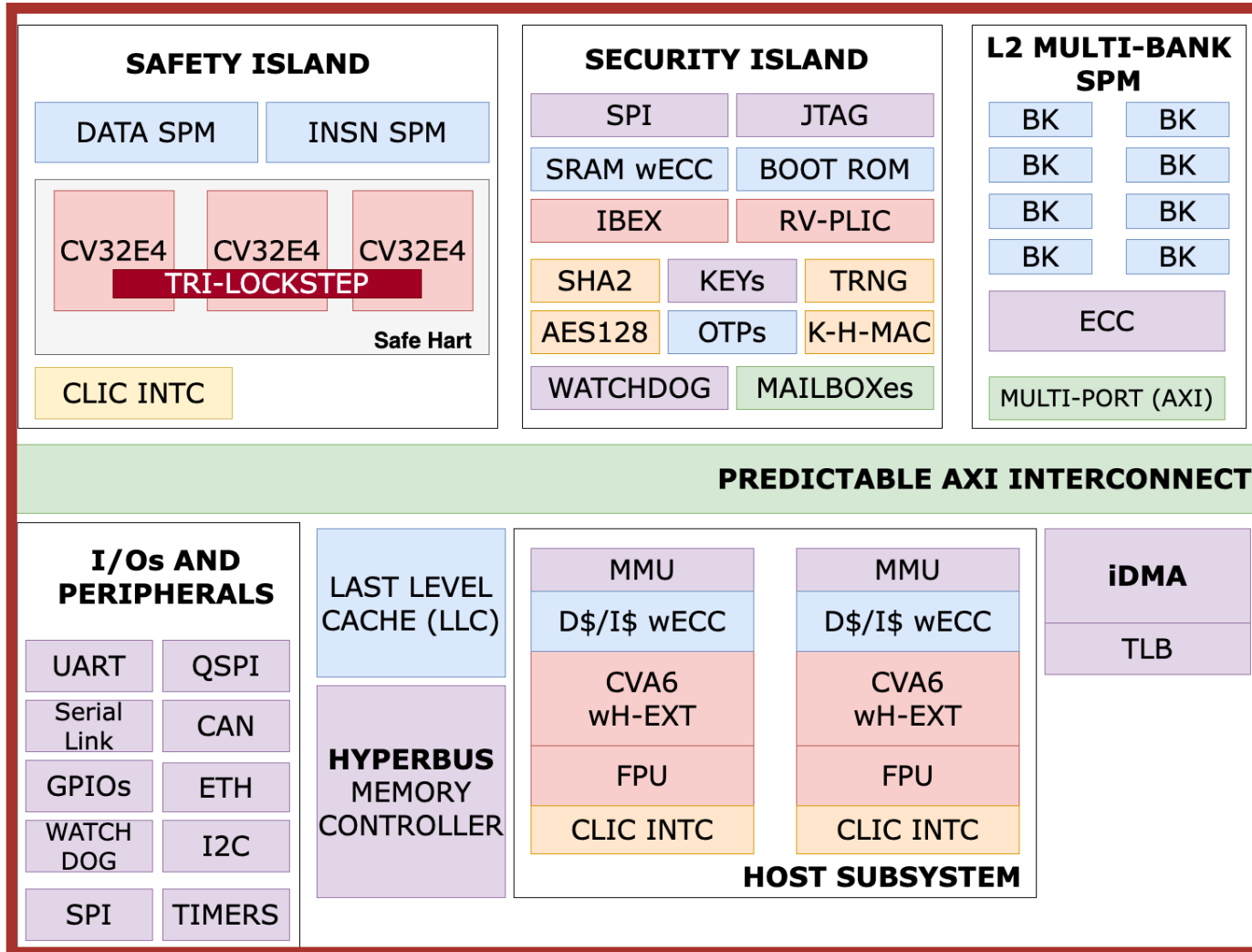


RISC-V is a key enabler → max agility, enabling SW build-up, without vendor lock-in

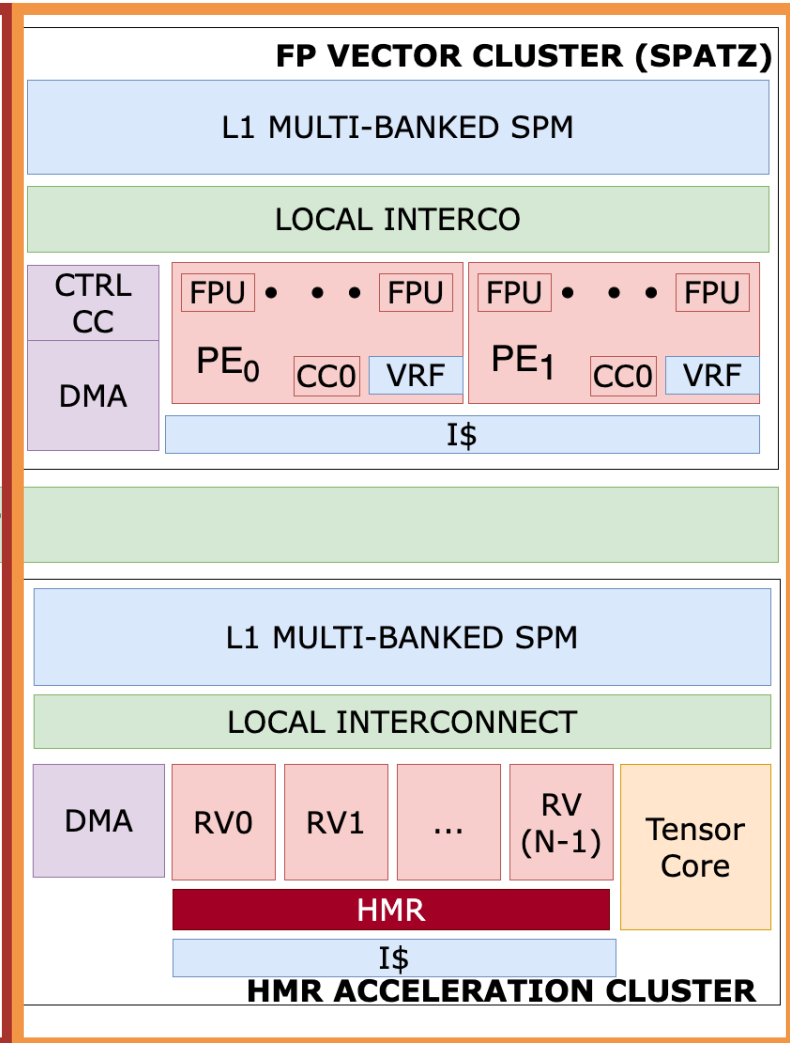
Carfield: H-DSA SoC for Autonomous Vehicles



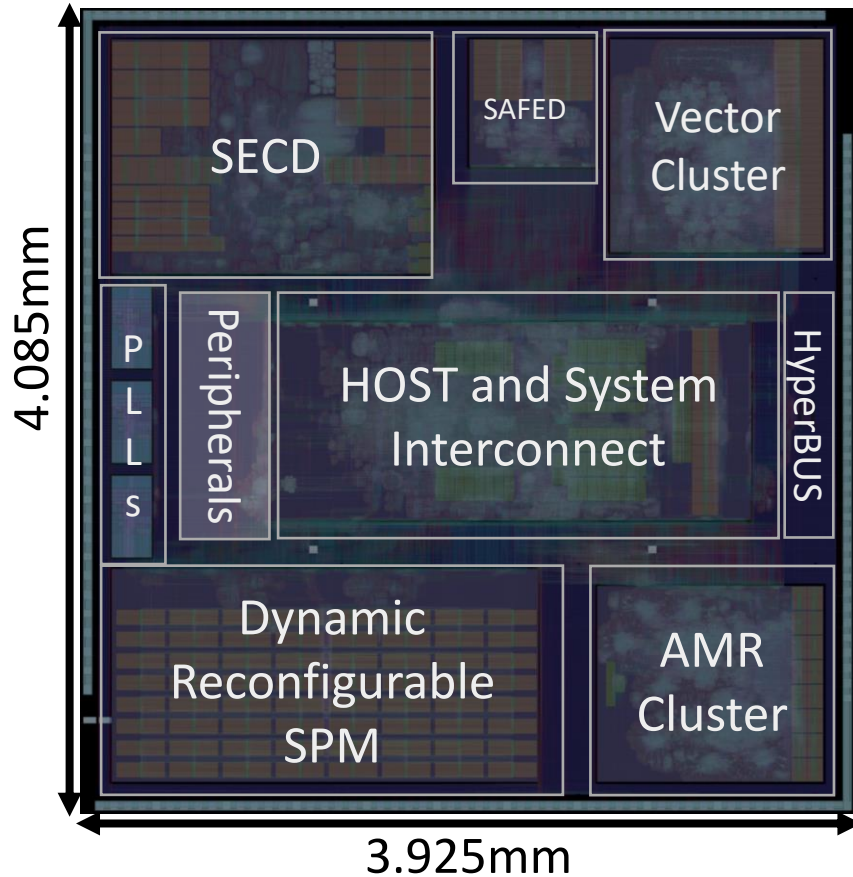
Main Computing and I/O System



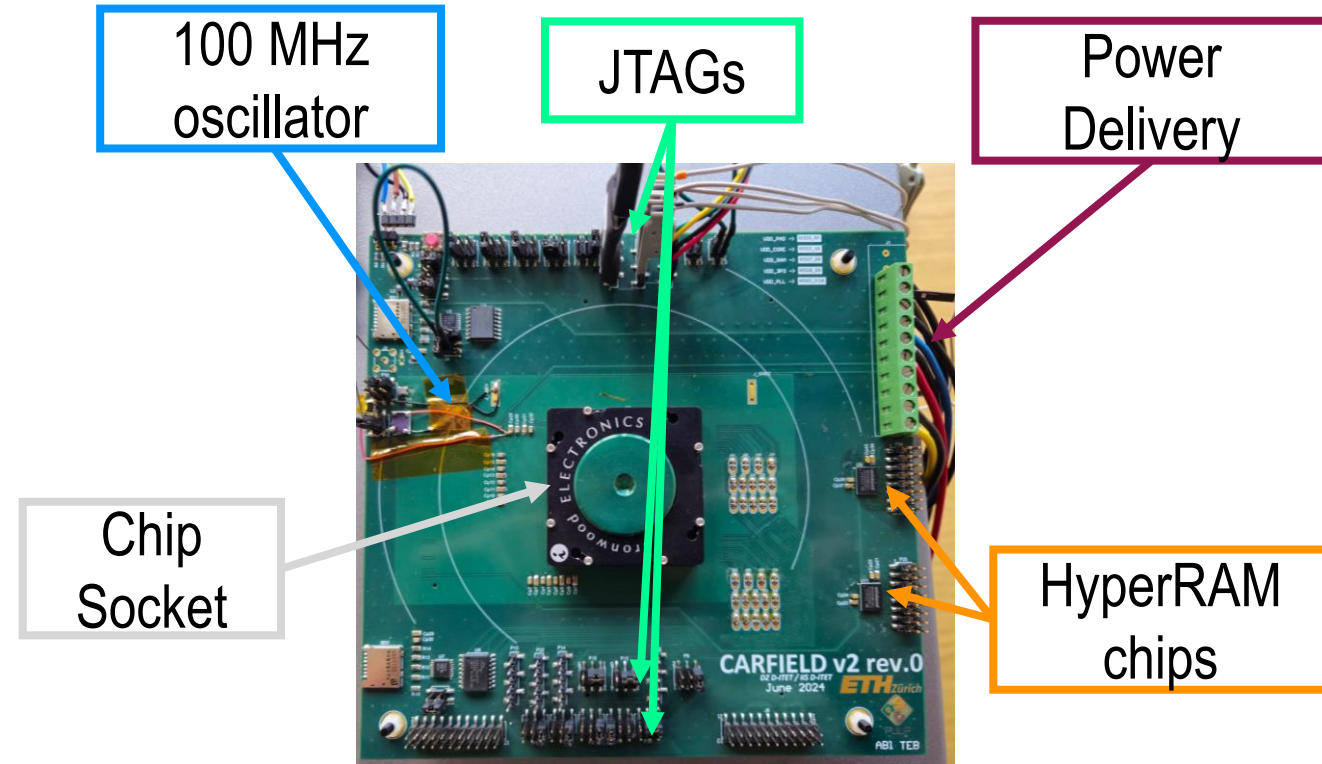
Accelerators Domain



Carfield: SoC in Intel16 FinFet



Up to 1 GHz, **1.2W** @ 0.8V



A. Garofalo et al., "A Reliable, Time-Predictable Heterogeneous SoC for AI-Enhanced Mixed-Criticality Edge Applications," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 72, no. 11, pp. 1625-1629, Nov. 2025

