



PULP PLATFORM

Open Source Hardware, the way it should be!

# Open Platforms for the Embodied AI era

Luca Benini <luca.Benini@unibo.it, lbenini@ethz.ch>



European Research Council



EuroHPC  
Joint Undertaking



FNSNF

FONDS NATIONAL SUISSE  
SCHWEIZERISCHER NATIONALFONDS  
FONDO NAZIONALE SVIZZERO  
SWISS NATIONAL SCIENCE FOUNDATION



KDT JU

KEY DIGITAL  
TECHNOLOGIES  
JOINT UNDERTAKING

**ETH** zürich



<http://pulp-platform.org>



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

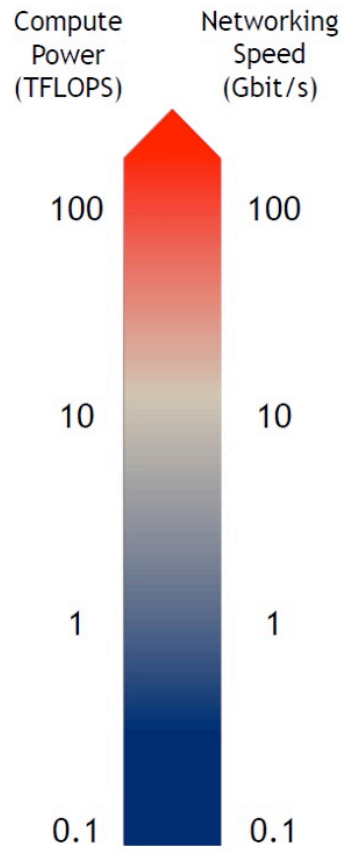
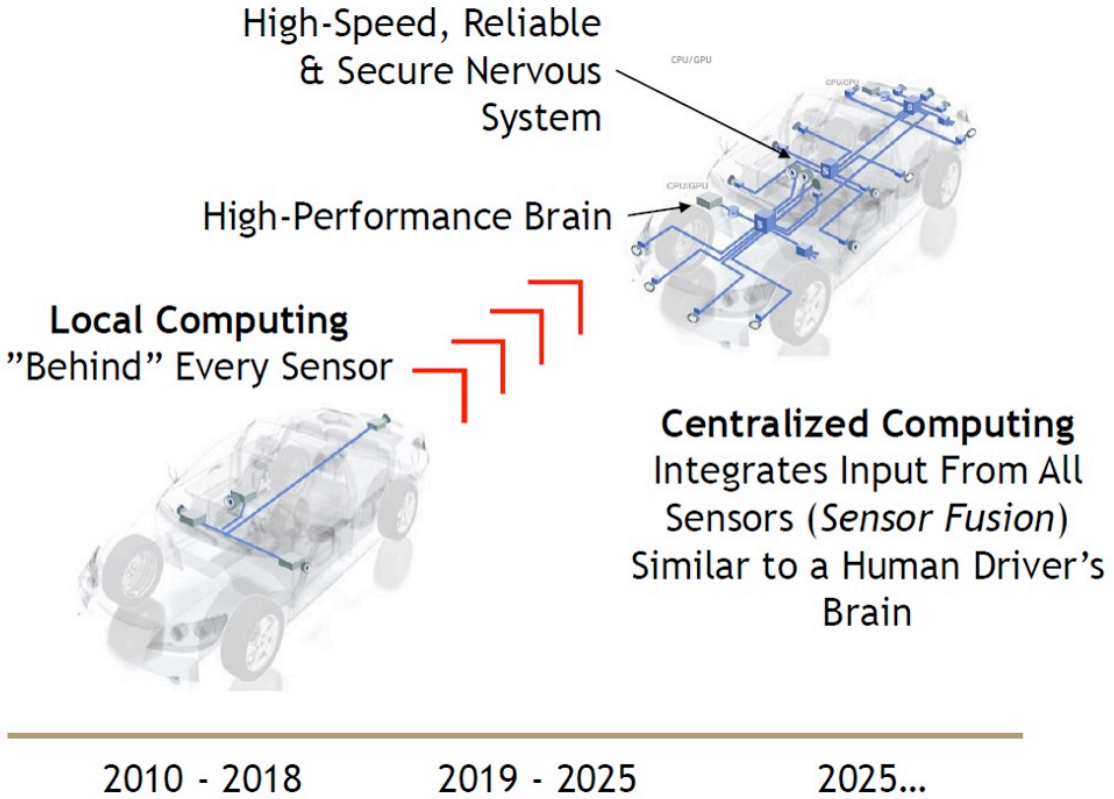


[https://www.youtube.com/pulp\\_platform](https://www.youtube.com/pulp_platform)

# Embodied AI

## Path Towards Full Autonomy

[SCR'23]



**Efficient**

**On-car Computing  
P<sub>MAX</sub> < 1.5KW**

**Energy Efficiency**  
 $\left( \frac{1}{\text{Power} \cdot \text{Time}} \right)$

**10x/12Y by scaling  
vs. model complexity  
10x/2Y**



**Safe**



**Real-time**



**Secure**

# Start Small: Open Platform for Autonomous Nano-Drones

## Advanced autonomous drone

[1] A. Bachrach, "Skydio autonomy engine: Enabling the next generation of autonomous flight," IEEE Hot Chips 33 Symposium (HCS), 2021



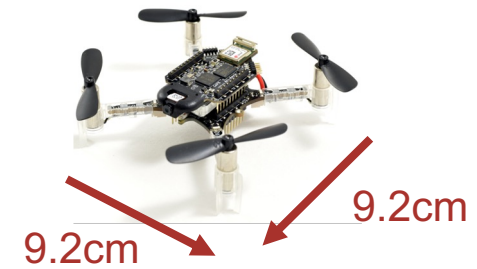
<https://www.skydio.com/skydio-2-plus>



- 3D Mapping & Motion Planning
- Object recognition & Avoidance
- 0.06m<sup>2</sup> & 800g of weight
- Battery Capacity 5410mAh



## Nano-drone

<https://www.bitcraze.io/products/crazyflie-2-1>

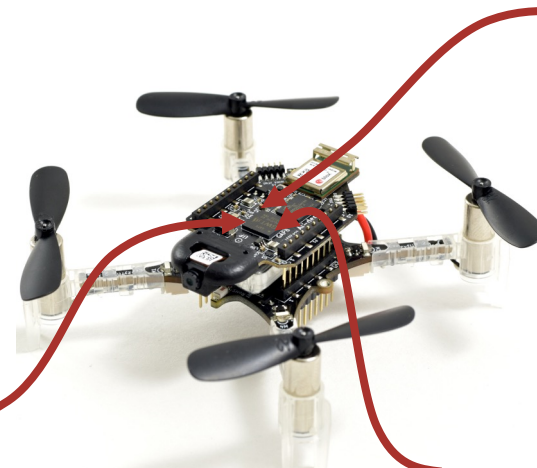


- Smaller form factor of 0.008m<sup>2</sup>
- Weight **27g (30X lighter)** 
- Battery capacity **250mAh (20X smaller)** 

**Can we fit sufficient intelligence in a 30X smaller payload, 20X lower energy budget?**

# Achieving True Autonomy on Nano-UAVs

Multiple, complex, heterogeneous tasks at high speed and robustness **fully on board**



Object detection



Obstacle avoidance & Navigation



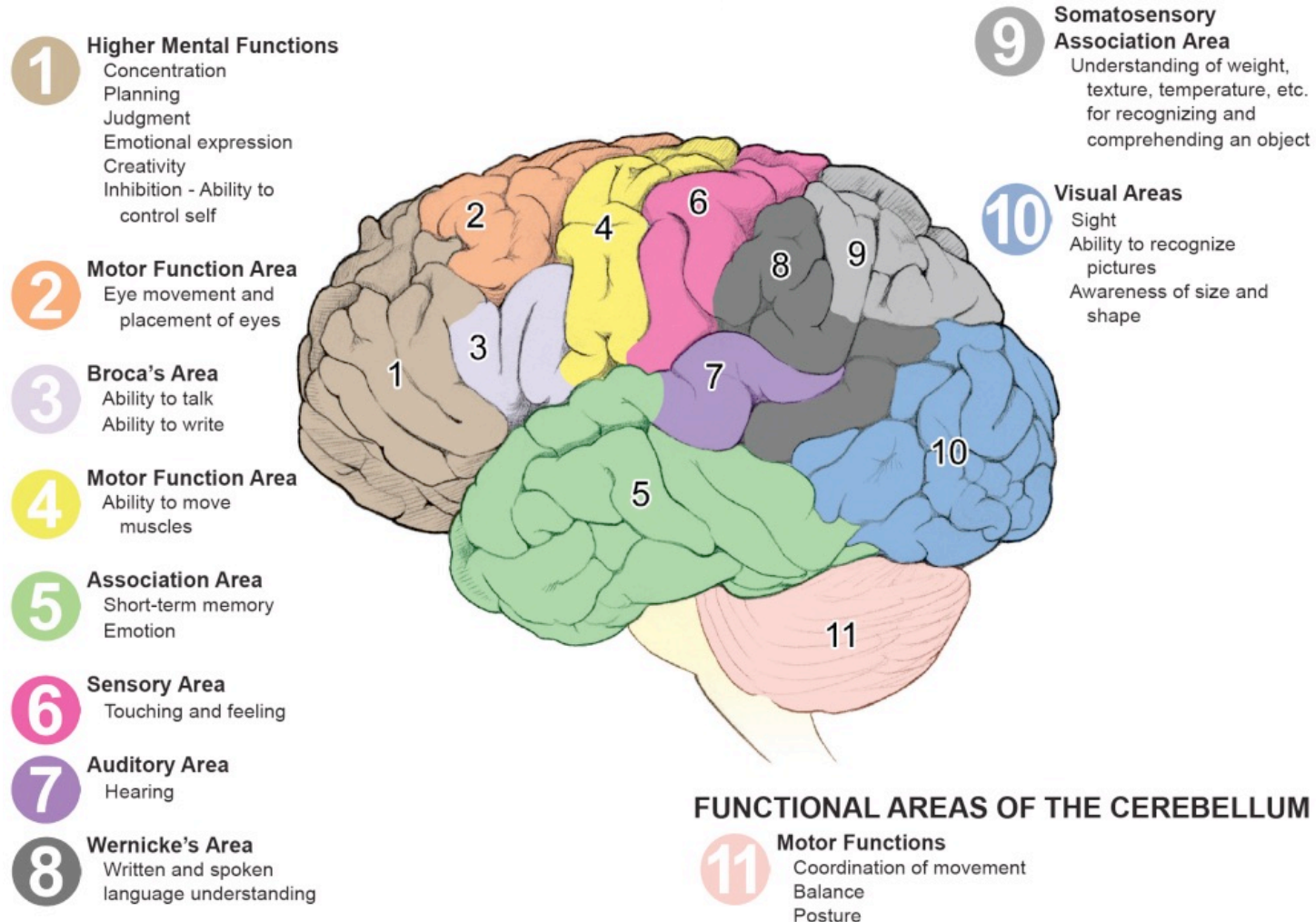
Environment exploration



**Multi-GOPS workload at extreme efficiency  $\rightarrow P_{\max}$  100mW**

# Multiple Heterogeneous Accelerators

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

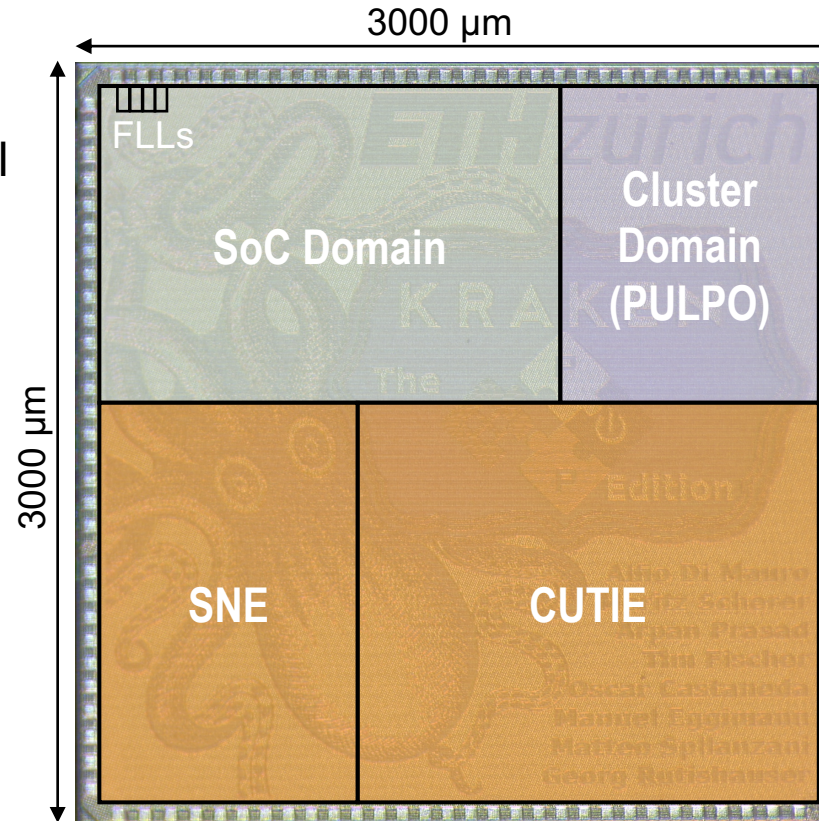


# Multiple Heterogeneous Accelerators

## The *Kraken*: an “Extreme Edge” Brain



- RISC-V Cluster (8 Cores + 1)
- CUTIE – dense ternary neural network accelerator
- SNE – energy-proportional spiking neural network accelerator



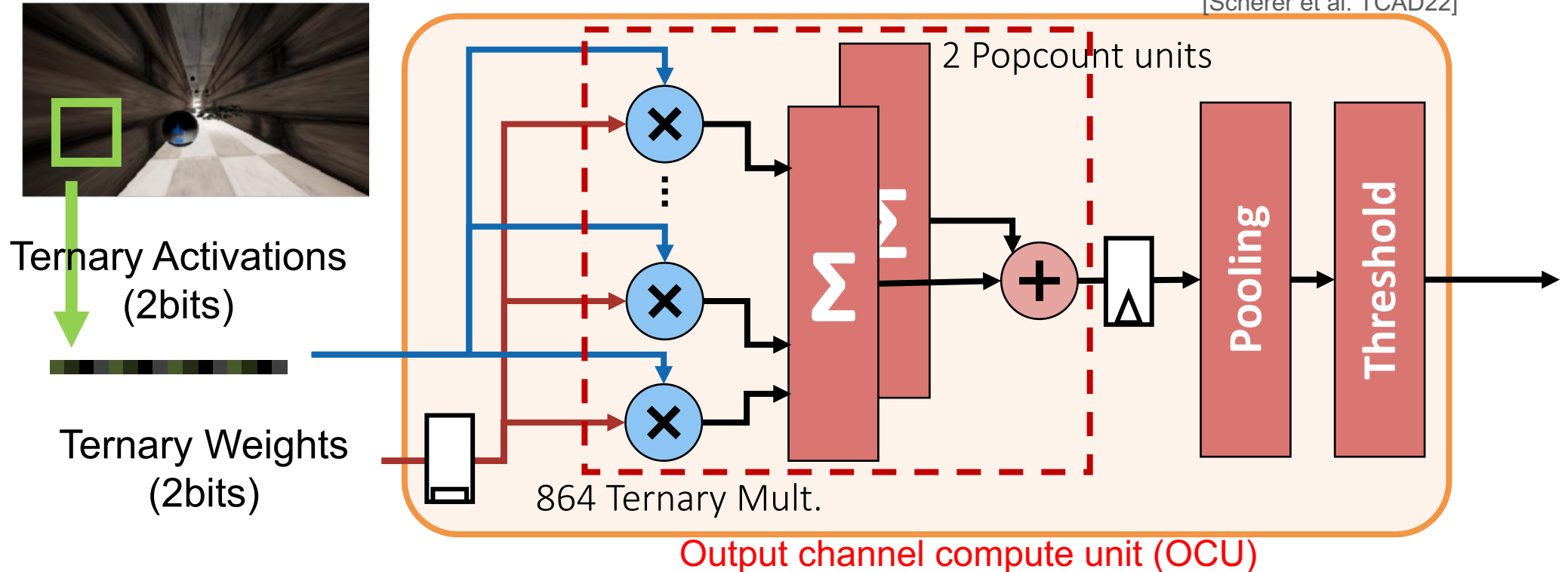
Technology	22 nm FDSOI
Chip Area	9 mm <sup>2</sup>
SRAM SoC	1 MB
SRAM Cluster	128 KB
VDD range	0.55 V - 0.8 V
Cluster Freq	~370MHz
SNE Freq	~250MHz
CUTIE Freq	~140MHz

[Di Mauro HotChips22]

**HOT**  
C H I P S

# CUTIE: Minimize Switching Activity & Data Movement

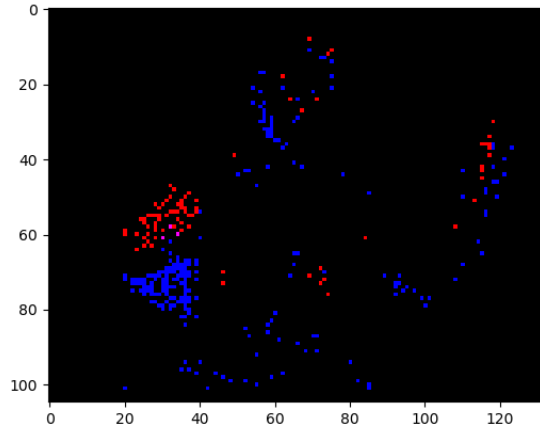
[Scherer et al. TCAD22]



- KxK window on all input channels unrolled, cycle-by-cycle sliding
- Completely unrolled inner products one output activation per cycle!
- Zeros in weights and activations, spatial smoothness of activations reduce switching activity
- 96 OCUs, 96 Input channels, 3x3 kernels:  $96 * 96 * 3 * 3 = 82'944 \text{ TMAC/cycle} (\sim 1\text{fJ/MAC})$

# Different Sensor Type, different Acceleration Engine

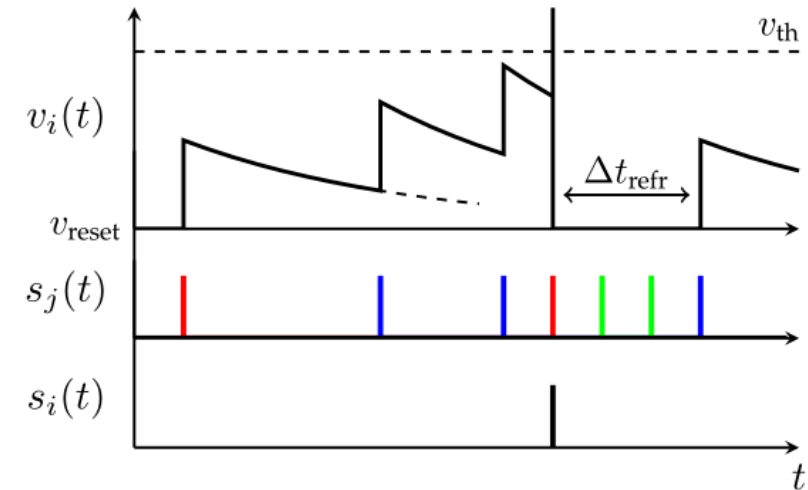
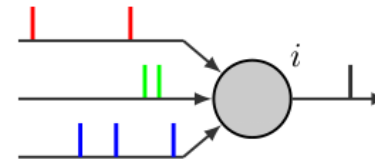
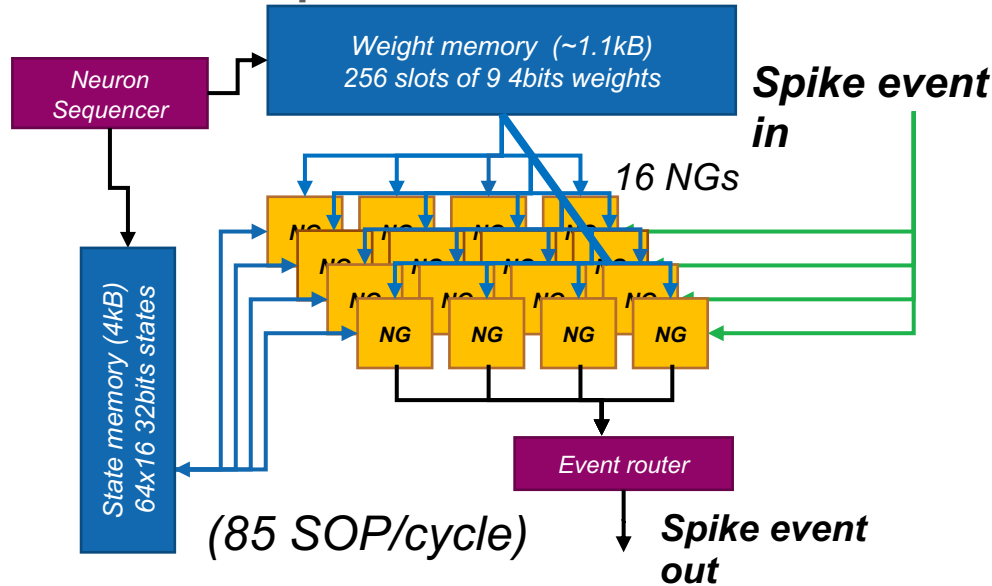
**Event Sensors:  
DVS  
Ultra-low latency  
Energy-  
proportional  
interface**



**Leaky Integrate & Fire (LIF) neurons**

**Spiking Neural Engine (SNE)**

[Di Mauro et al. DATE22]



**SNE works seamlessly with DVS (event-based) sensors**



# General Purpose PE: Domain-Specialized RV32 Core

 **RISC-V<sup>®</sup> 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

**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!

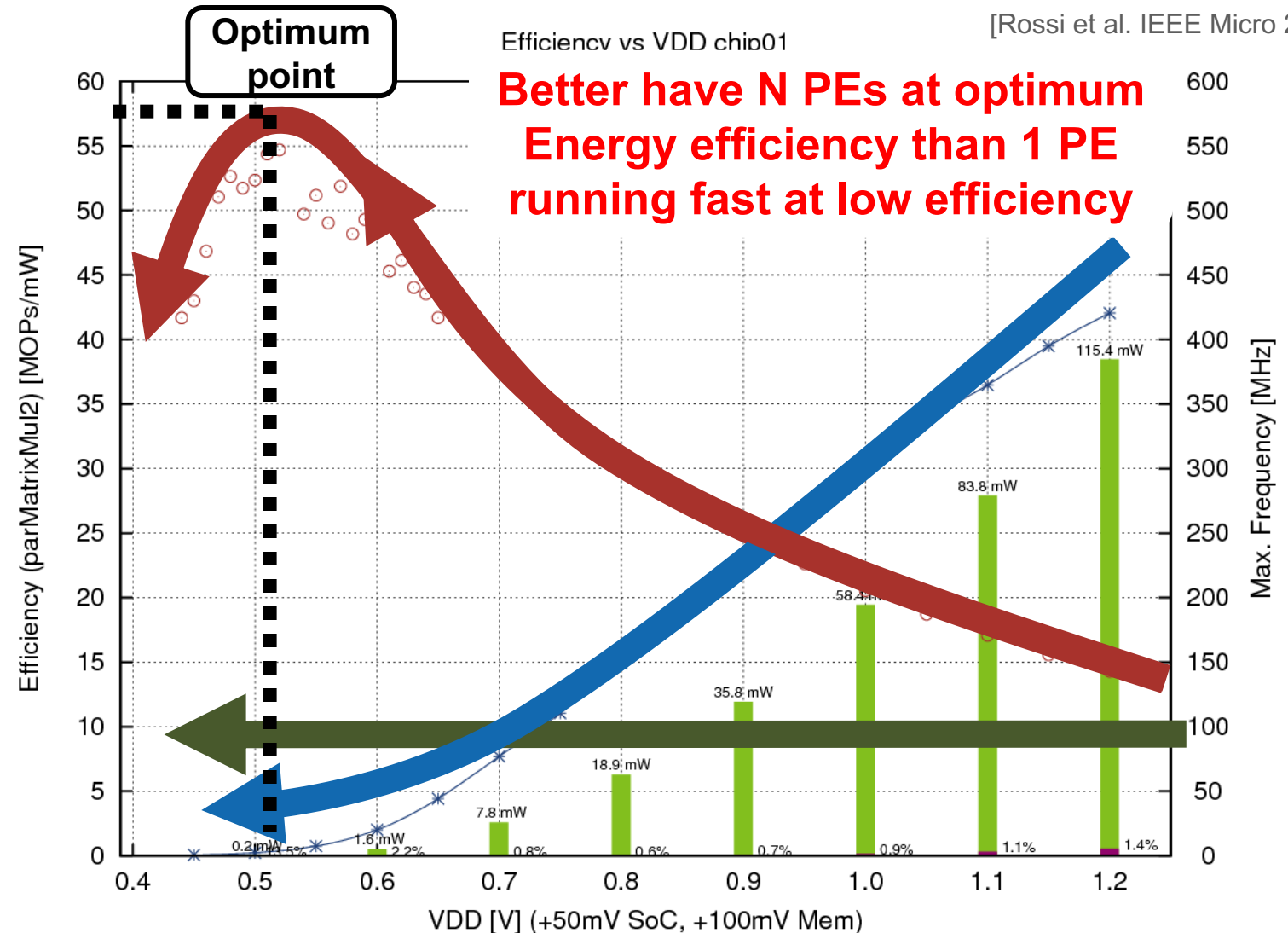
**Specialization Cost: Power,Area: 1.5x<sup>↑</sup> but Time 15x<sup>↓</sup> → E = PT 10x<sup>↓</sup>**

# Parallel, Ultra-Low Power (PULP) PE Cluster

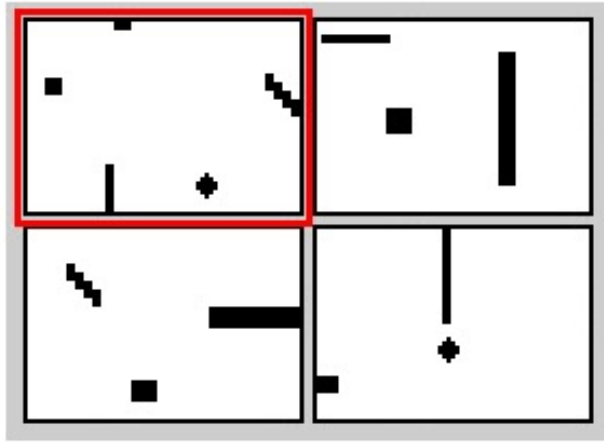


- As VDD decreases, operating speed decreases
- However efficiency increases → more work done per Joule
- Run parallel to get performance and efficiency!