Specialization in Perspective



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

Energy-Efficient RV Core $\rightarrow$ **20pJ (8bit)**



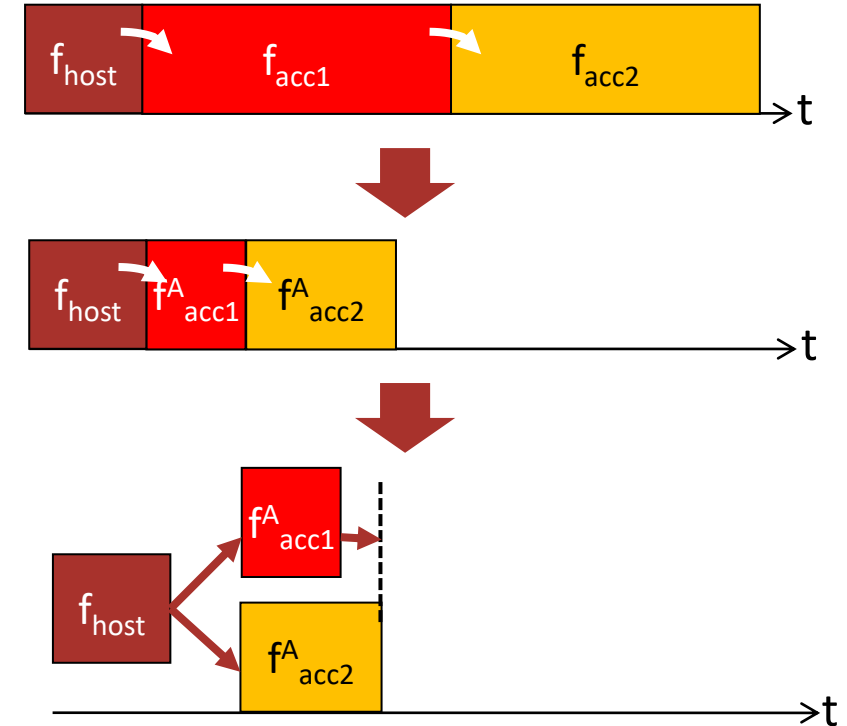
ISA-based 10-20x $\rightarrow$ **1pJ (4bit)**



Tightly coupled 10-20x $\rightarrow$ **100fJ (4bit)**



Decoupled 100x $\rightarrow$ **1fJ (ternary)**

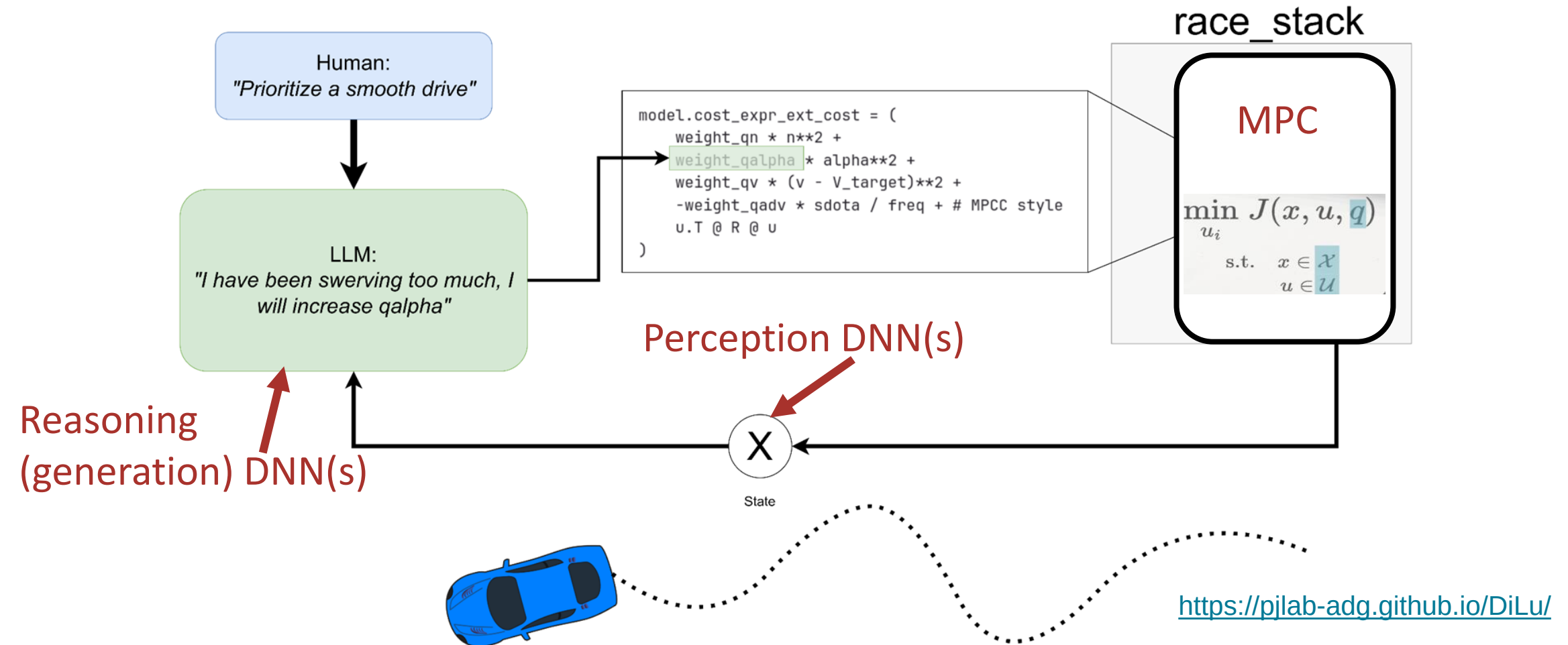


Specialization game: efficiency vs. utilization Tradeoff with Amdahl's Law

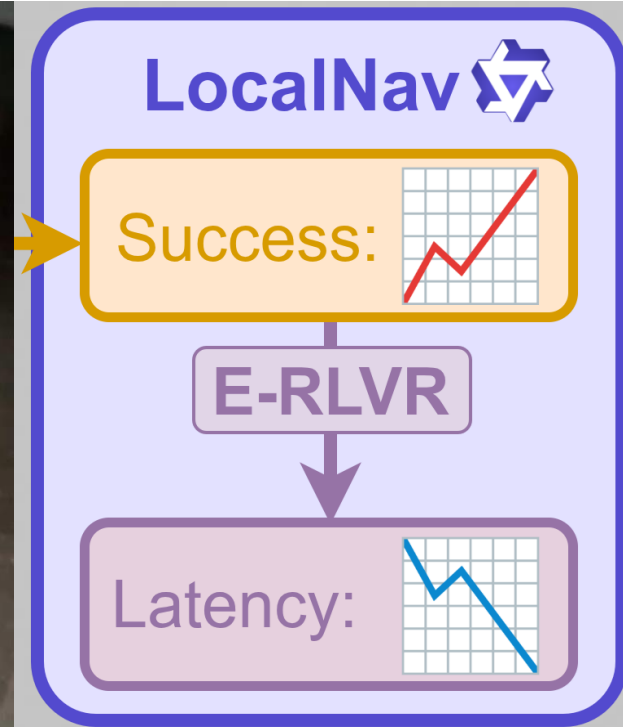
Beyond Perception: Agentic Reasoning



LLM Reasoning on Human Commands & Robot Observations



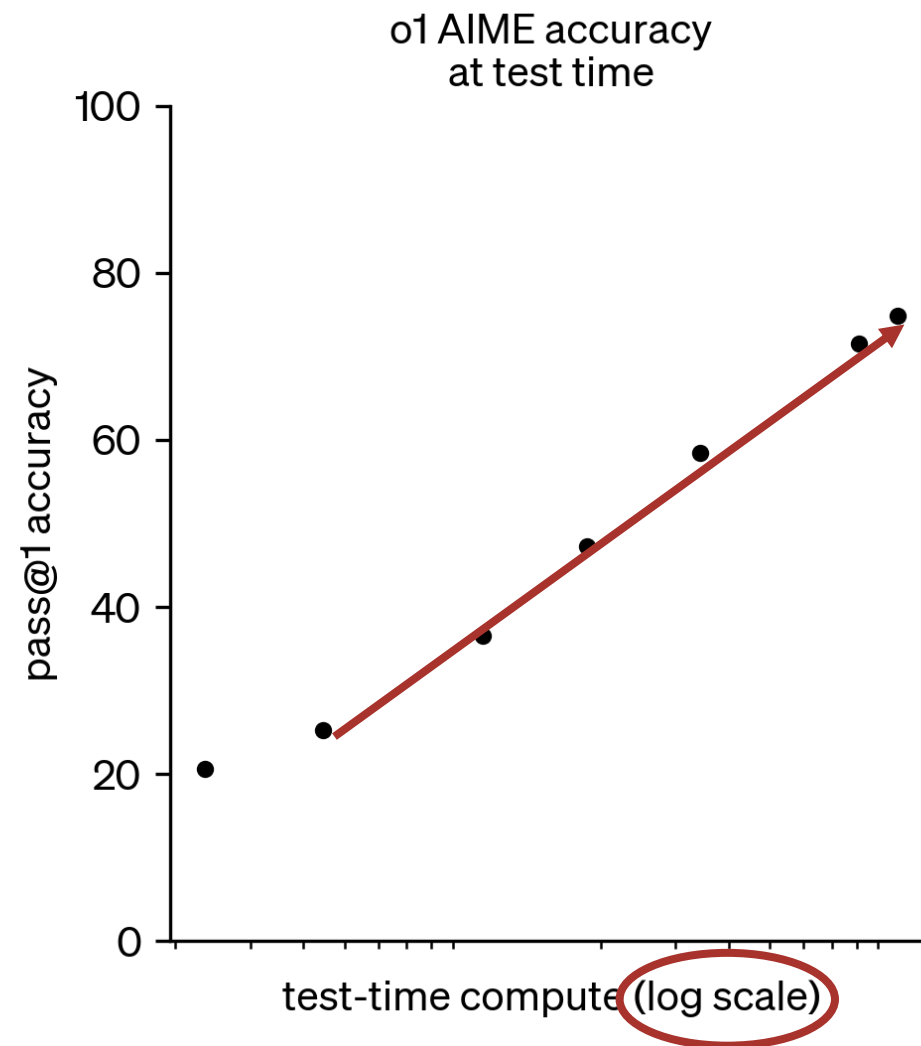
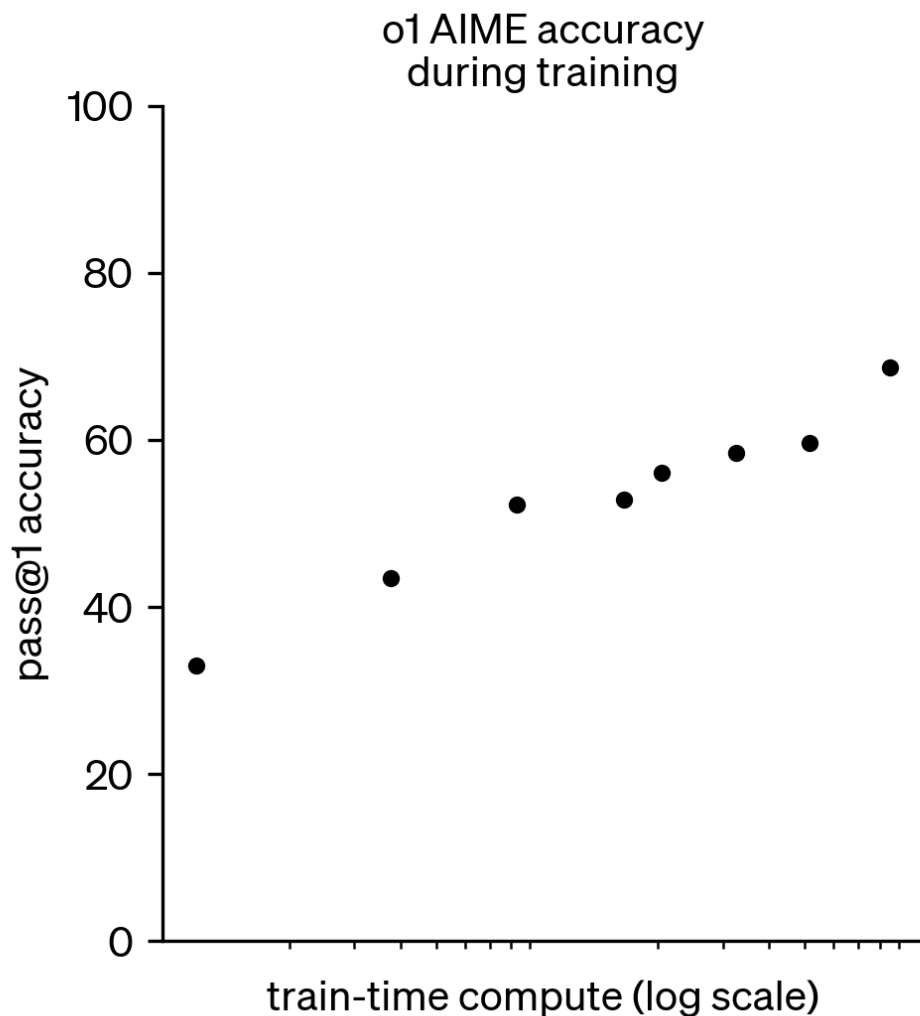
Beyond Perception and Fusion: Agentic Reasoning



12.1% loss output tokens

1.8% loss latency

Generative AI Scaling Laws



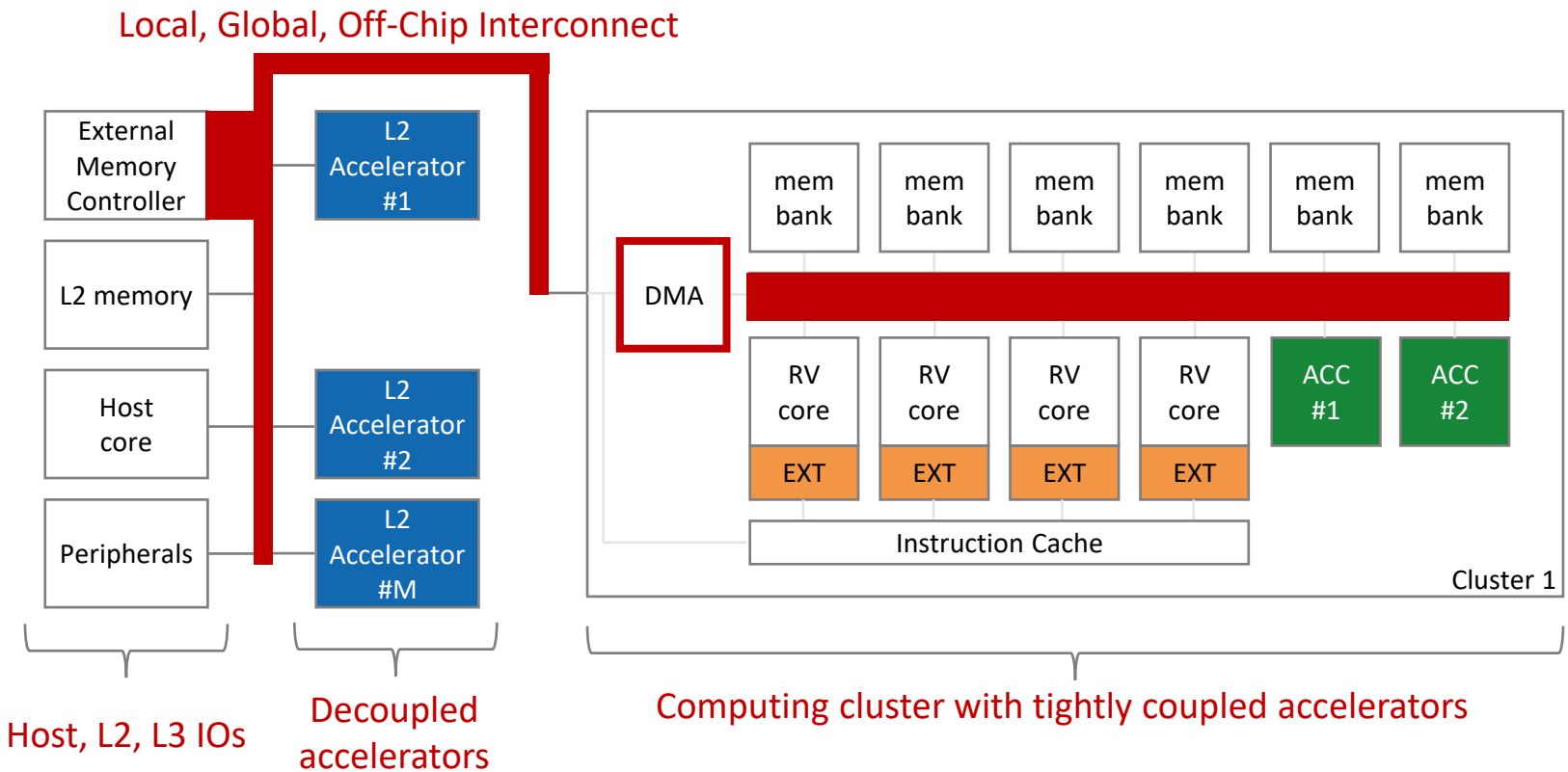
[openai.com/index/learning-to-reason-with-llms/]



There is no Other Way to Go, but UP

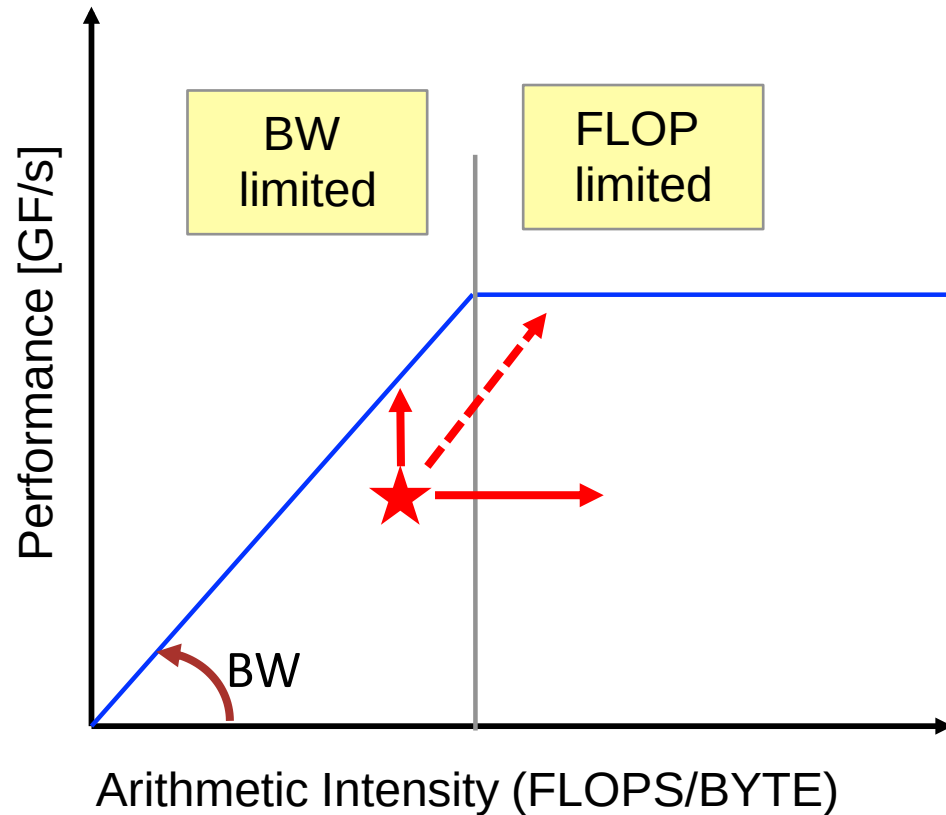
Multiple Scales of specialization

- Extensions to processor cores
 - Tightly-coupled, fine sychro
 - Shared L0
- Tightly Coupled Accelerators
 - Fast offload, coarse synchro
 - Shared L1
- Multiple Decoupled Accelerators
 - Decoupled
 - Shared L2 (or higher)



Specialize interconnects too! Latency (and Bandwith) Challenge!

Patterson's Law (aka Roofline), Little's Law



Mean jobs in system = arrival rate x mean response time



BufferSize (B) = Bandwidth (B/s) x Latency (s)

Eg. 1KB/ns x 100 (ns) → 100KB

Need IO-bandwidth, on-chip Buffering, and latency tolerant Architecture

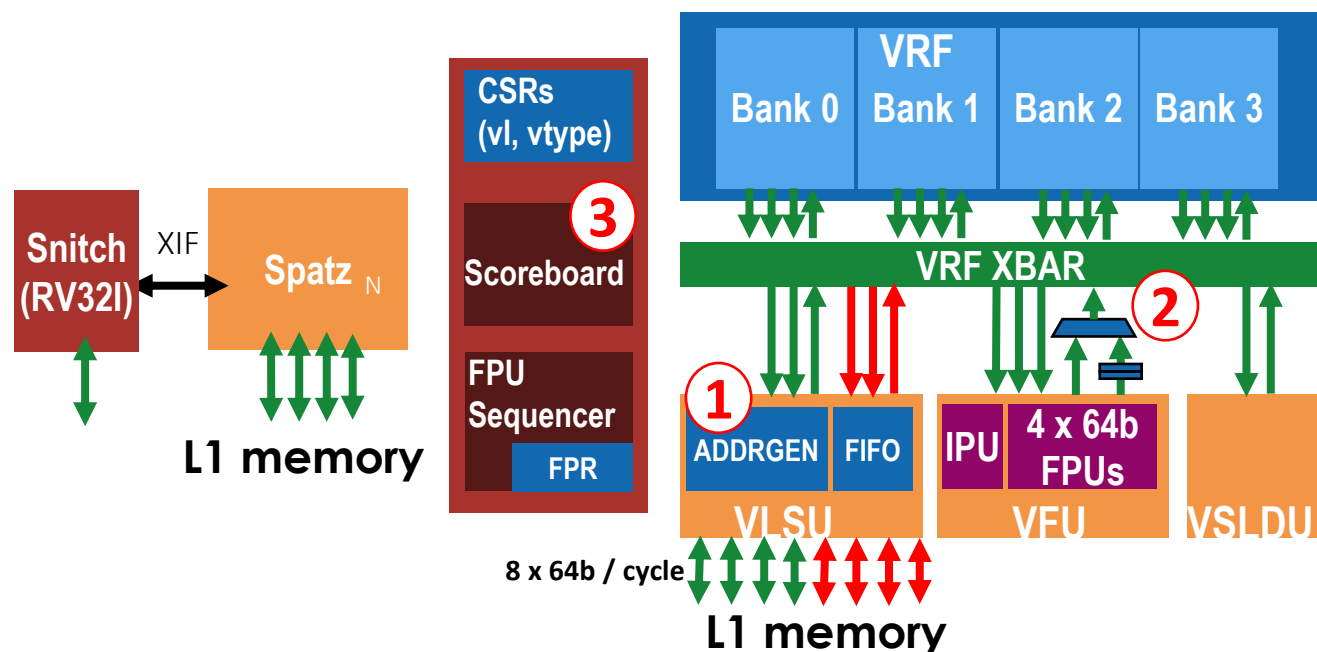


Spatz: RVV Unit for Snitch

Set of uA optimizations for VPEs

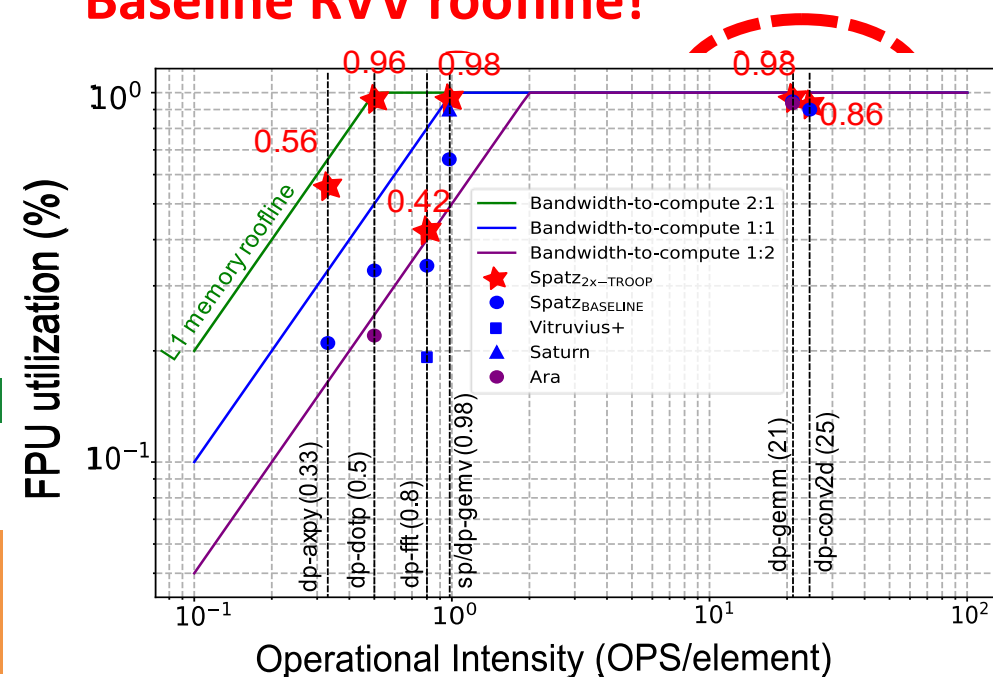
- Decoupled VLSU interfaces
- Dynamic Priority allocation and buffering
- Improved vector chaining

Fully utilize L1-memory BW + tolerate L1 latency (Troop)



Troop reaches L1-memory roofline!

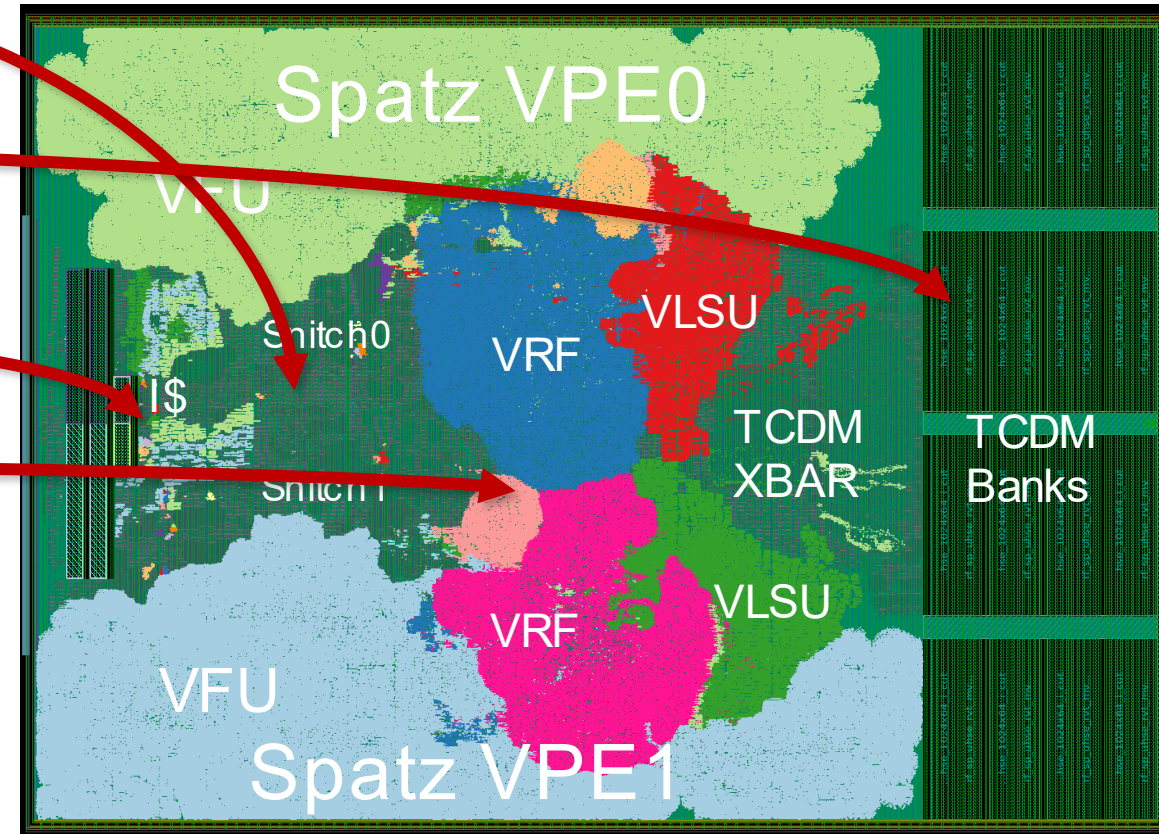
Baseline RVV roofline!