**AI is parallel and scales  
More parallel with NN  
size**

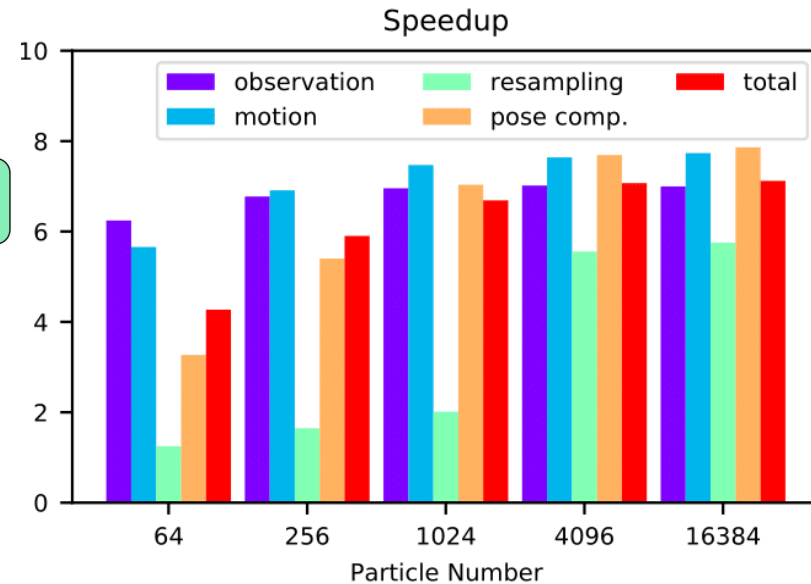
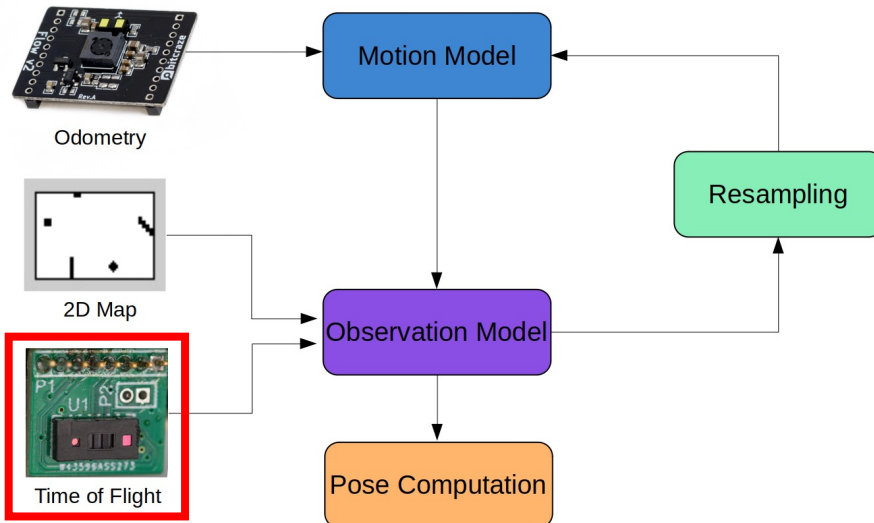


# Not only Perception: SLAM, Planning



## Particle filter-based

## Convergence + Low ATE for $N_{part} > 1024$ , 2ToF, FP16 acceptable



12MHz, 1Kpart. 13mW, 60msec  
 400MHz, 1Kpart 61mW, 1msec  
 400MHz 16Kpart 61mW, 30msec

# Advancing the SOA on all tasks

## RISC-V Cluster

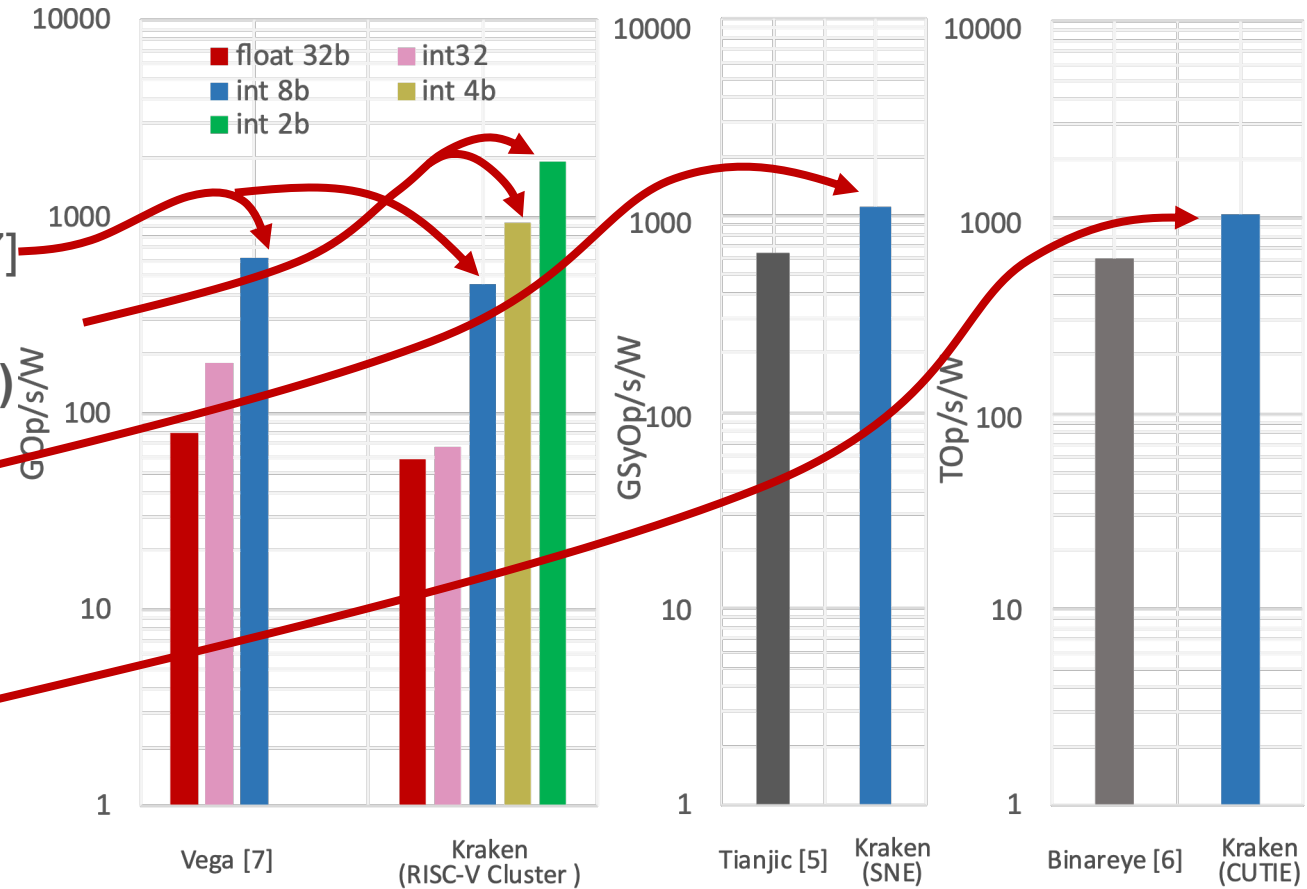
- Comparable 32bits-8bits SOA Energy efficiency to other PULPs [7]
- The highest energy efficiency on sub-byte SIMD operations (4b-2b)

## SNE

- 1.7X higher than SOA [5] energy/efficiency

## CUTIE

- 2X higher energy efficiency improvement over SOA [6]



**CUTIE, SNE can work concurrently for SNN + TNN “fused” inference (never done so far)**

# From Drones to Cars: Stepping up

## ▪ Microcontroller class of devices

- Infineon AURIX Family MCUs
- **Control tasks, low-power sensor acquisition & data processing** Features: lockstepped **32-b HP TriCore CPU** , HW I/O monitor, dedicated accelerators

## ▪ Powerful real-time architectures

- ST Stellar G Series (based on ARM Cortex-R cores)
- **Domain controllers and zone-oriented ECUs**
- Features: HW-based virtualization, Multi-core **Cortex-R52** (+NEON) cluster in split-lock, vast I/Os connectivity

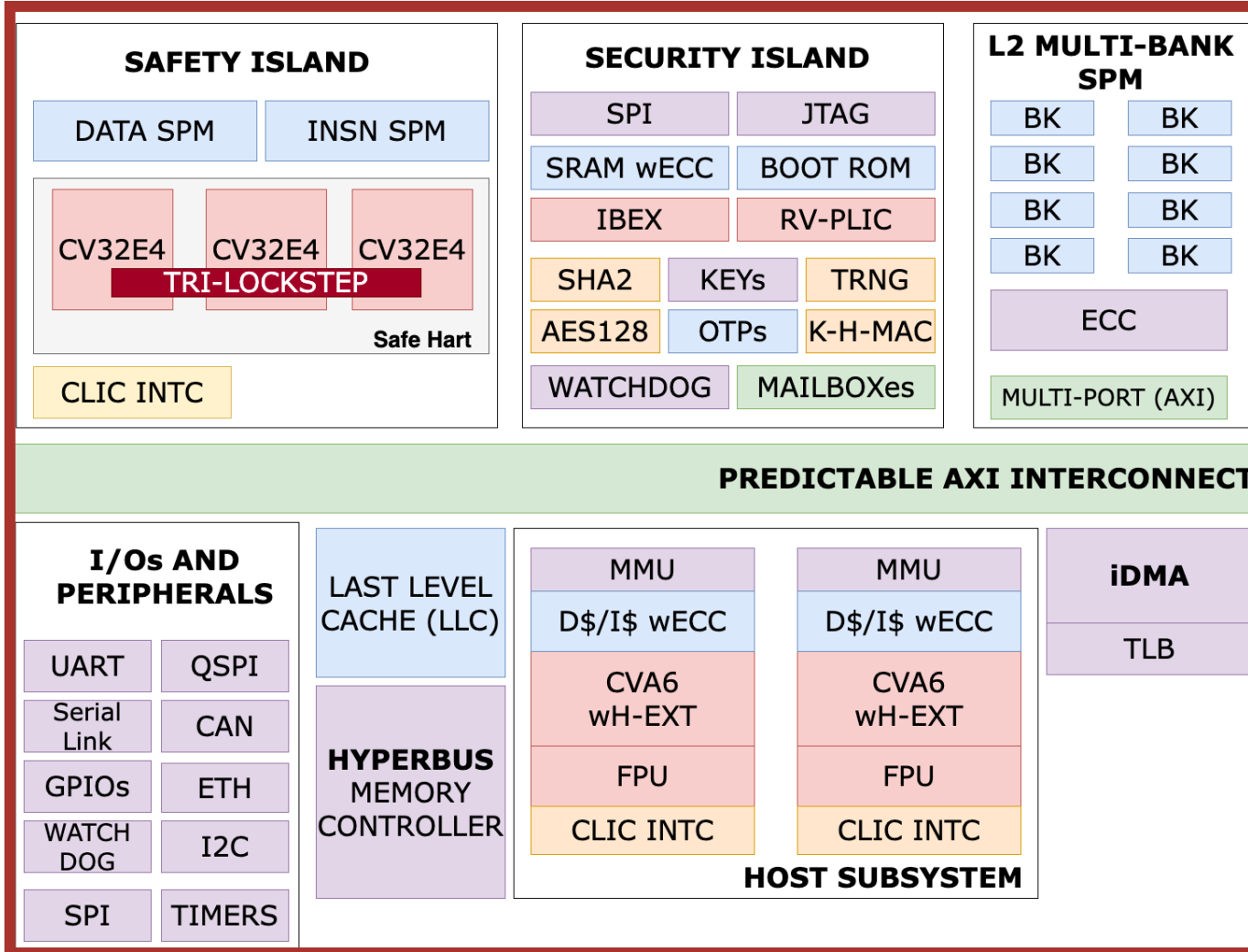
## ▪ Application class processors

- NXP i.MX 8 Family
- **ADAS, Infotainment**
- Features: Cortex-A53, **Cortex-A72**, HW Virtualization, **GPUs**