Snitch Cluster: The Fundamental Compute Block



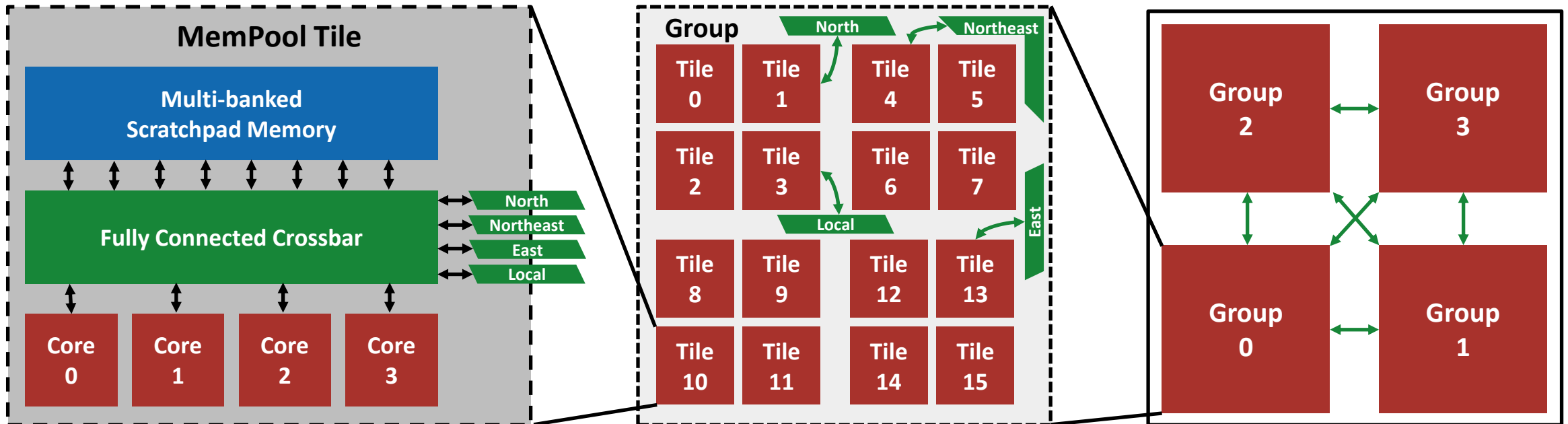
- **2 Snitch+Spatz compute cores**
 - 8 SIMD FPU
- **128 KiB Shared TCDM**
 - 32-bank, low-latency shared scratchpad
- **Shared I-cache and peripherals**
- **DMA engine**
 - 512b interface to interconnect
 - HW support for multi dim. transfers
- **Shared DMA (10% overhead) for latency tolerance of L2+→L1: 100s vs 10s cycles**



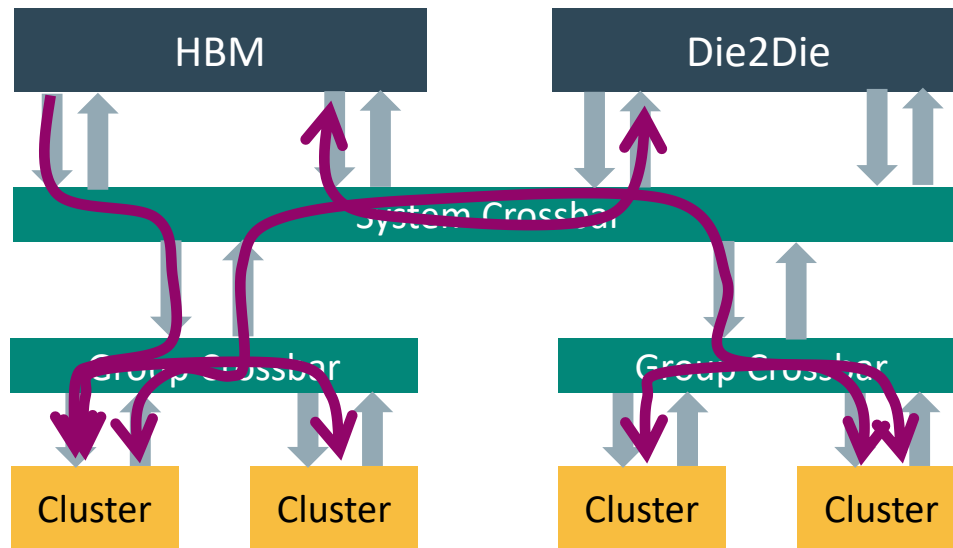
How Large should a Cluster be?



- The push for “Larger”
 - Better global latency tolerance if $L1_{size} > 2 \times L2_{latency} \times L2_{bandwidth}$ (Little’s law + double buffer)
 - Smaller data partitioning overhead
 - Larger Compute/Boundary bandwidth ratio: N^3/N^2 for MMUL grows linearly with N!
- A large “MemPool”: 256+ cores and 1+ MiB of shared L1 data memory



Scaling UP: Efficient and Flexible Data Movement



- **Problem:** HBM Accesses are critical in terms of

- Access energy
- Congestion
- High latency

- Instead reuse data on lower levels of the memory hierarchy

- Between **clusters**
- Across **groups**

- Smartly distribute workload

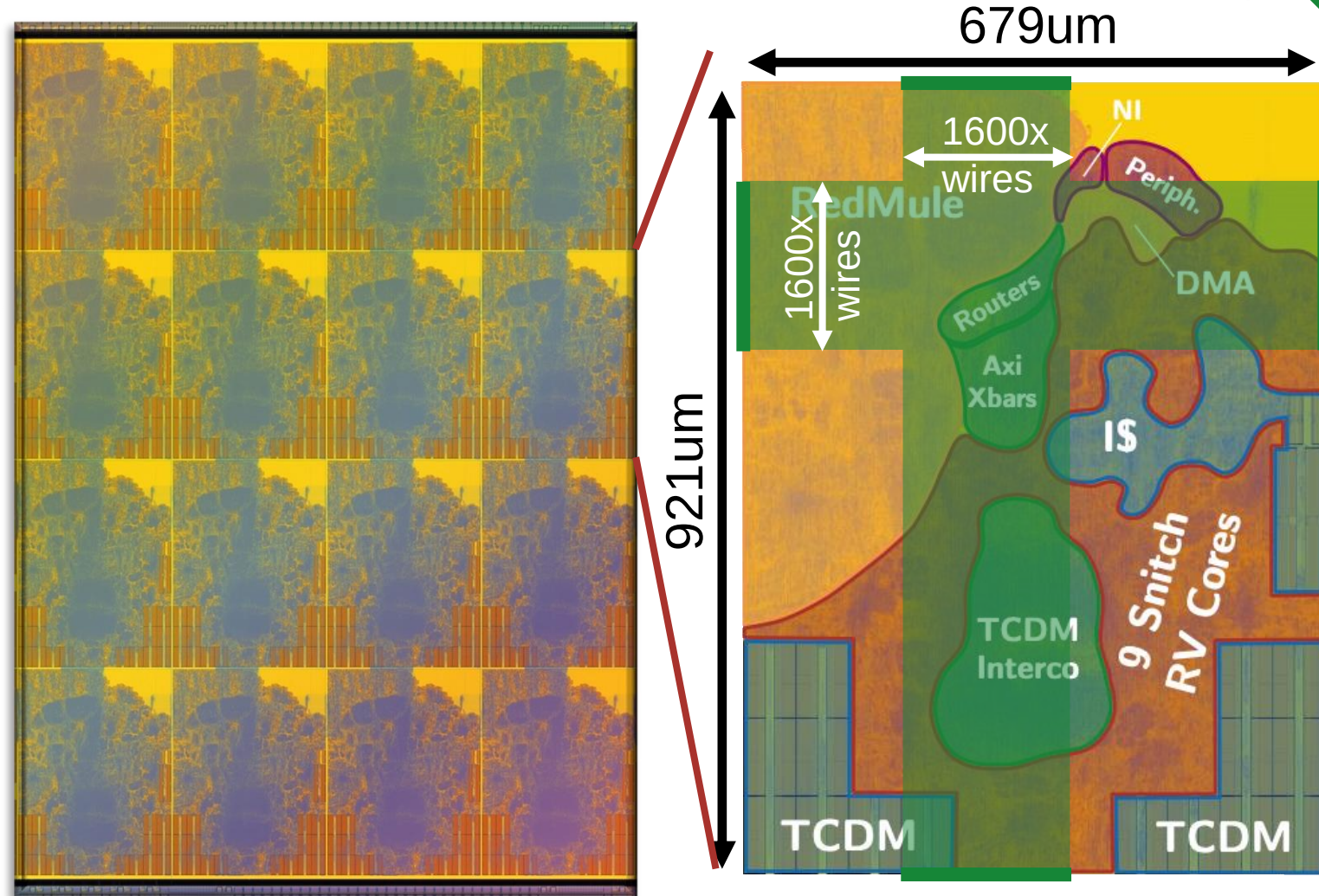
- **Clusters:** Tiling, Depth-First
- **Chiptlets:** E.g. Layer pipelining

Big trend!

Physically Scalable NoC: FlooNoC



- **Key Ideas in *FlooNoC***
 - Fully AXI4 compatible
 - Wide physical channels (multiplane vs. VC)
 - Over-the-Macro-Routing
 - Only 3% of std. cell area taken up by NoC
 - Used only ~10% of available routing resources
- **Big area/performance gains**
 - *High Bandwidth: 629Gbps/link*
 - *High Energy-Efficiency: 0.19pj/B/hop*



Here is Picobello: our 1st Gen Phy-AI Engine in TSMC 7nm

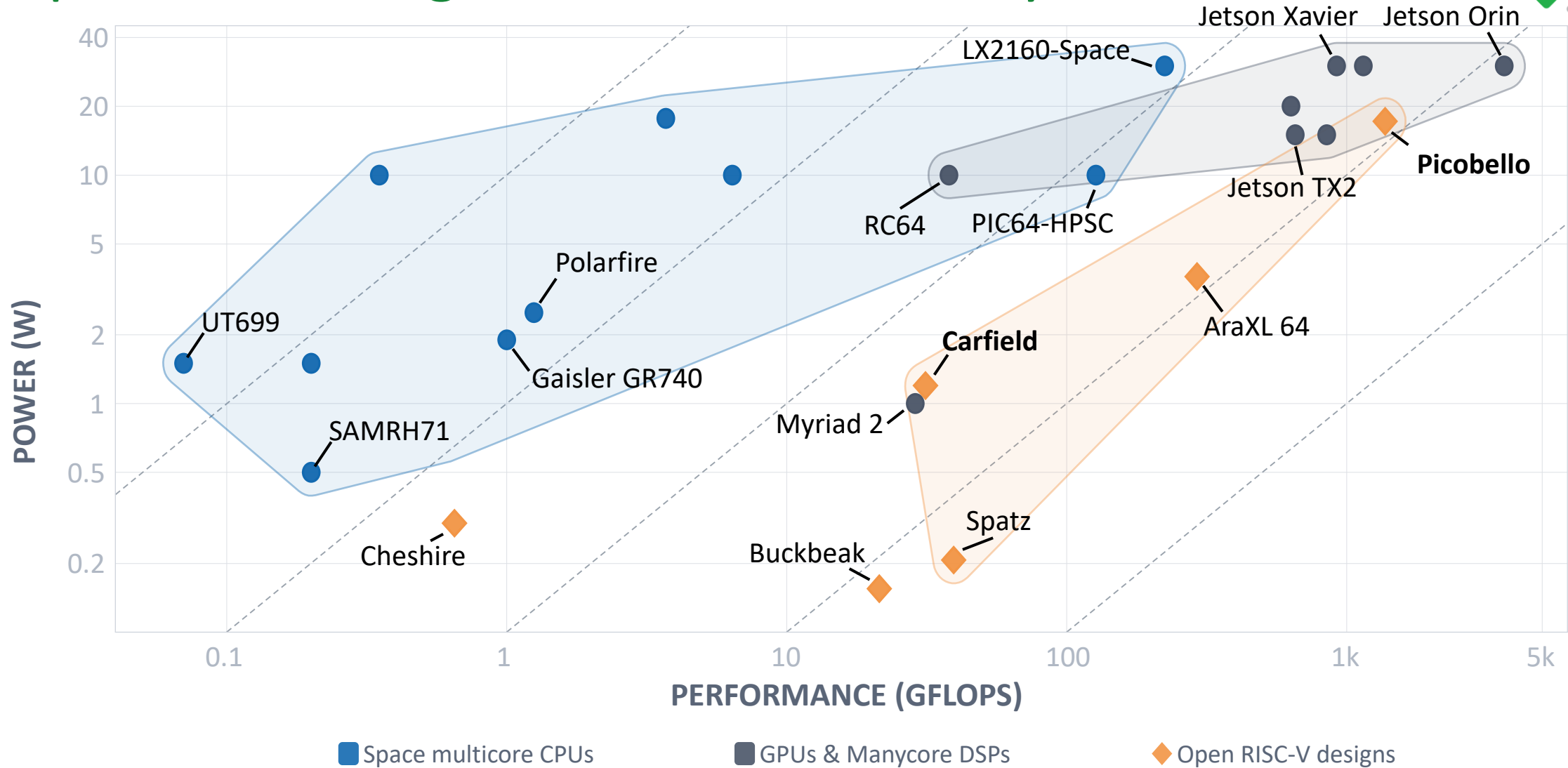


- 16 clusters totaling 144x RISC-V cores with FP8-FP64-bit support
- 8x 1MB of on-chip L2
- Linux capable CVA6 Host
- Peripherals (JTAG, SPI, I2C)
- Running at 1+ GHz (WC),
> 256 GFLOP/s (FP64)
> 2 TFLOP/s (FP8)
- Tape-out August 2025