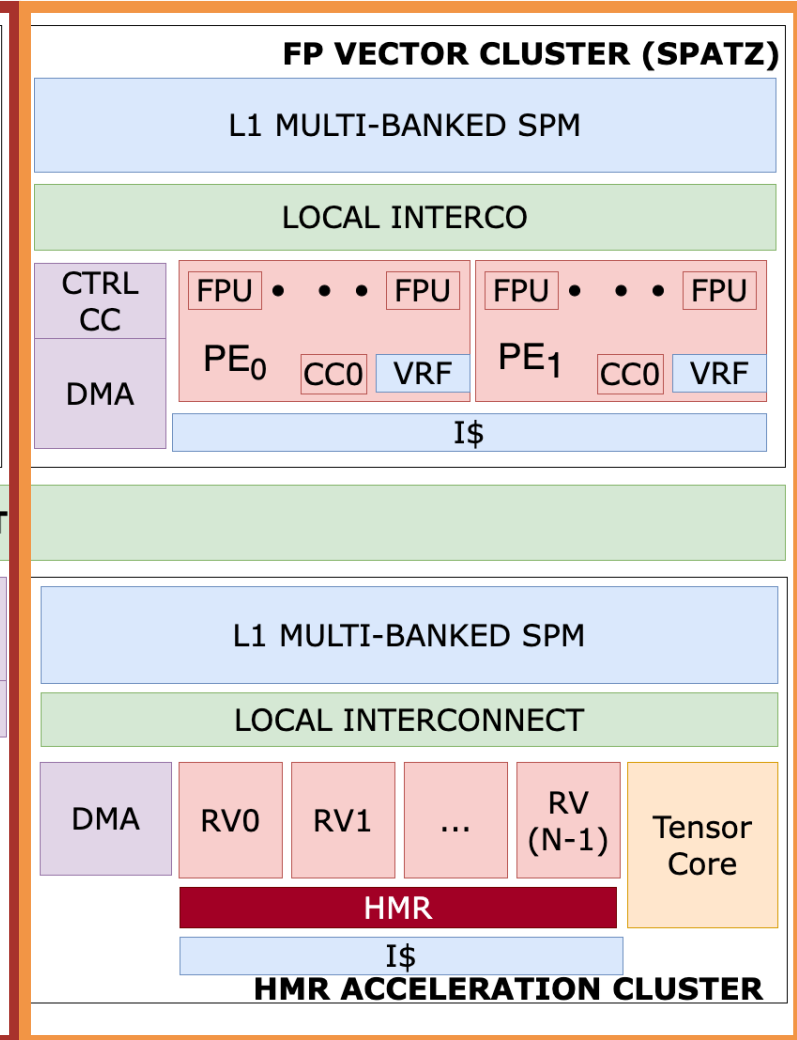


# Carfield: Efficiency + Safety, Security, RT-Predictability

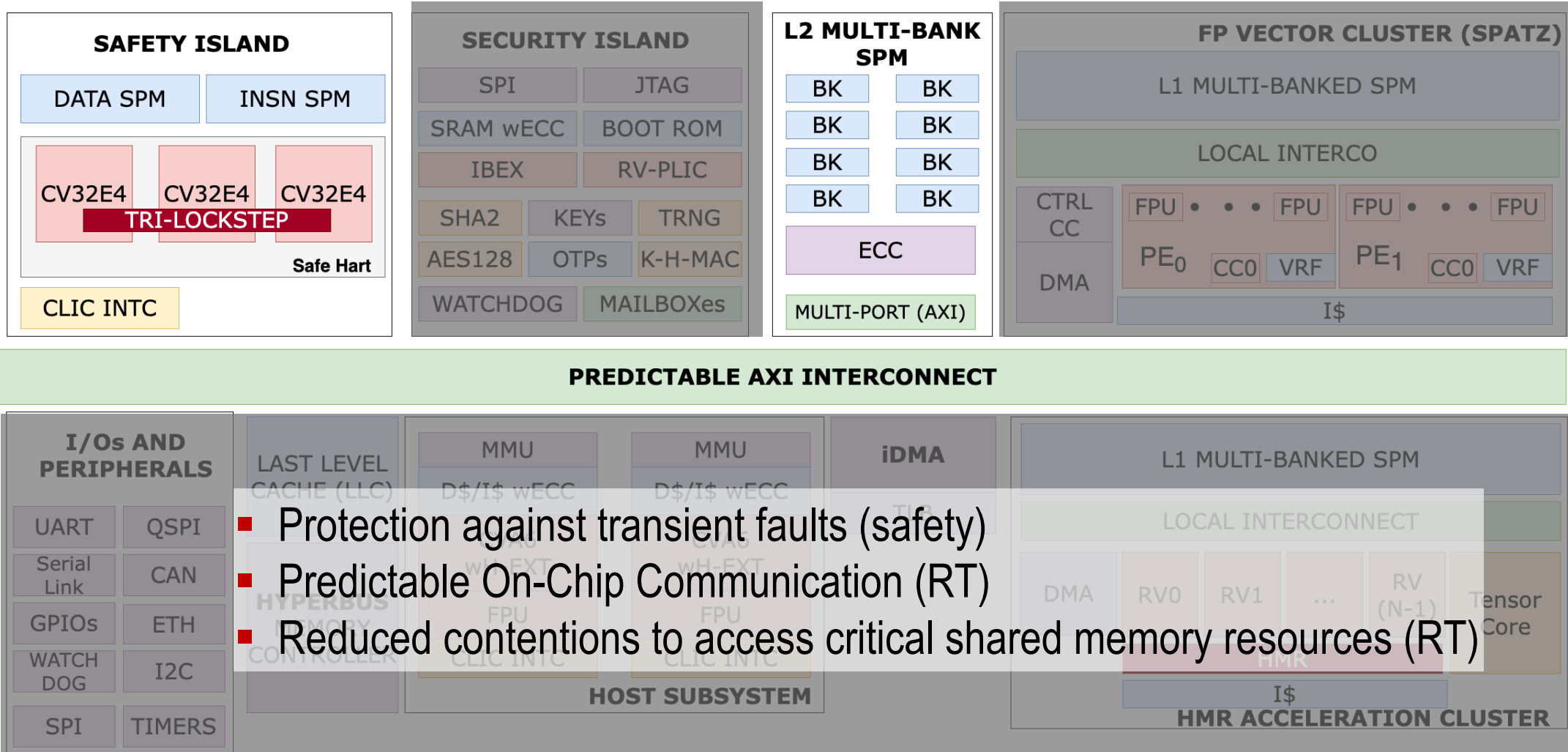
## Main Computing and I/O System



## Accelerators Domain

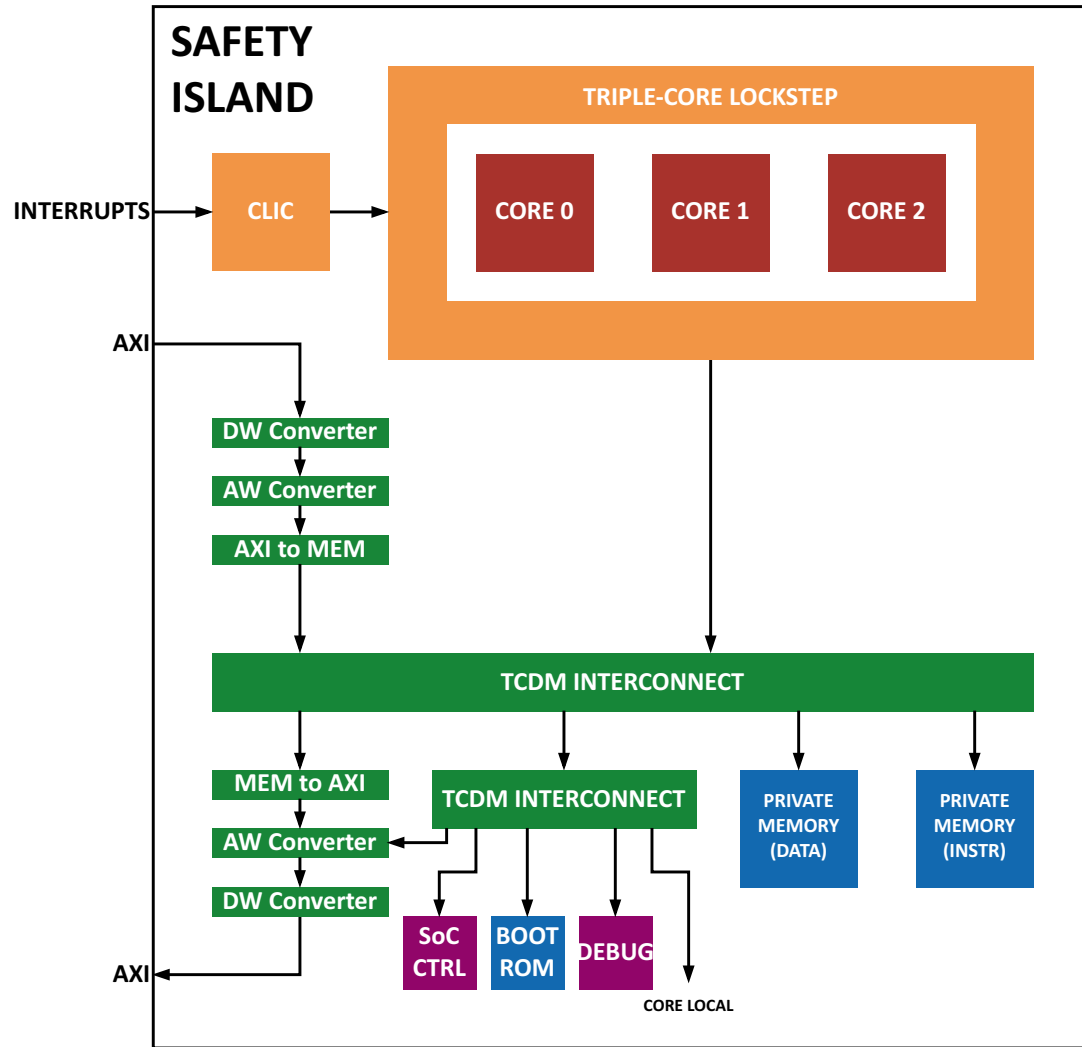


# How Do We Handle Safety-Critical and Real-Time Tasks?



- Protection against transient faults (safety)
- Predictable On-Chip Communication (RT)
- Reduced contentions to access critical shared memory resources (RT)

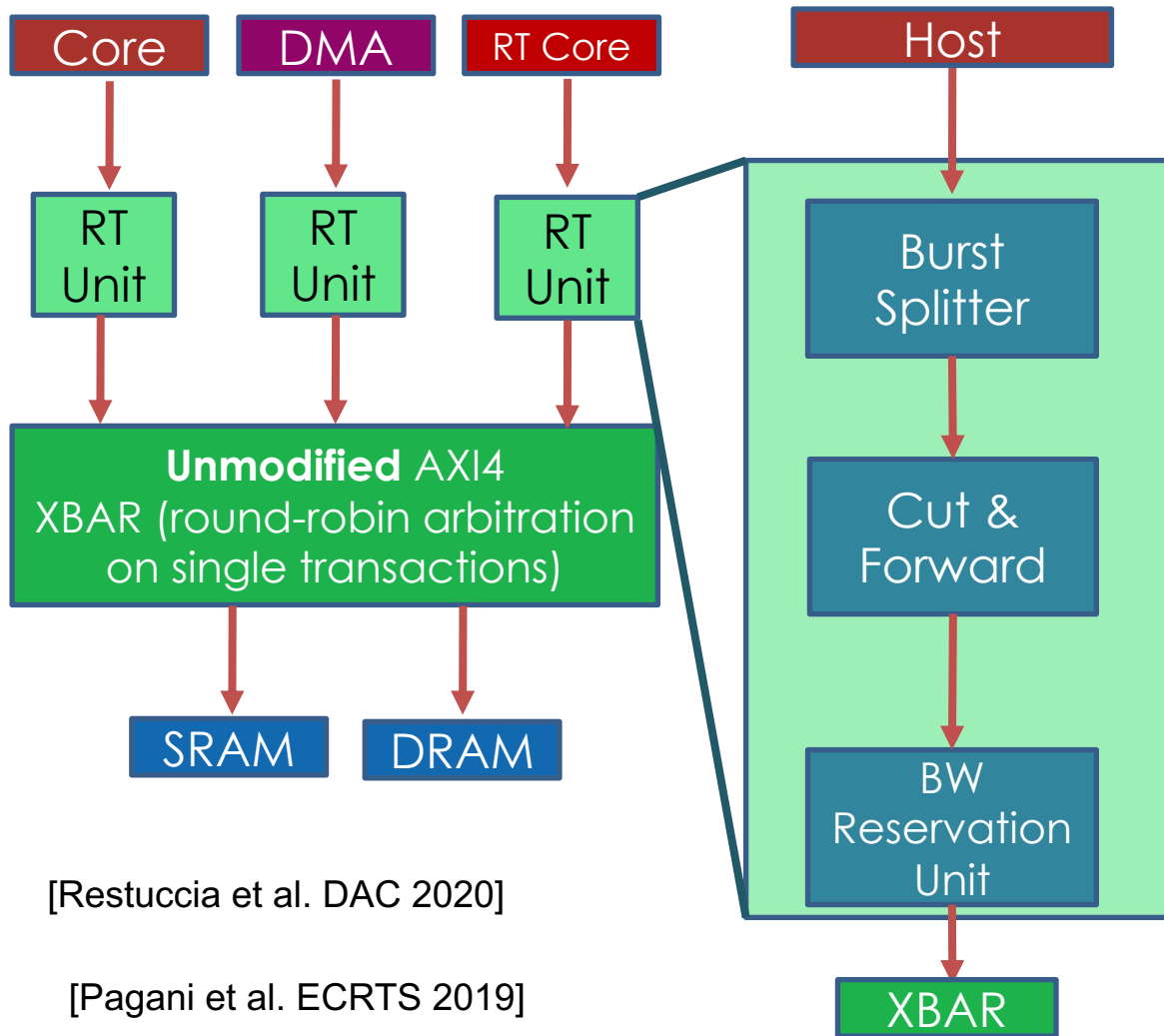
# The Safety Island



- Safety-critical applications running on a RTOS
- **Three CV32E40 cores** physically isolated operating in **lockstep** (single HART) and **fast HW/SW recovery** from faults
- **ECC protected scratchpad memories** for instructions and data
- **Fast and Flexible Interrupts Handling** through RISC-V compliant CLIC controller
- AXI-4 port for in/out communication



# Predictable On-Chip Communication (AXI RT)



[Restuccia et al. DAC 2020]

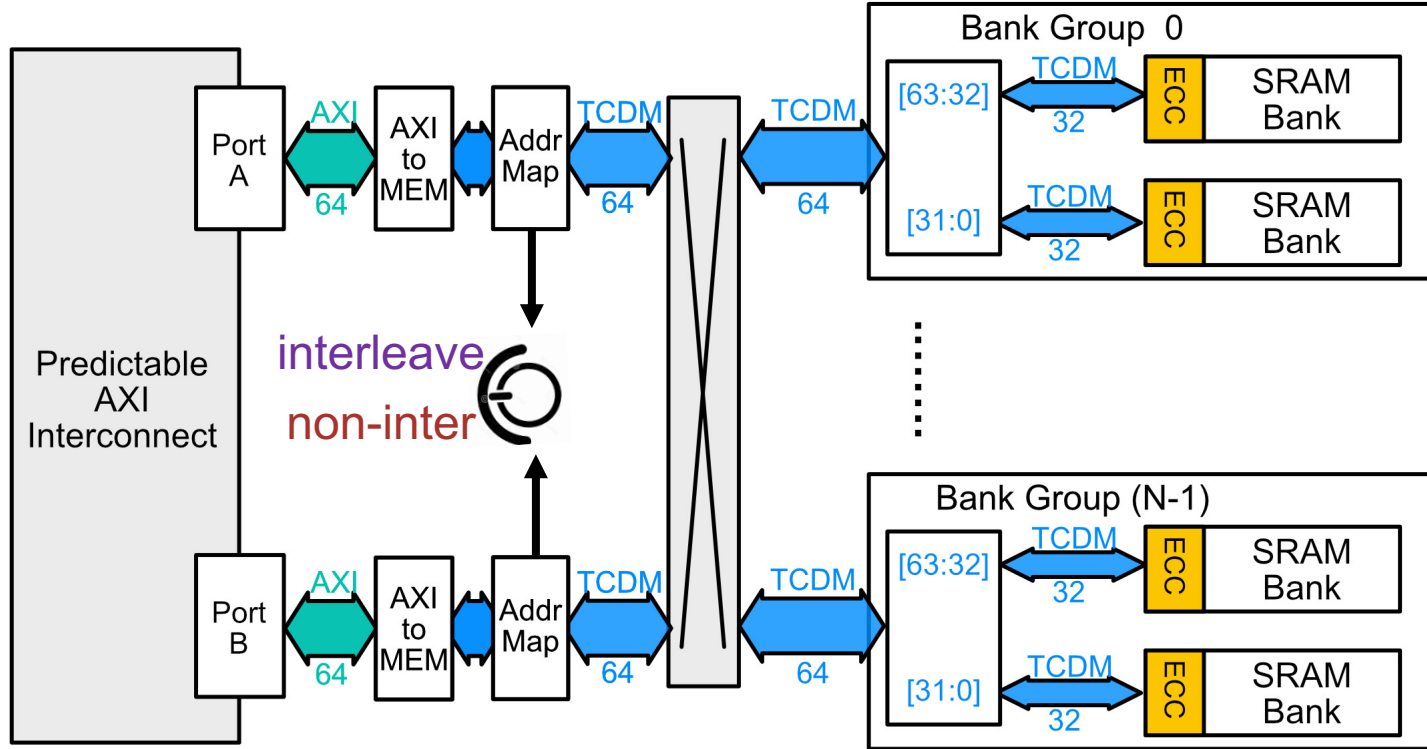
[Pagani et al. ECRTS 2019]

- AXI4 inherently **unpredictable**
- **Minimally Intrusive Solution**
  - No huge buffering, limited additional logic
  - **Solution verified in systematic worst-case real-time analysis**
- **AXI Burst Splitter**
  - **Equalizes length of transactions** to avoid unfair BW distribution in round-robin scheme
- **AXI Cut & Forward**
  - Configurable **chunking unit** to avoid long transaction delays influencing access time to the XBAR
- **AXI Bandwidth Reservation Unit**
  - Predictably enforces a given **max nr of transactions per time period** (to each master)
  - **Per-address-range credit-based** mechanism
  - Periodically **refreshed** (or by user)