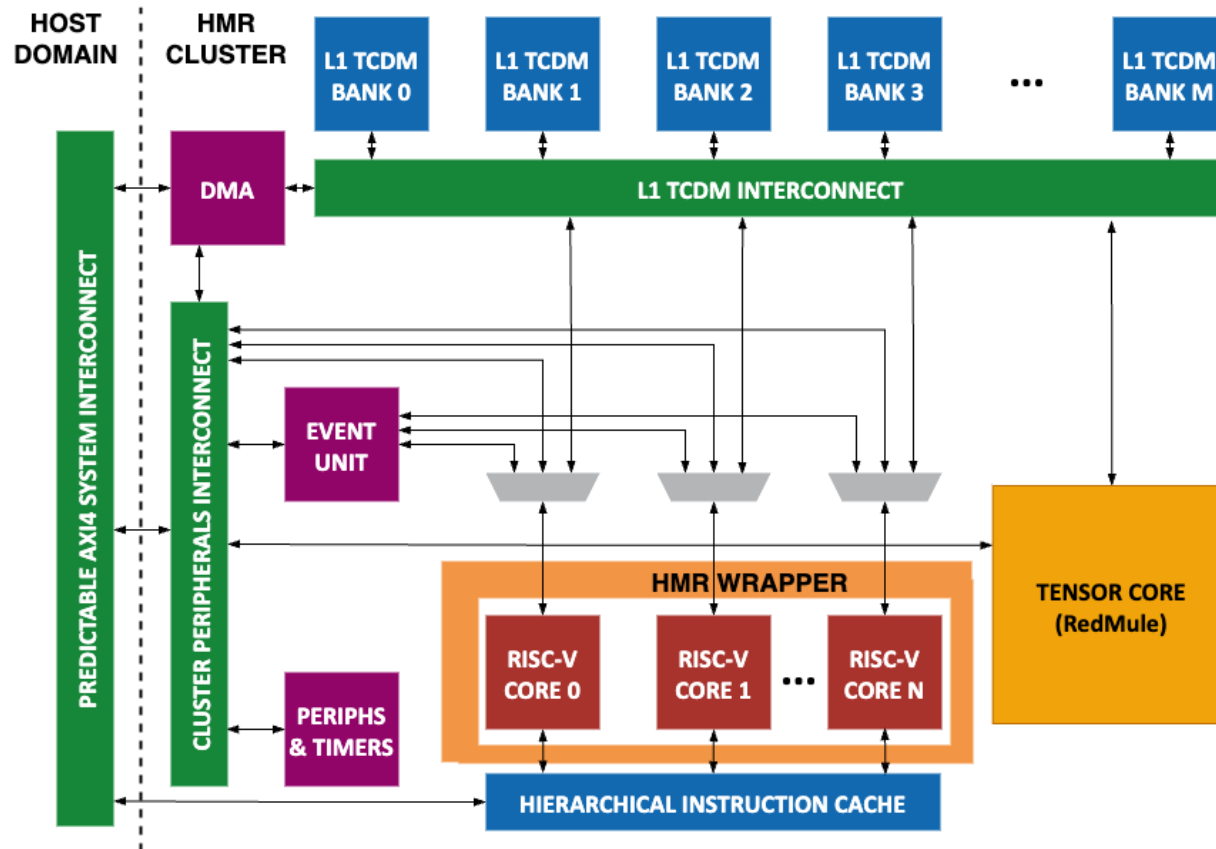
THE **EUPILOT**

Efficiency ✓ ... What about Reliability?

Space Processing - The Cost of Reliability



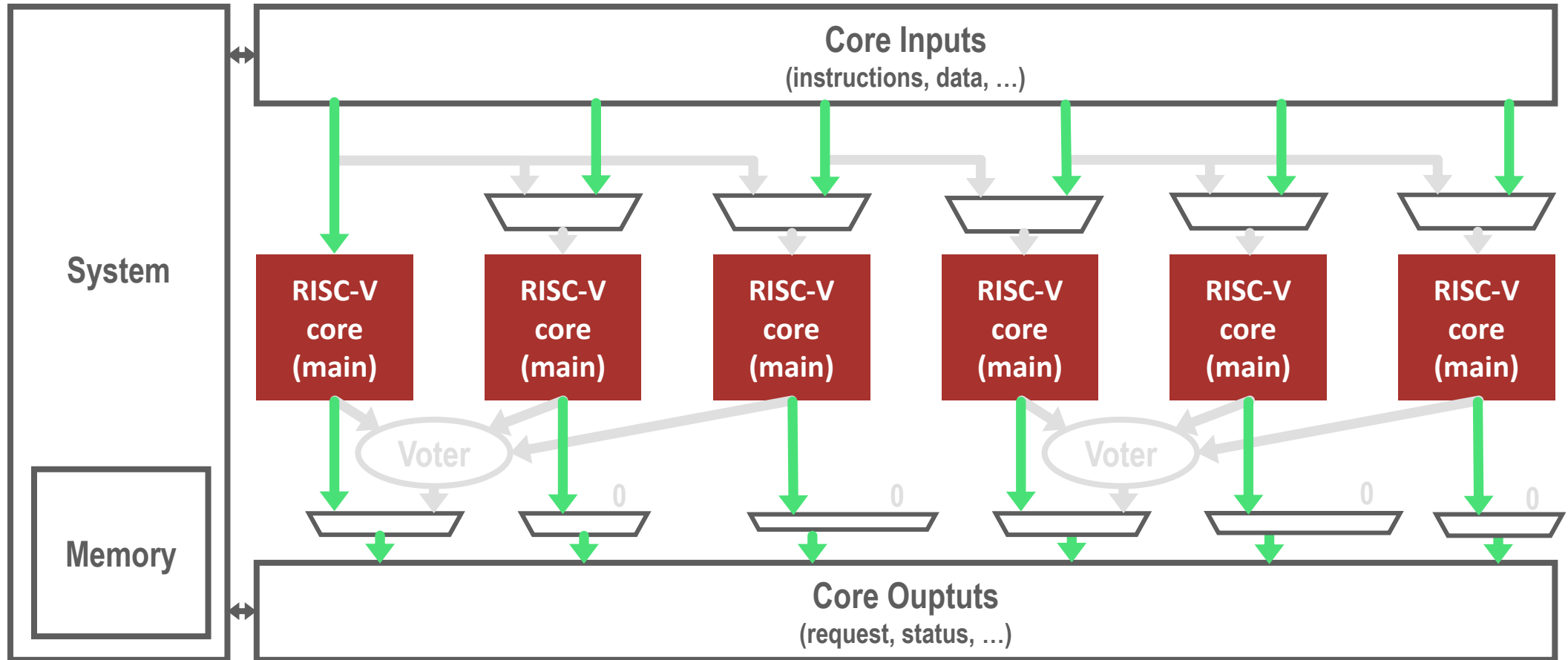
The Reliable Cluster



- **Need to protect**
 - Cores
 - Memories
 - Tensor Engine
 - Interconnect
 - Event Unit, Synchronizer, DMA
- **Key challenge: reduce the silicon cost of the large area contributors**
 - For “small stuff” TMR is OK

Hybrid Modular Redundancy (HMR)

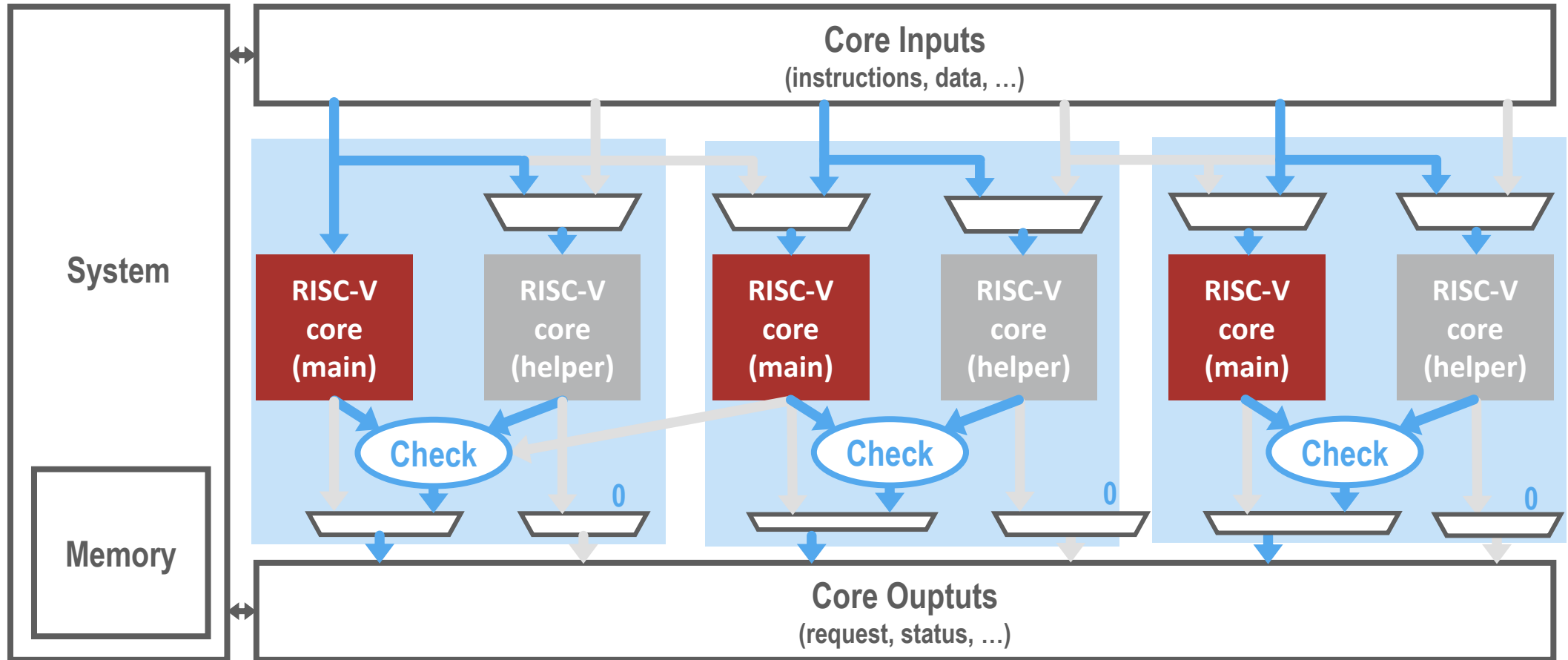
Independent Mode: high performance, no reliability