# Contention-Free Shared L2 Scratchpad Memory

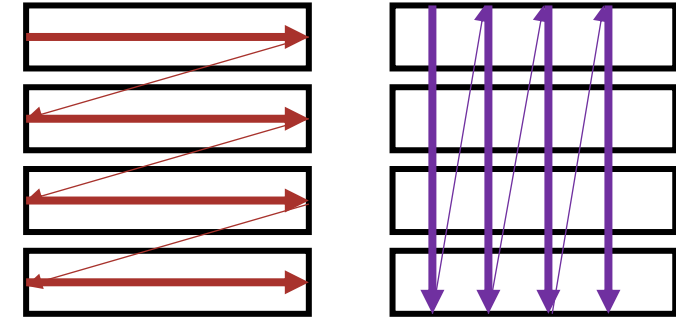
## 1. Dual-AXI-Port L2 Mem Subsystem

Multi-banked L2 SPM accessible from two different AXI ports



4. We determine in SW which port and which mode to use  
By using different address space!

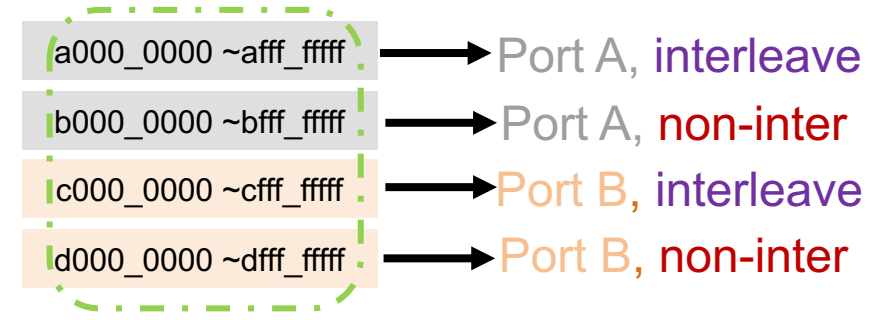
## 2. Two Address Mapping Modes



Non-interleaved

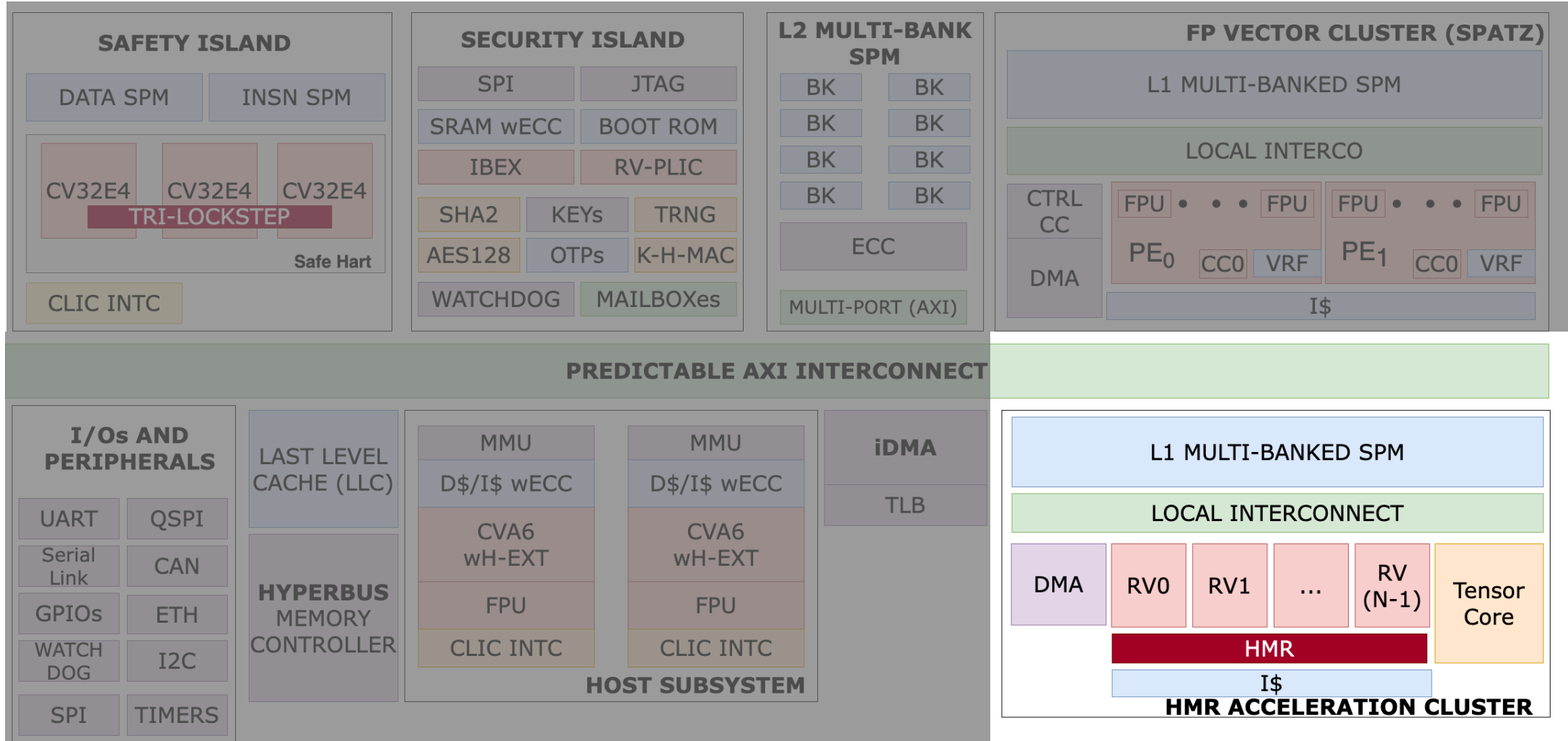
Interleaved

## 3. Dynamic Address Mapping by Address spaces, eg:

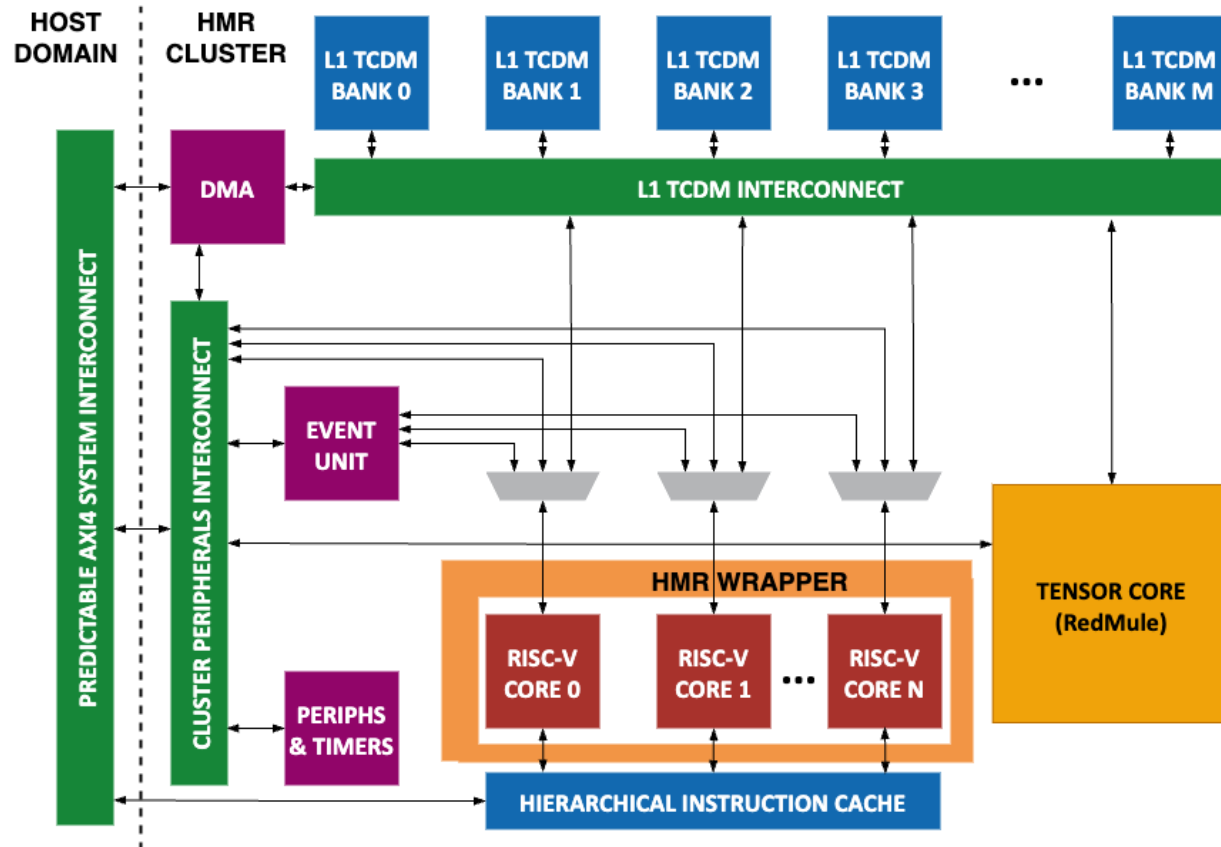


Point to the same L2 physical Mem space

# The HMR Acceleration Cluster



# The HMR Cluster for DNN-Oriented INT/FP Workloads



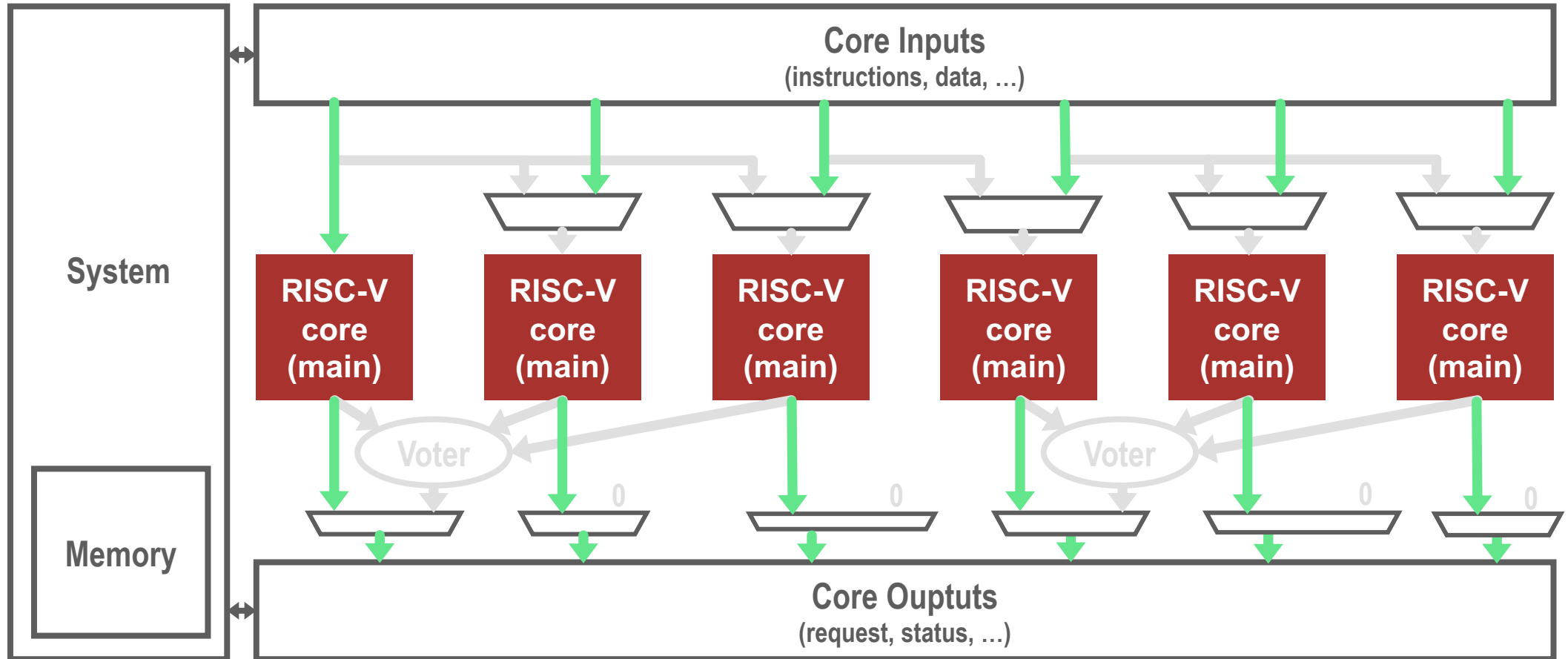
- 12x 32-bit RISC-V cores with support for DSP/QNN ISA Extensions
- Single-Cycle Multi-Banked Tightly-Coupled Data Memory (Scratchpad)
- Hardware Synchronizer
- DMA Controller for Explicit Memory Management
- L1-coupled **TensorCore** (RedMule)
- **Runtime-configurable Dual/Triple core redundancy mode** + hw/sw-based quick recovery mechanism

[Rogenmoser et al., arXiv, 2023]

[Tortorella et al., arXiv, 2023]

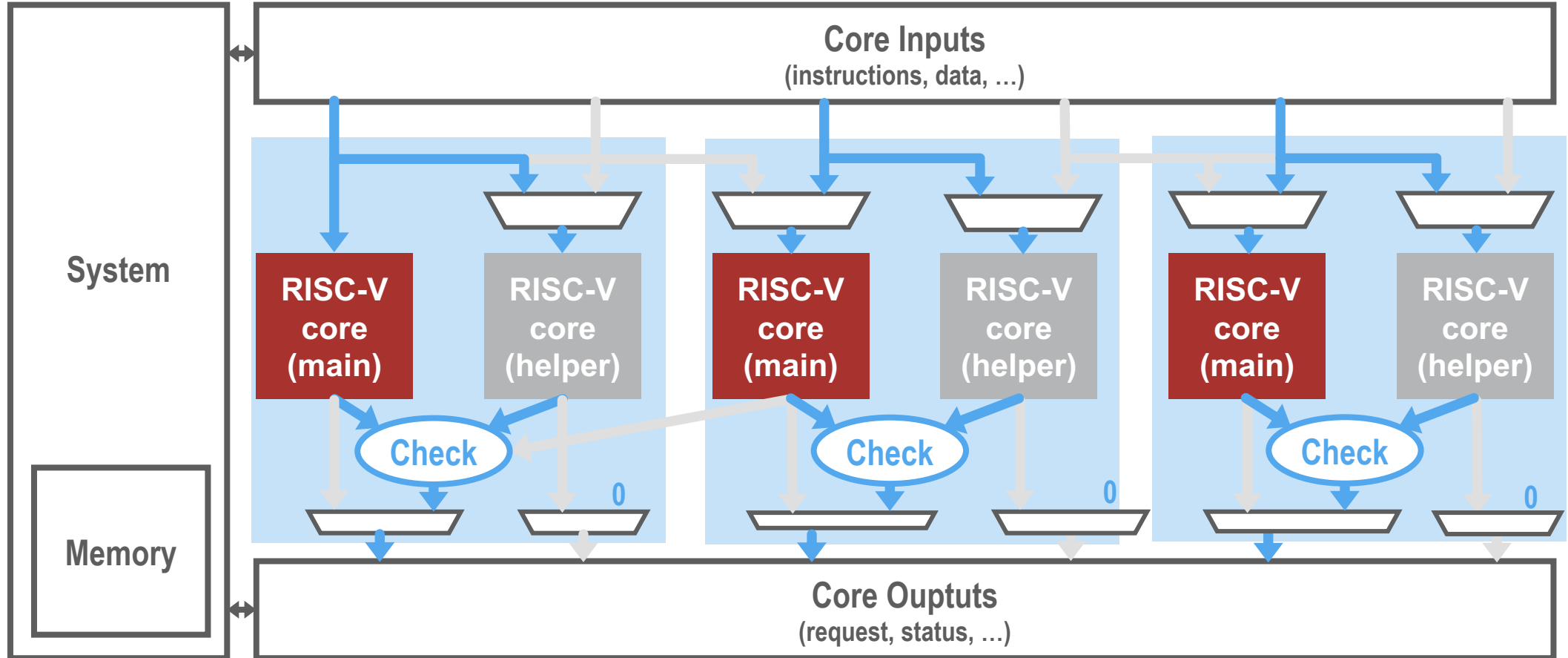
# Hybrid Modular Redundancy (HMR): Reconfigurable

**Independent Mode:** high performance, no reliability



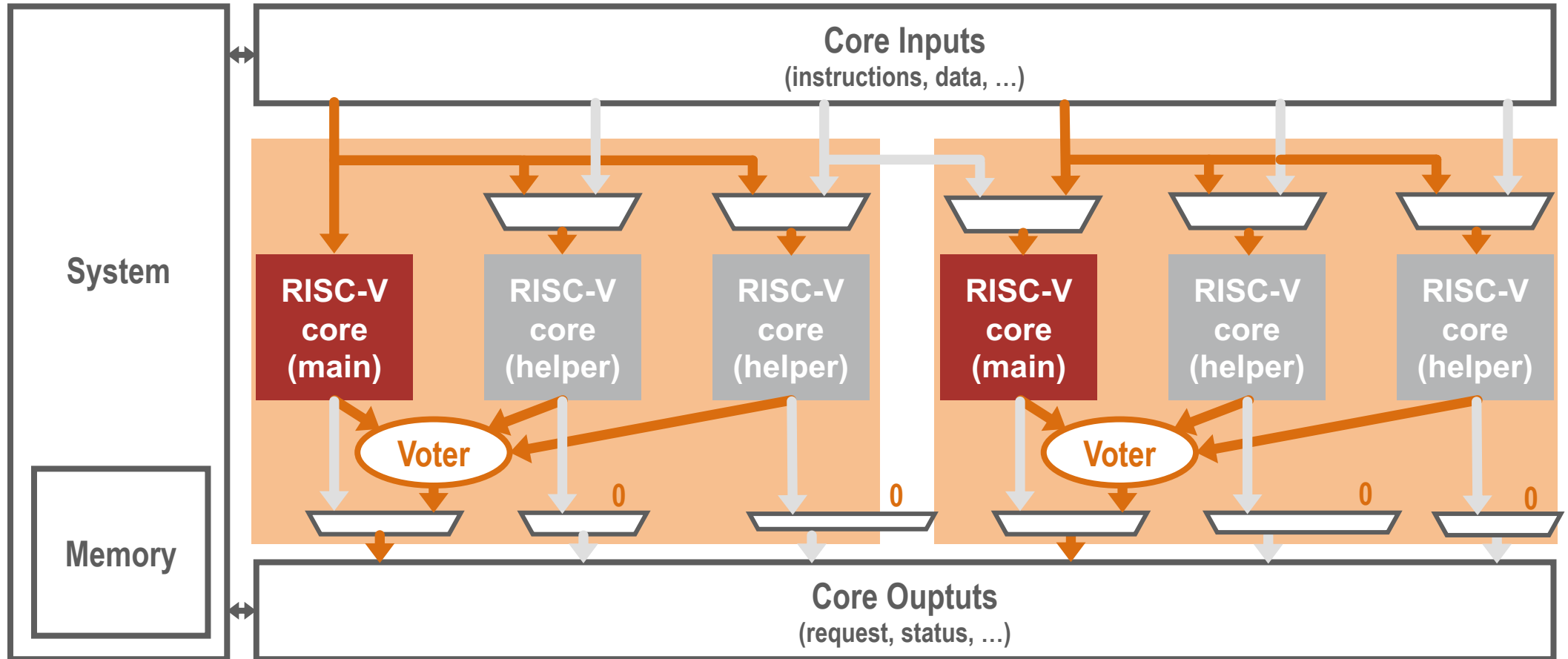
# Hybrid Modular Redundancy (HMR): Reconfigurable

**DMR Mode:** good performance, good reliability, slow recovery



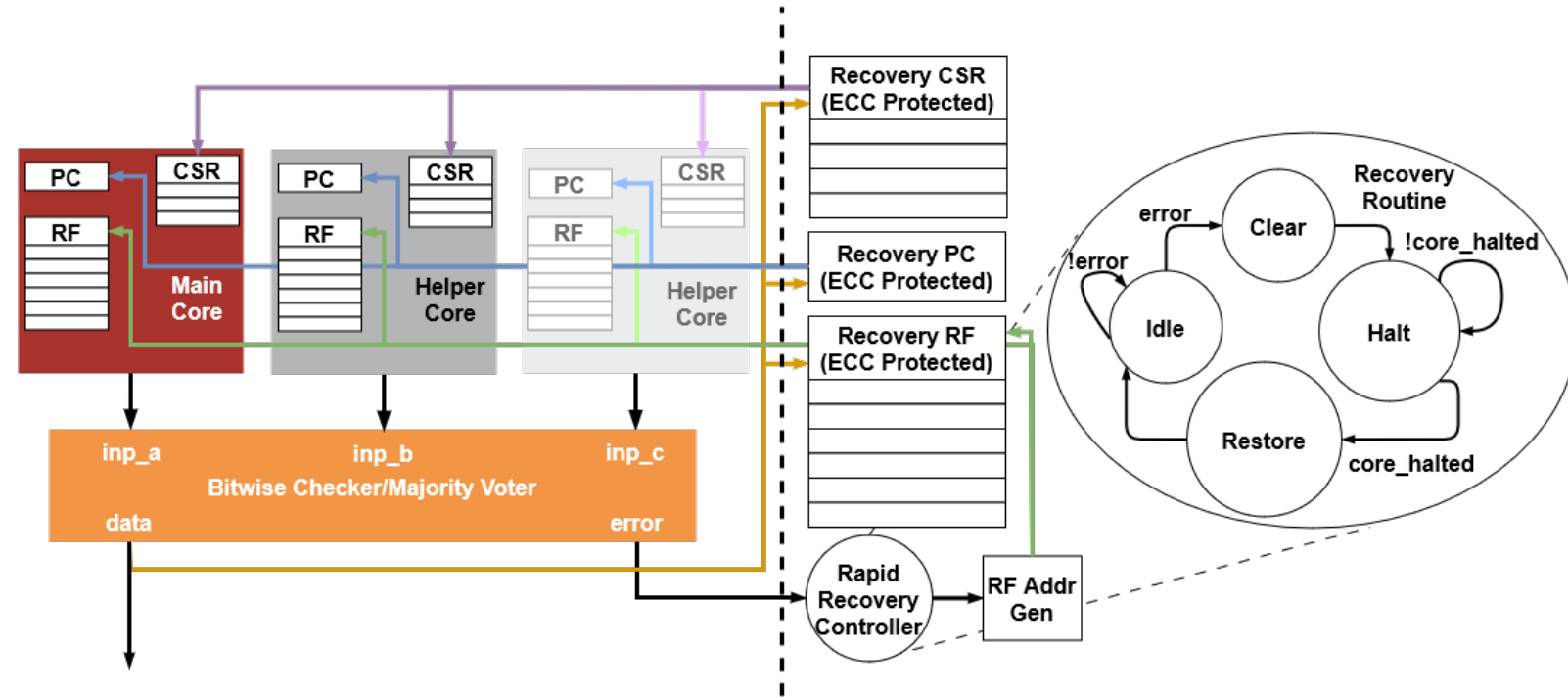
# Hybrid Modular Redundancy (HMR): Reconfigurable

**TMR Mode:** low performance, high reliability, quick recovery



# Rapid Recovery: shared hardware extension

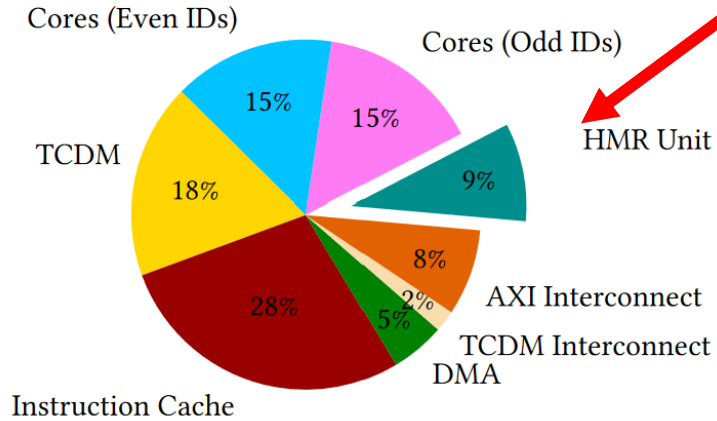
- Cycle-by-cycle backup of the cores state in ECC-protected Status Registers
- Quick recovery procedure (24 cycles!)
- Shared logic between TMR and DMR modes



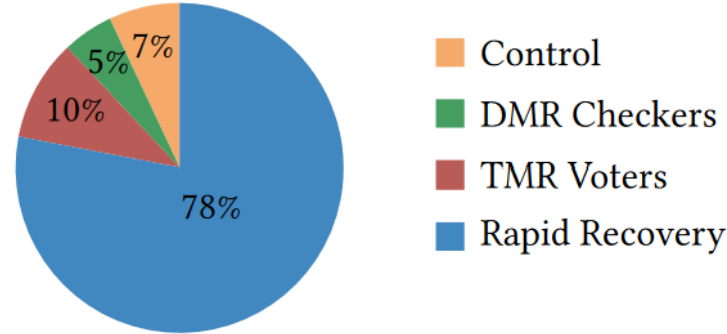


# HMR, yes... but at which cost?

Cluster Area breakdown with HMR Unit



HMR Unit Area Breakdown



Area Overhead of HMR Configurations

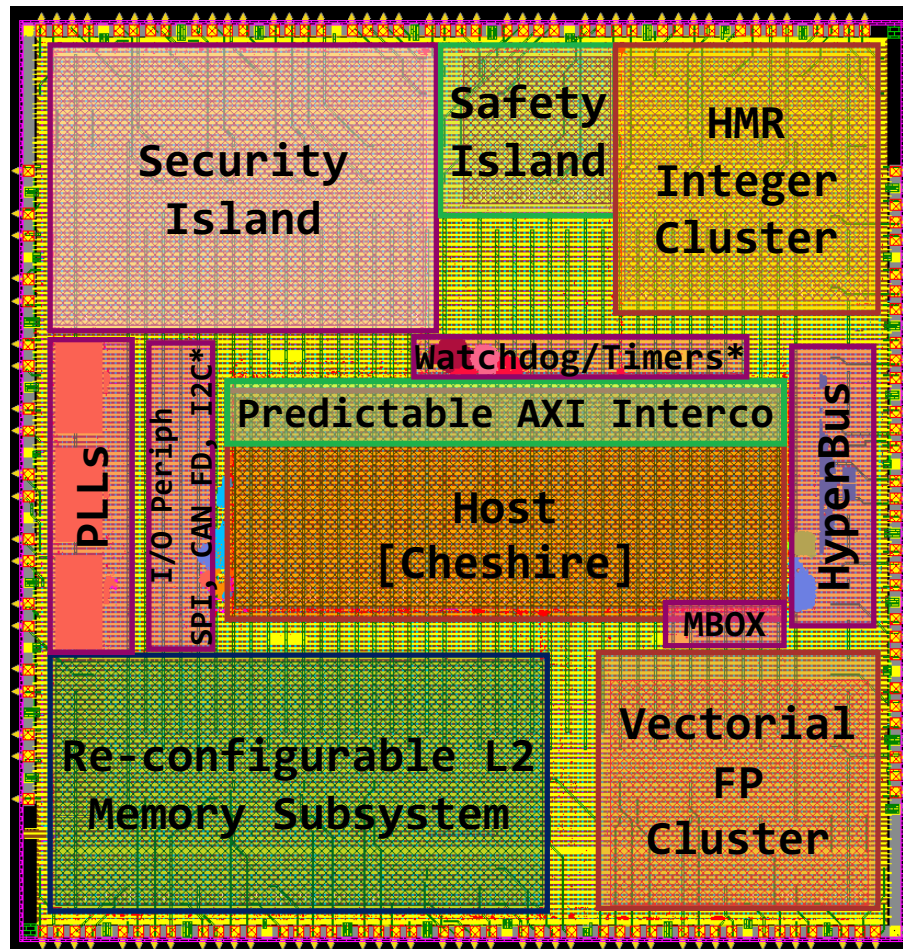
PULP Cluster Area [mm <sup>2</sup> ]	Overhead
Baseline	-
DMR	0.3%
TMR	0.7%
HMR	1.3%
With Rapid Recovery	
DMR	8.4%
TMR	8.8%
HMR	9.4%

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

[Rogenmoser et al., arXiv, 2023]

# Carfield SoC Flooplan – Taped out 11/2023



4 mm<sup>2</sup>

4 mm<sup>2</sup>

Modules marked with (\*) are not in scale

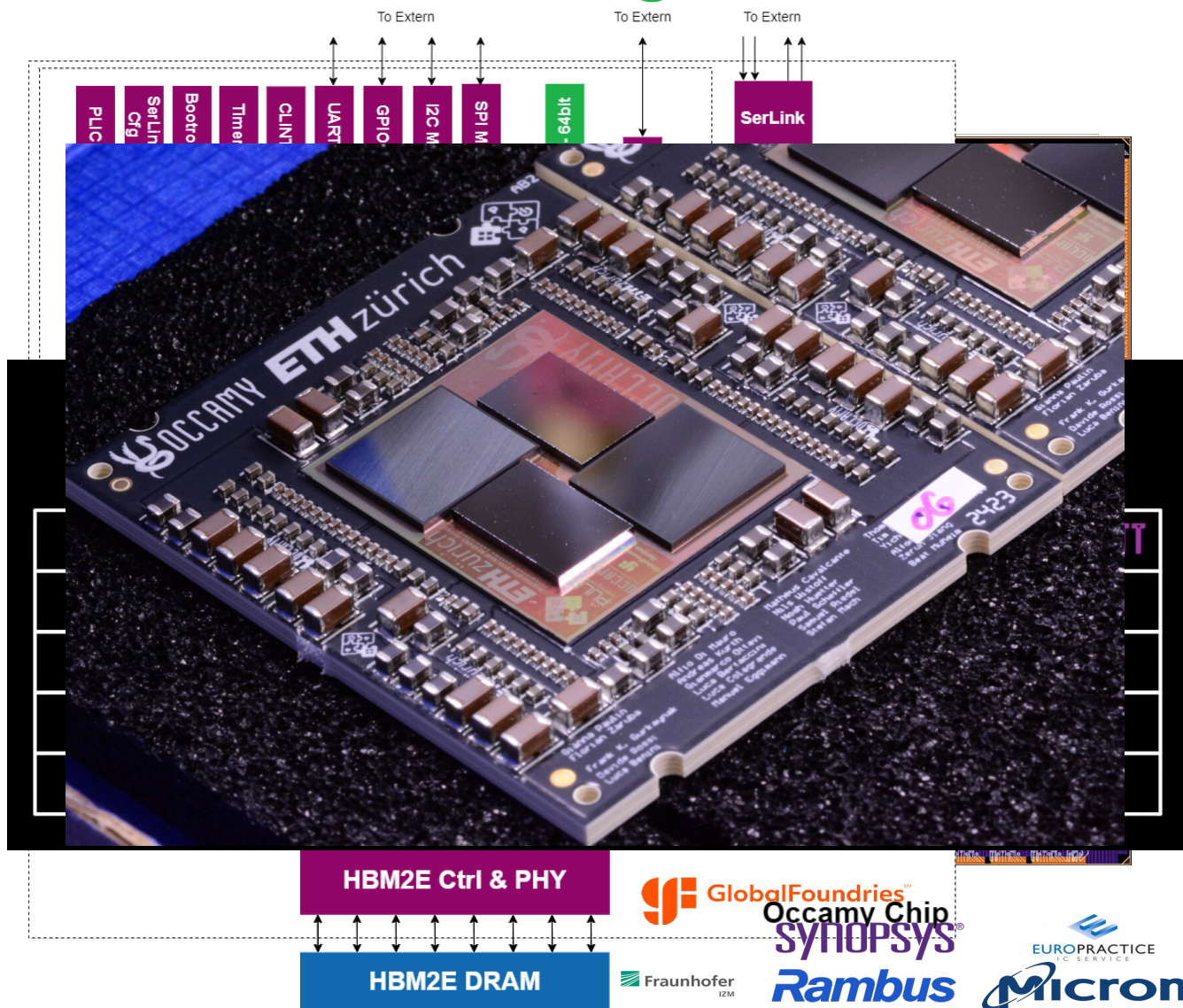


ETH zürich

- **Host [Cheshire]**
  - Dual-Core 64-bit RISC-V processor; **2.45 mm<sup>2</sup>**; 600 MHz;
- **Security Island**
  - Low-power secure monitor; **1.94 mm<sup>2</sup>** ; 100 MHz;
- **Safety Island**
  - **0.42 mm<sup>2</sup>**; 500 MHz
- **Re-configurable L2 Memory Subsystem**
  - 1MB; **2.33 mm<sup>2</sup>**; 500 MHz
- **HMR Integer Cluster**
  - **1.17 mm<sup>2</sup>**; 500 MHz;
- **Vectorial FP Cluster**
  - **1.14 mm<sup>2</sup>**; 600 MHz;
- **Hyperbus**
  - 2 PHY, 2 Chips; 200 MHz; Max BW **400 MB/s**

Frequency bound by RAMs (limited availability in Intel offering for Universities)

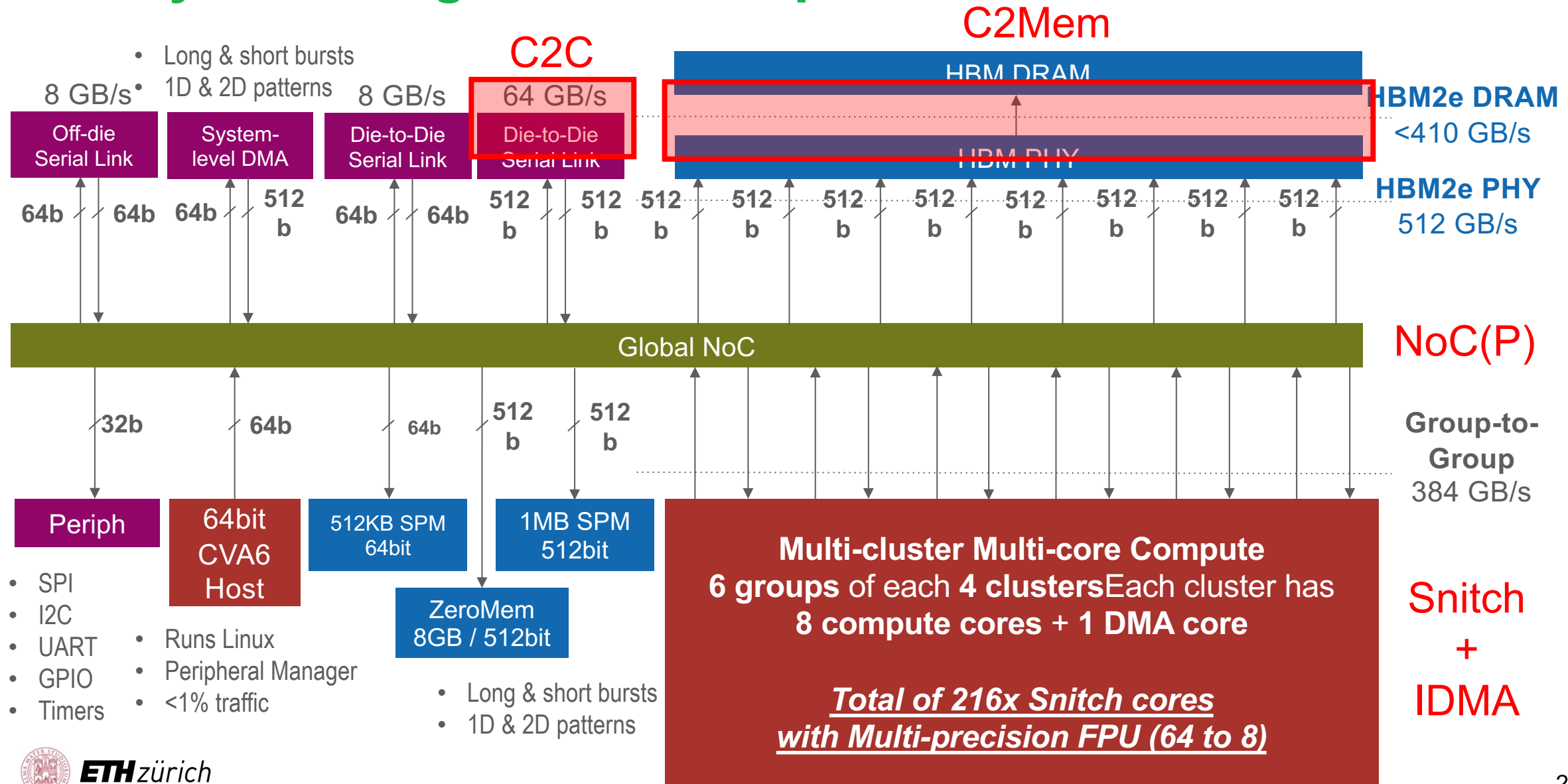
# Toward Self-Driving Cars



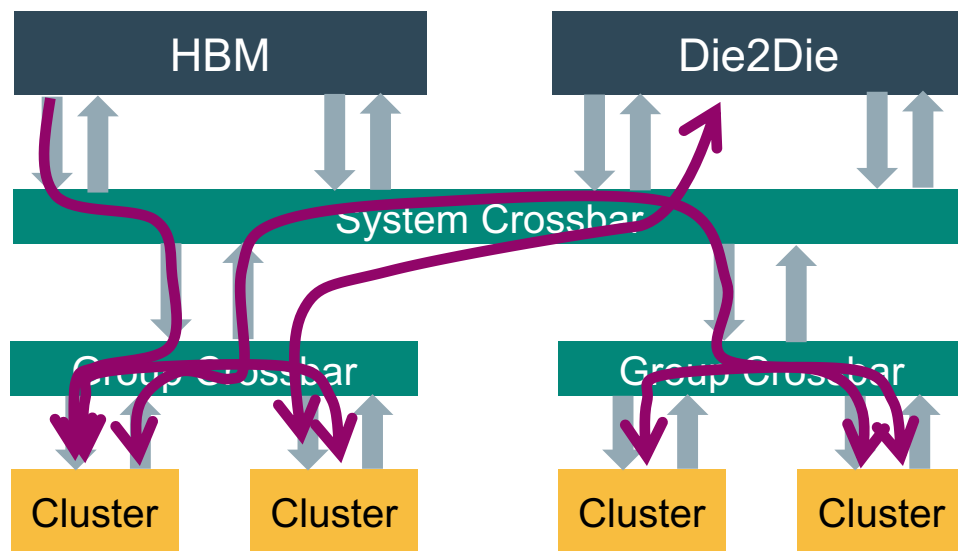
- GF12, target **1GHz** (typ)
- 2 AXI NoCs (multi-hierarchy)
  - 64-bit
  - 512-bit with “interleaved” mode
- Peripherals
- Linux-capable manager core CVA6
- 6 Quadrants: 216 cores/chiplet
  - 4 cluster / quadrant:
    - 8 compute +1 DMA core / cluster
    - 1 multi-format FPU / core (FP64,x2 32, x4 16/alt, x8 8/alt)
- 8-channel HBM2e (8GB) **512GB/s**
- D2D link (Wide, Narrow) **70+2GB/s**
- System-level DMA
- SPM (2MB wide, 512KB narrow)