Hybrid Modular Redundancy (HMR): Reconfigurable



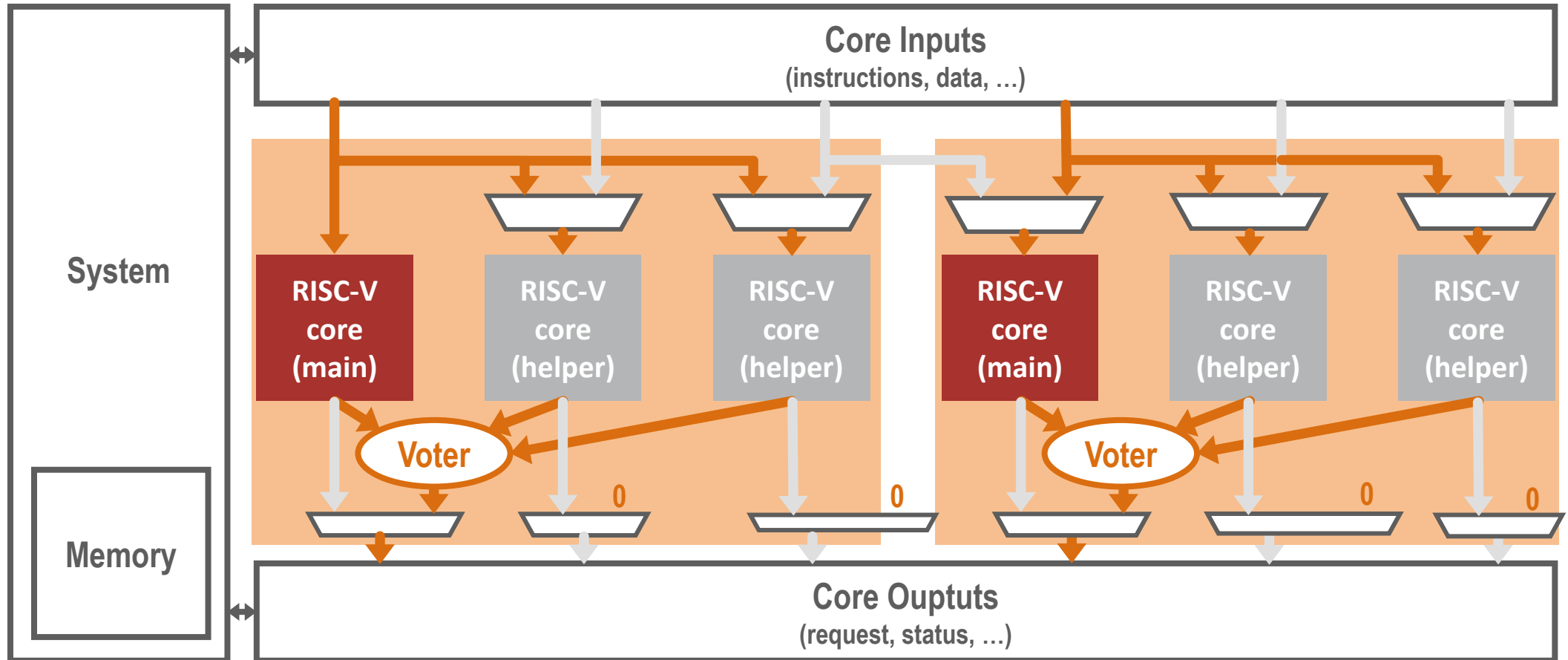
DMR Mode: good performance, good reliability, slow recovery



Hybrid Modular Redundancy (HMR)

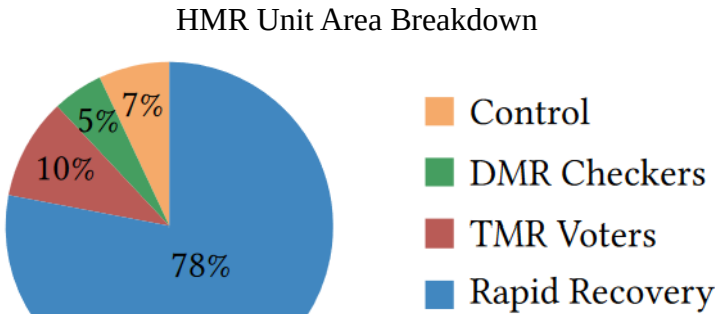
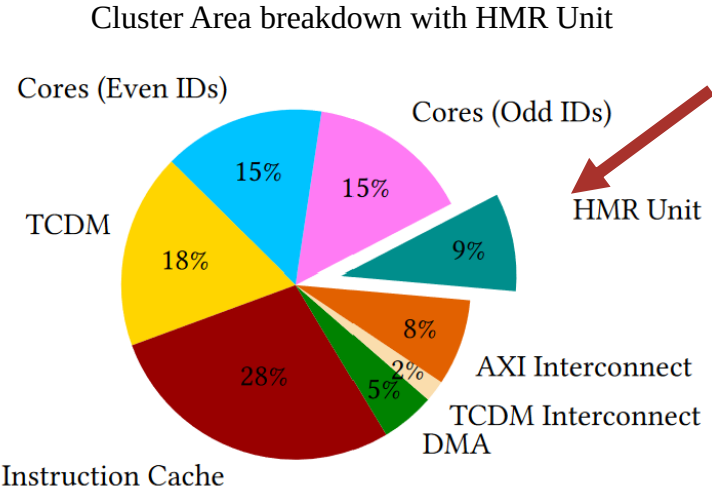


TMR Mode: low performance, high reliability, quick recovery



-
- The diagram illustrates the system architecture and the recovery routine. On the left, three cores are shown: a Main Core (red) and two Helper Cores (grey). Each core contains a Program Counter (PC), a Register File (RF), and a Control Status Register (CSR). The Main Core's RF is connected to the RFs of the Helper Cores. A Bitwise Checker/Majority Voter (orange block) receives inputs from the cores and outputs *inp_a*, *inp_b*, and *inp_c* to the cores. It also outputs an *error* signal to the Rapid Recovery Controller (black circle). The Rapid Recovery Controller outputs an RF Addr Gen signal to the RF registers. The Recovery Routine (black circle) is a state machine with four states: Idle, Clear, Halt, and Restore. Transitions are triggered by *error*, *!core_halted*, and *core_halted* signals. The Recovery Routine is connected to the RF registers via the RF Addr Gen signal.

The Cost of HMR



Area Overhead of HMR Configurations

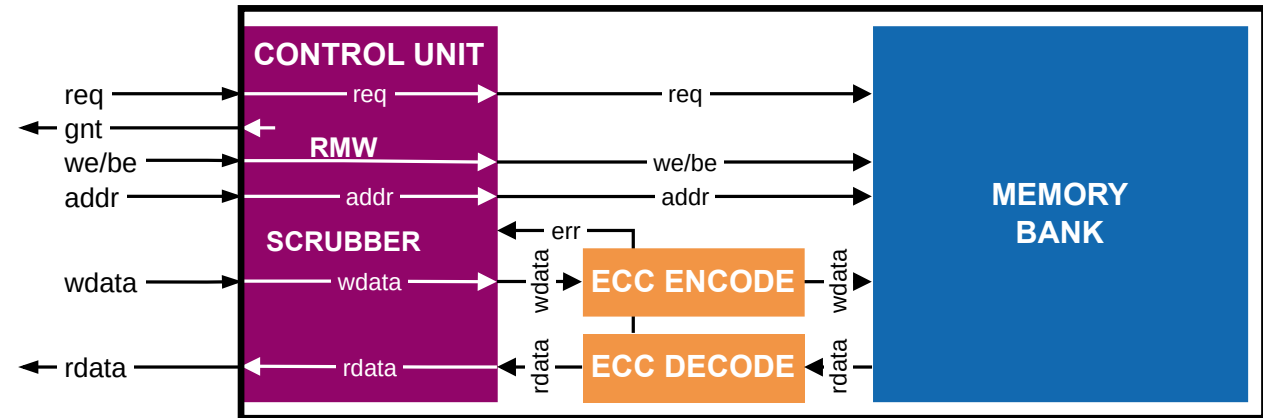
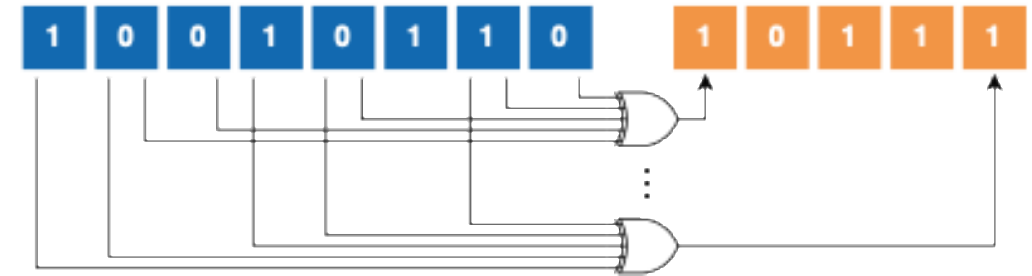
PULP Cluster Area [mm ²]	Overhead
Baseline	-
DMR	0.3%
TMR	0.7%
HMR	1.3%
With Rapid Recovery	
DMR	8.4%
TMR	8.8%
HMR	9.4%

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

Error-Correcting Codes (ECC) for Memory



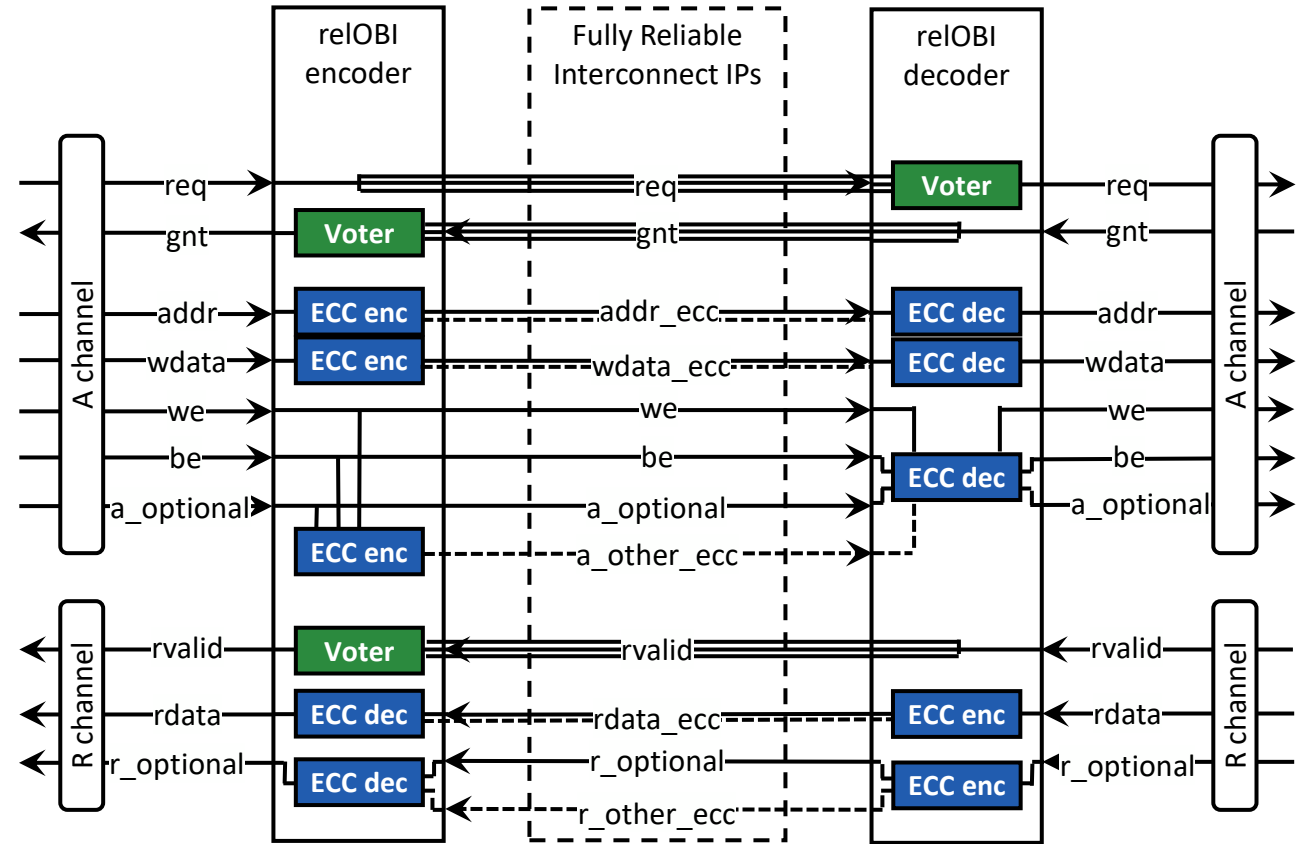
- **Hsiao code for efficient encoding**
 - Single Error Correction, Double Error Detection
- **32bit word stored as 39bit with parity**
- **Read-modify-write for efficient byte-wise access**
- **Scrubber for continuous correction**
 - Avoids latent errors causing uncorrectable error



relOBI: reliable low-latency interconnect

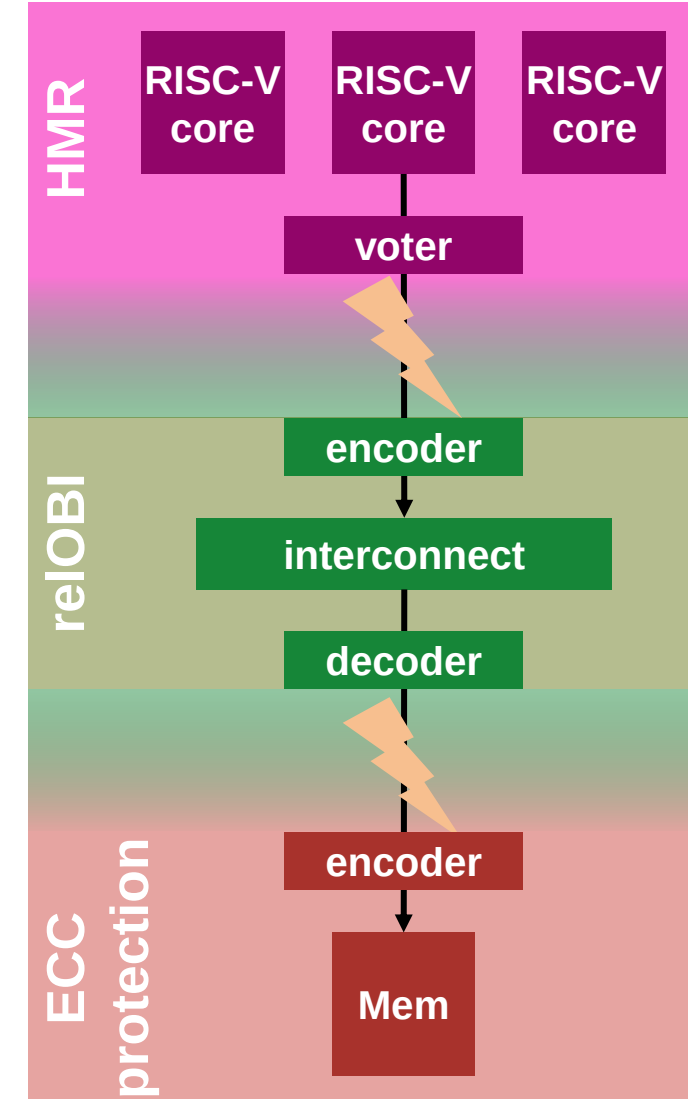


- **Open Bus Interface (OBI)**
 - SRAM-like with flow control
- **TMR for Handshake signals**
 - Minimal overhead for critical signals
- **ECC for data signals**
 - More compact for transport
 - Reuse memory encoding
- **En- & decoding for standard**
- **Reliable interconnect IP**
 - In-flight correction



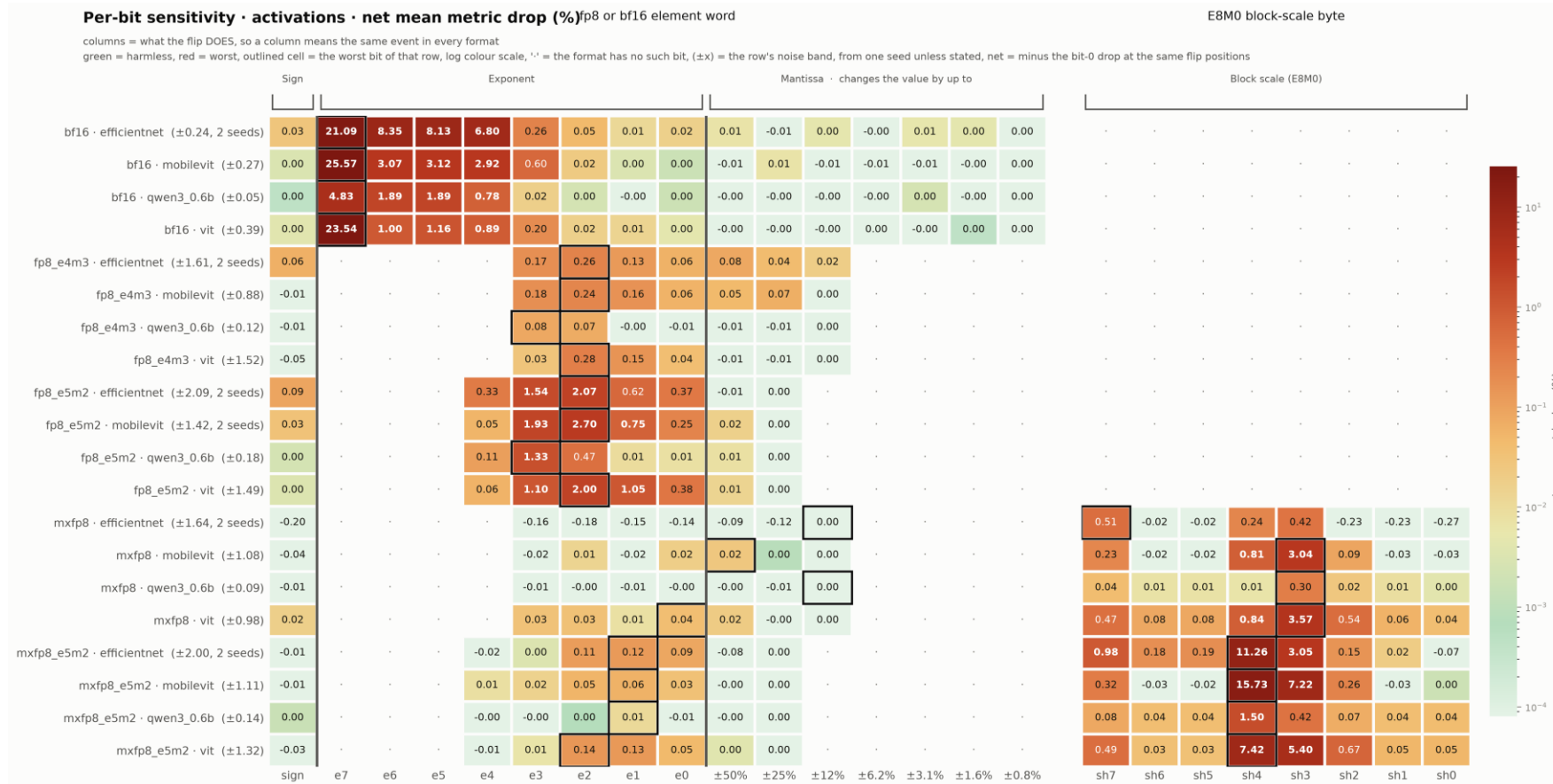
Who Checks the Checker?

- **Main domains in system now protected**
 - Cores, memory, interconnect
- **No protection in between!**
 - Connections, Voters, En-/decoders
- **Idea: overlap the protection domains**
 - Encode Interconnect before voting cores
 - Encode for memory before decoding interco.
- **Intermediate faults handled by overlapping protections!**



What About the Tensor Engine? FiReS-AI

The heat table: Each row is a model in one number format, and each column is a bit. Each cell shows how much the model's score drops when that bit flips, and different metrics can be chosen.

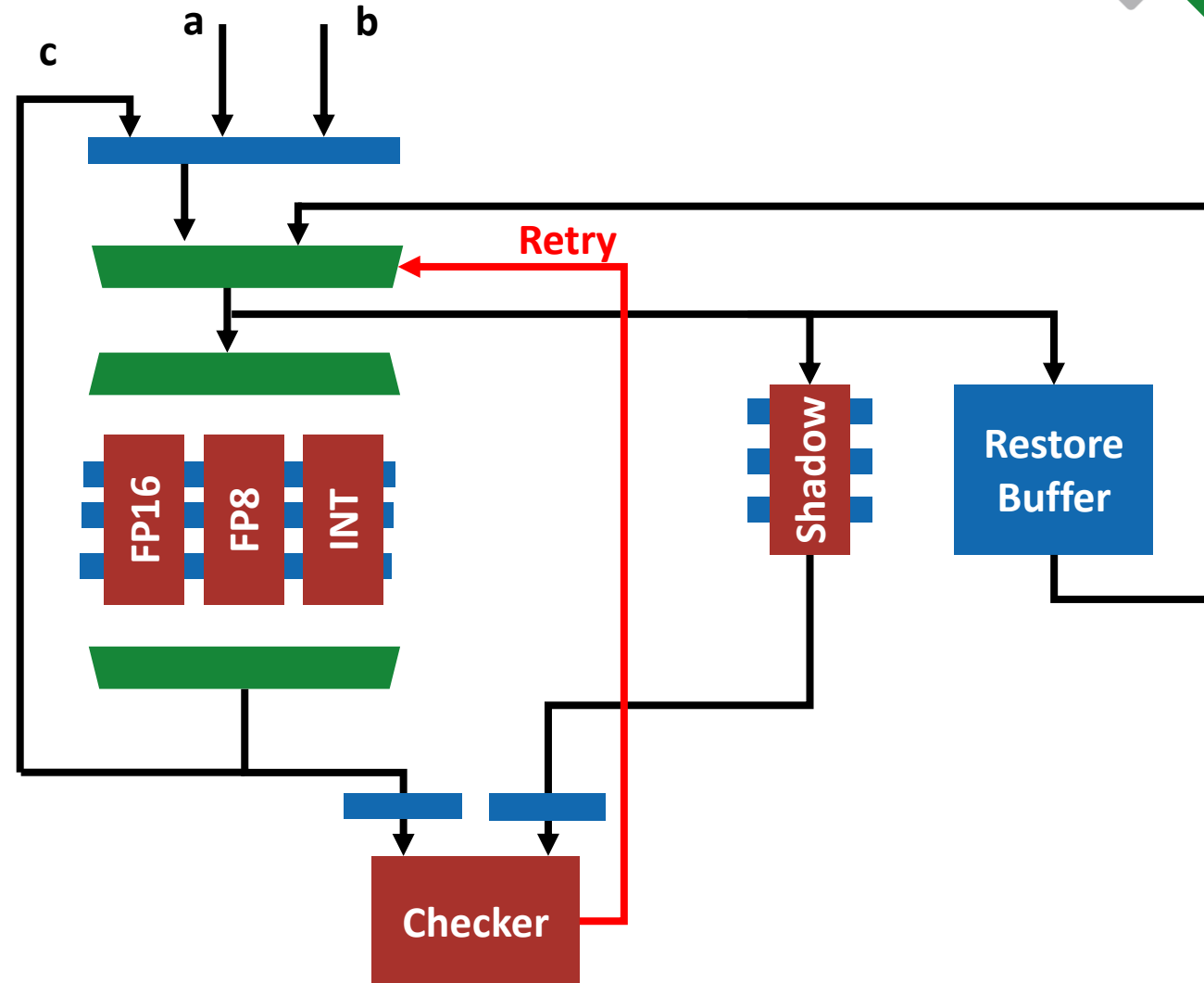


What about the Tensor Engine?

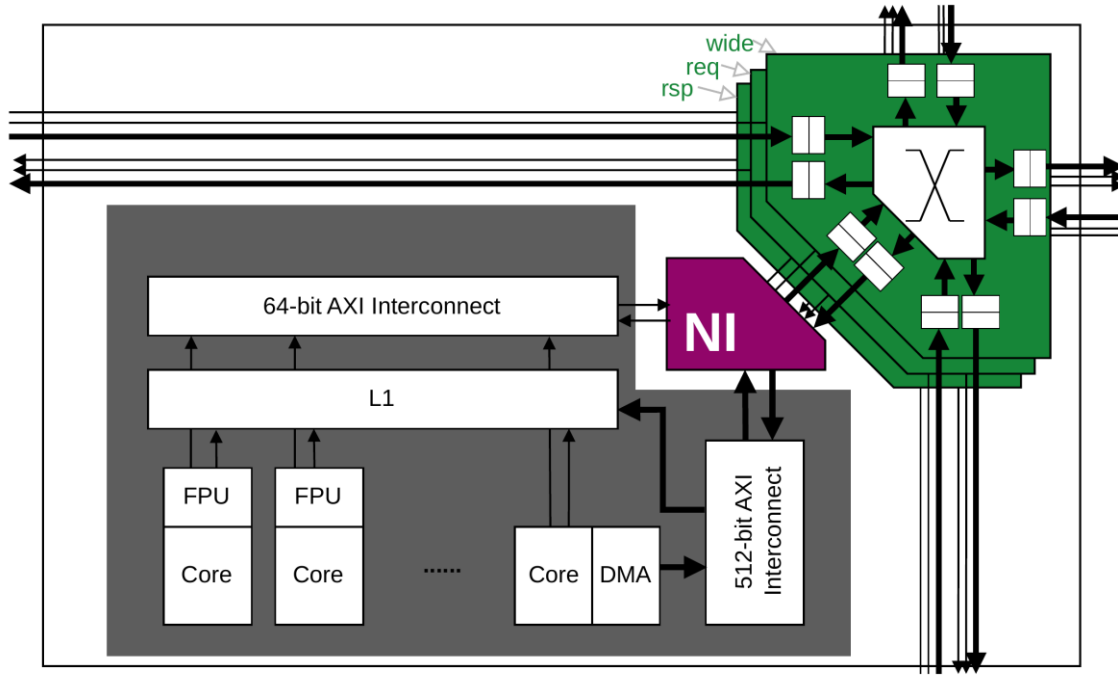


MAC unit with shadow Unit

- Reduced precision MAC unit for all FP types
- E.g. FP32 shadowed to BF16, FP16 to FP8..
- **Working principle:**
 - Main FPU computes at "nominal" precision
 - Shadow FPU performs approximate computing
 - Exponent width preserved
 - Fewer mantissa bits
 - Output checking done on exp + few mantissa bits
- **Advantages**
 - Reduced area for FPU-protection (66% wrt DMR)
 - Fewer recovery procedures compared to DMR



What about the NoC?

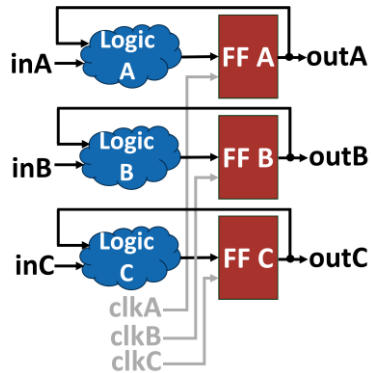


- **FlooNoC router** (open source, AXI4)
 - 2-cycle latency per hop, no internal pipeline stage
 - 3 physical links: 512-bit wide data, 2 × 64-bit narrow req/rsp
 - network interface (NI) maps AXI transactions onto flits

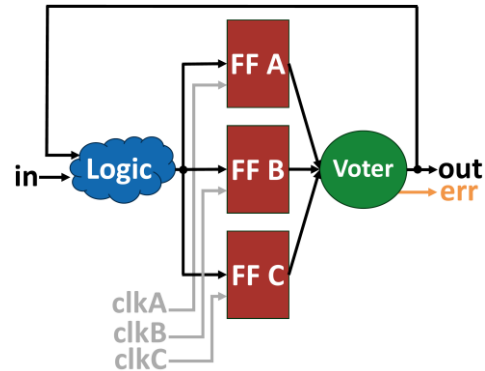
TMR Protection Variants for the NoC



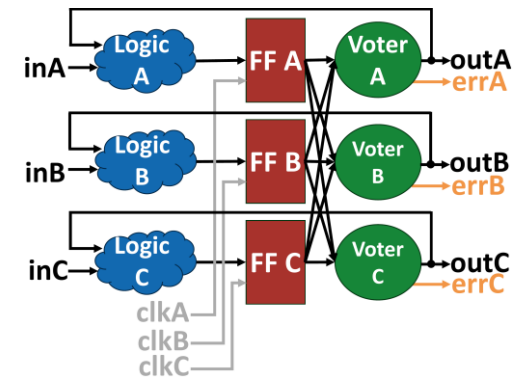
Coarse TMR



State-only TMR

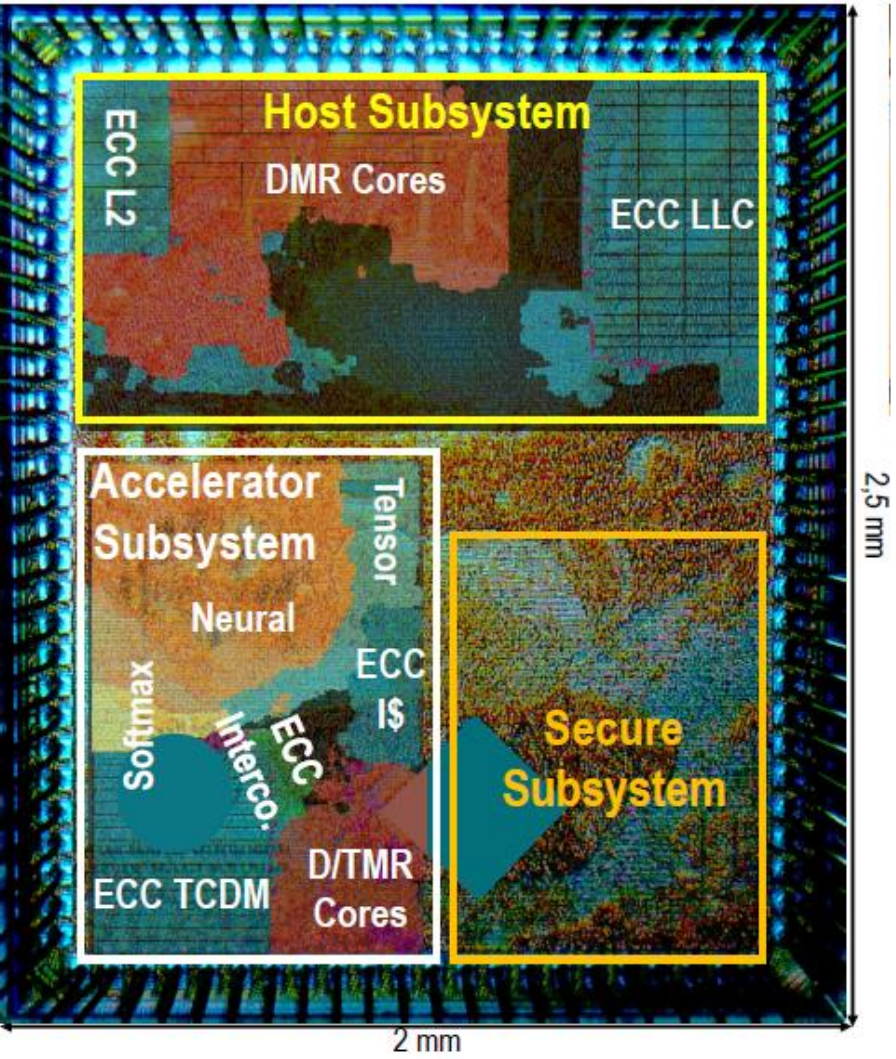


Full TMR



Is it Affordable? See Chen Wu's Talk, later today!

Astral: PULP's H-DSA for Space



Technology	Global Foundries 12 nm LP+
Area	5 mm ²
Power	380 mW (Host) 350 mW (Accelerator)
Frequency	780 MHz (Host) 620 MHz (Accelerator)
INT Performance	578 GOPS
INT Energy Efficiency	4.7 TOPS/W
FP Performance	58 GFLOPS
FP Energy Efficiency	557 GFLOPS/W
E2End Appl. Latency	17.5 ms (CloudViT) 1.5 ms (Reliable An. Det.)

Fault Recovery Latency 38.6 ns



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

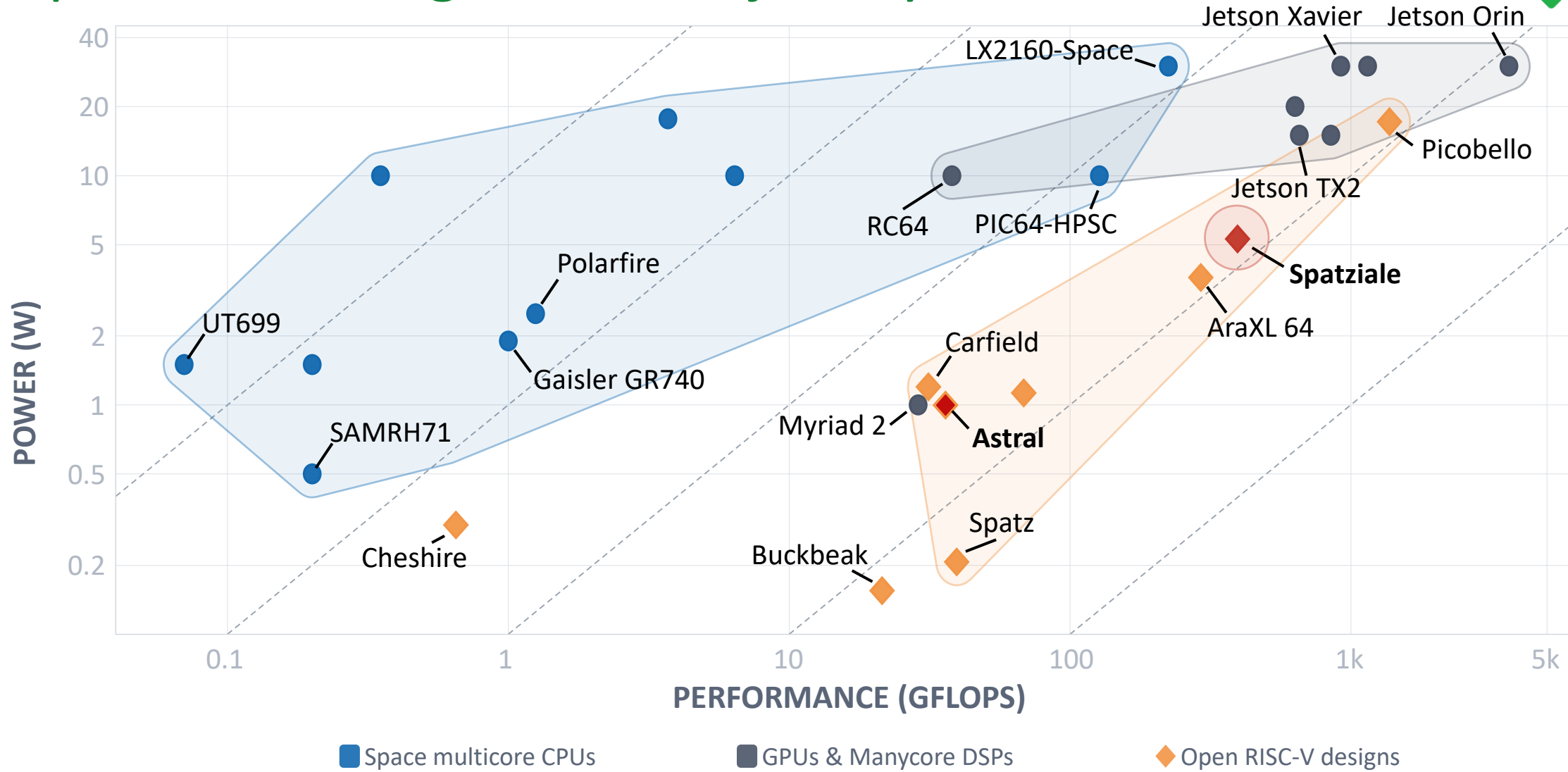
Chips-it
FONDAZIONE

ThalesAlenia
a Thales / Leonardo company
Space

ETH zürich



Space Processing: PULP's Trajectory



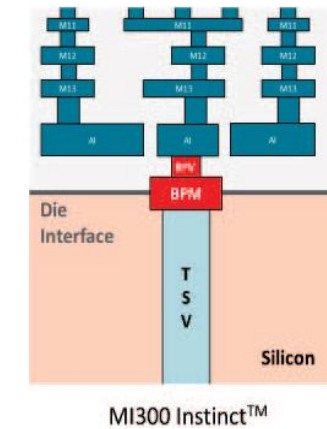
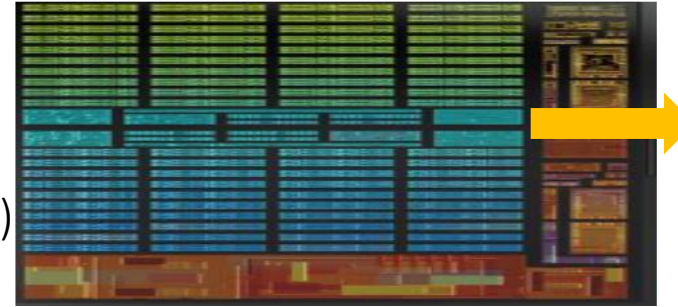
What's Next: 3D!



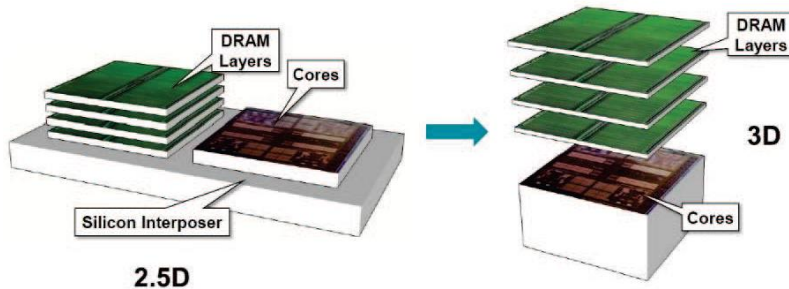
Logic die
Memory + IO die

- 3.5D v1

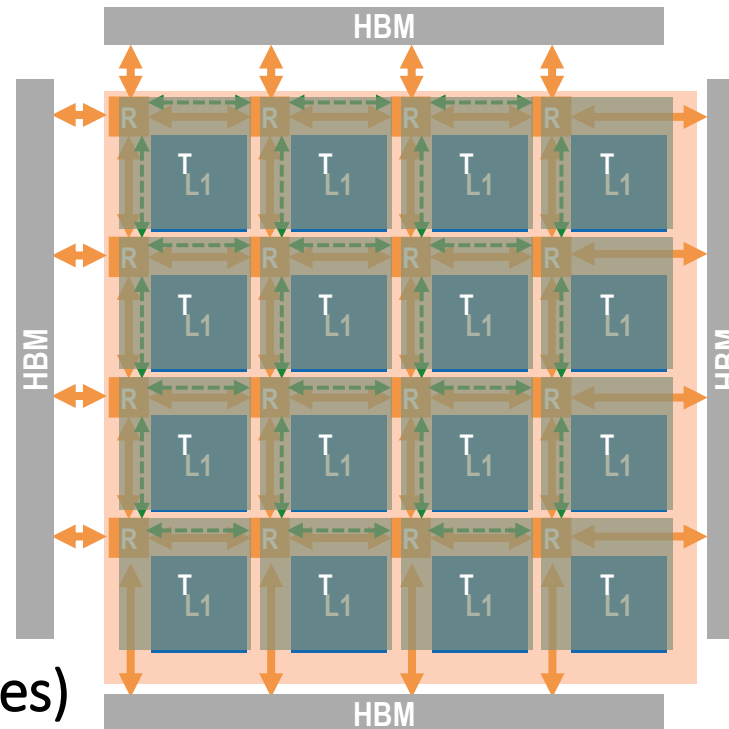
- 3D stacking on logic + 2.5D HBM (AMD MI300)
- Face (top) to Back (bottom)
- Die (top) to Wafer (bottom)



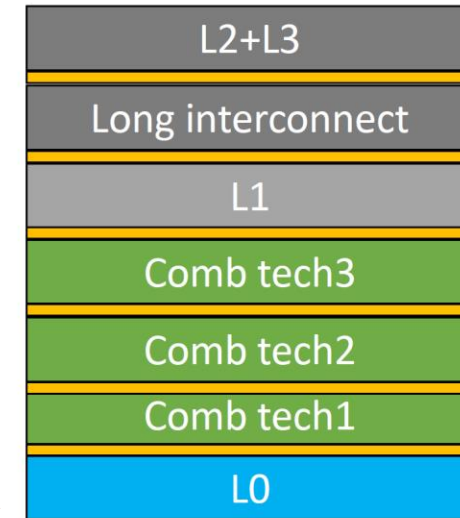
- 3.5D v2?



- Monolithic 3D (CMOS2.0+3D memories)



V1
SRAM+NOC+IO at the bottom



Toward “full 3D” → Open Phy.AI Efficient&Reliable 3D platform



Thank You!



youtube.com/pulp_platform



pulp-platform.org



[@pulp_platform](https://twitter.com/pulp_platform)