**Peak 384 GDPflop/s per chiplet**

# Occamy: RISC-V goes HPC Chiplet!



# NoC(P): Efficient and Flexible Data Movement



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

- Access energy
- Congestion
- High latency

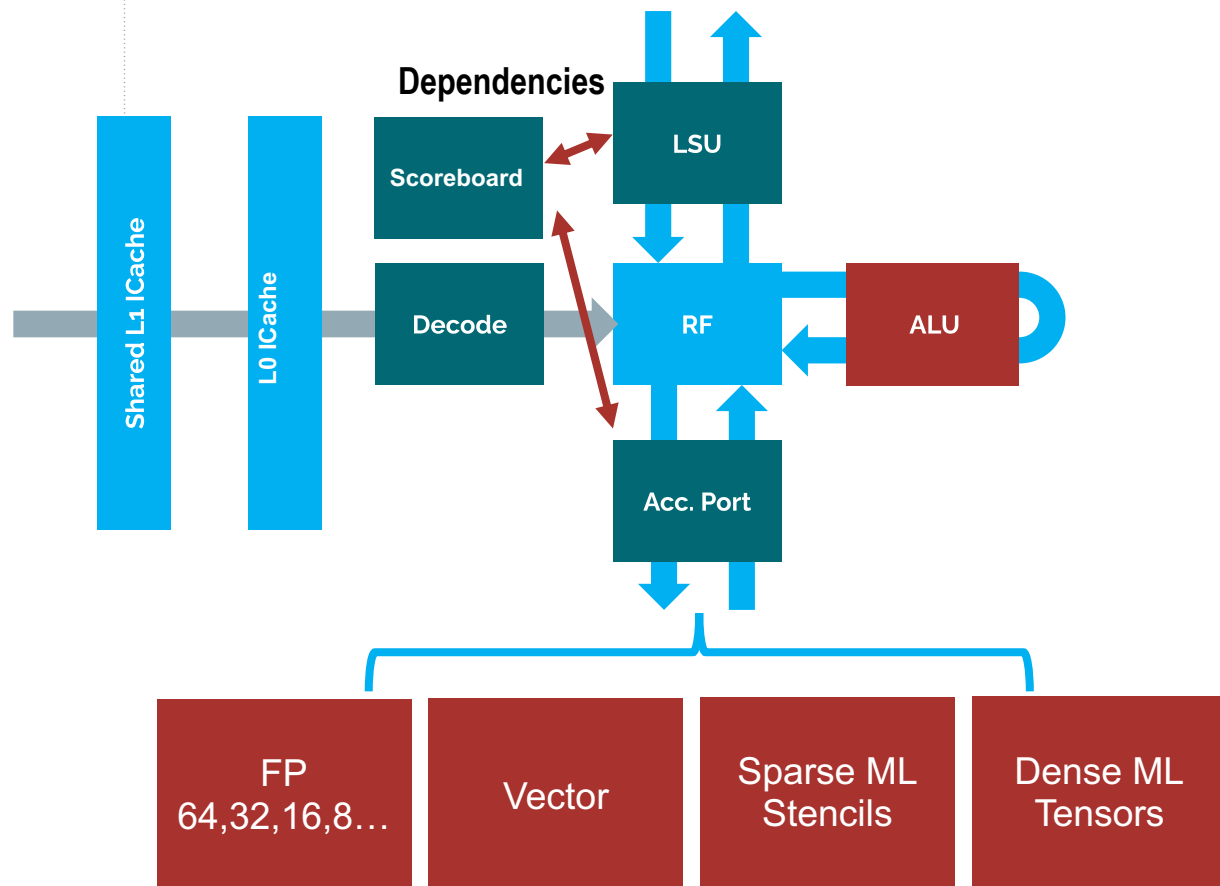
Instead reuse data on lower levels of the memory hierarchy

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

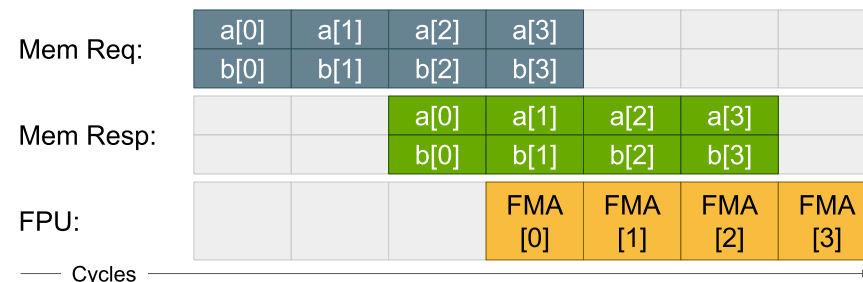
Smartly distribute workload

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

# Snitch – Latency-Tolerant, Efficient, Extensible



- **Snitch** core: around 20KGE
  - Speed via simplicity (1GHZ+)
  - L0 Icache/buffer for low energy fetch
  - Parametric # of LD/ST ports in LSU (1-4)
- **Extensible** → “Accelerator” port
  - Minimal baseline ISA (RISC-V)
  - Extensibility: Performance through ISA extensions (via accelerator port)
- **Latency-tolerant** → Scoreboard
  - Tracks instruction dependencies
  - Much simpler than OOO support!



# IDMA: Efficient *Explicit* Global Data Mover

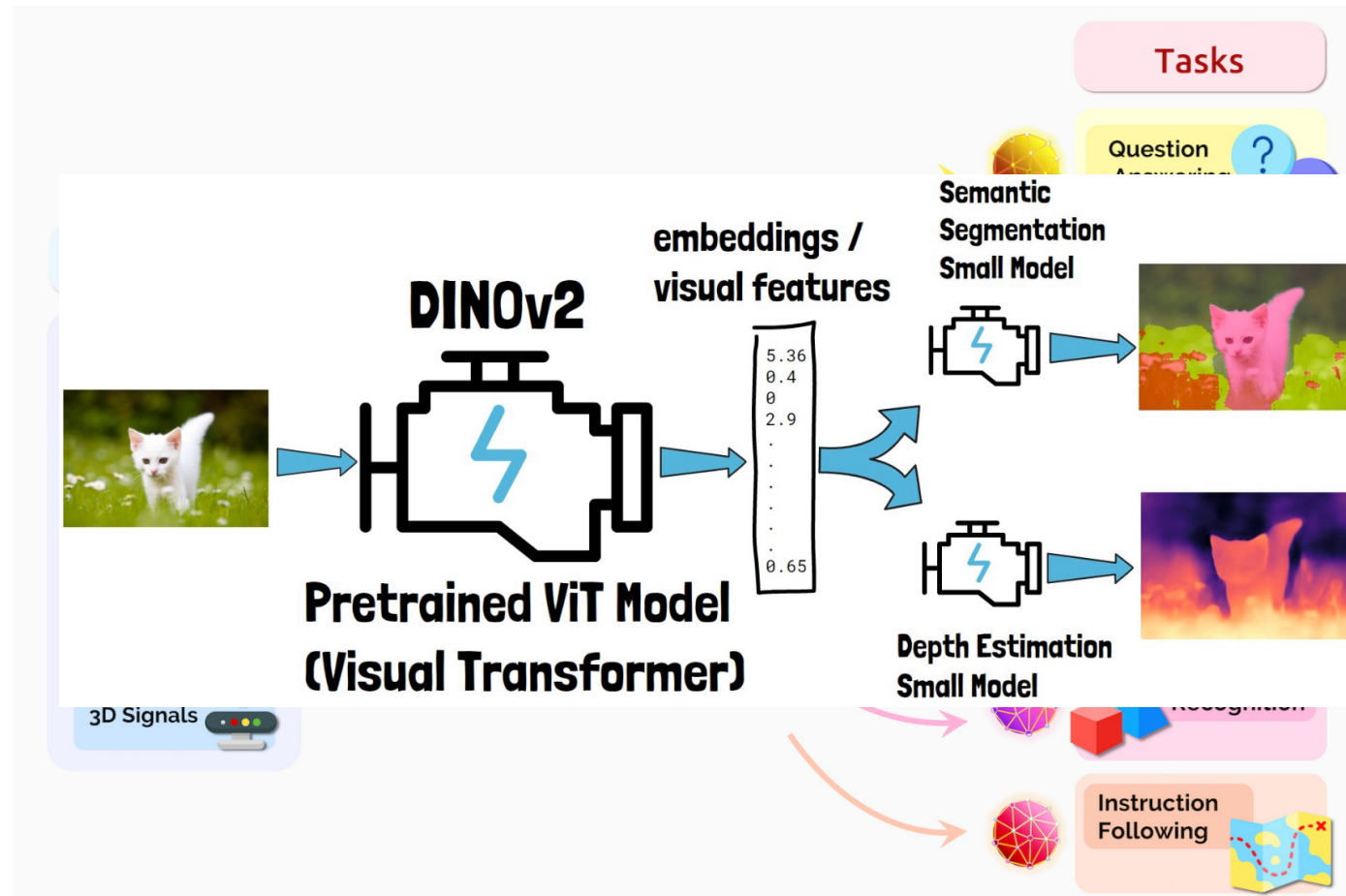


- 512-bit AXI DMA – double-buffered transfers
- Tightly coupled with Snitch (<10 cycles configuration)
- Operates on wide 512-bit data-bus
- Hardware support to copy 2-4-dim shapes
- Higher-dimensionality handled by SW
- Intrinsic/library for easy programming
- **Sparse data support**

```
// setup and start a 1D transfer, return transfer ID
uint32_t __builtin_sdma_start_oned(
    uint64_t src, uint64_t dst, uint32_t size, uint32_t cfg);
// setup and start a 2D transfer, return transfer ID
uint32_t __builtin_sdma_start_twod(
    uint64_t src, uint64_t dst, uint32_t size,
    uint32_t sstrd, uint32_t dstrd, uint32_t nreps, uint32_t cfg);
// return status of transfer ID tid
uint32_t __builtin_sdma_stat(uint32_t tid);
// wait for DMA to be idle (no transfers ongoing)
void __builtin_sdma_wait_for_idle(void);
```

# What's Next? The era of Foundation Models

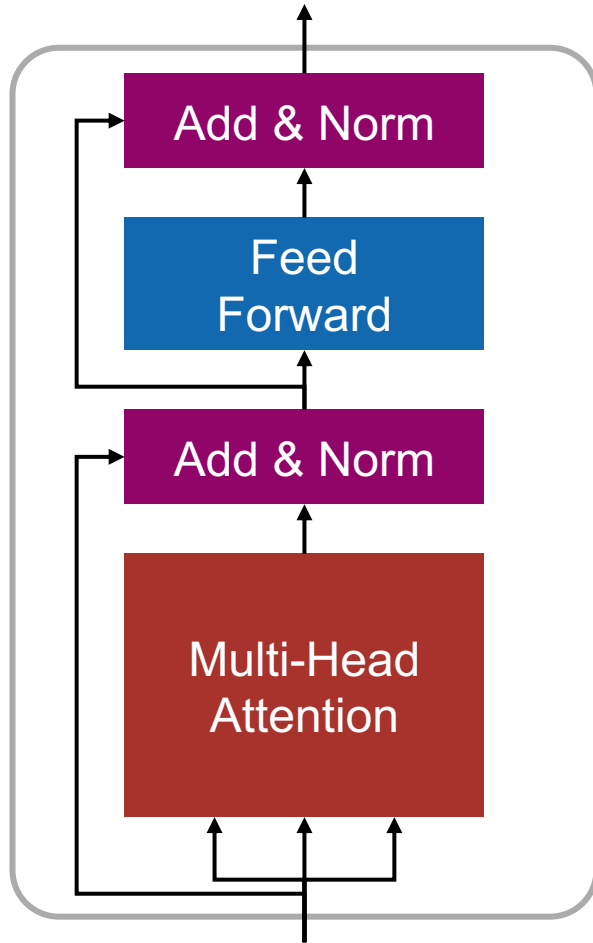
- Versatility and Multi-modality
  - Natural language processing, computer vision, robotics, biology, ...
- Homogenization of models
  - **Transformers as foundation models**
- Self-supervision, Fine-tuning
  - Self-supervised training on large-scale unlabeled dataset
  - Fine-tune (few layers) on specific tasks with smaller labeled datasets.
- Zero-shot specialization
  - Prompt engineering for new tasks



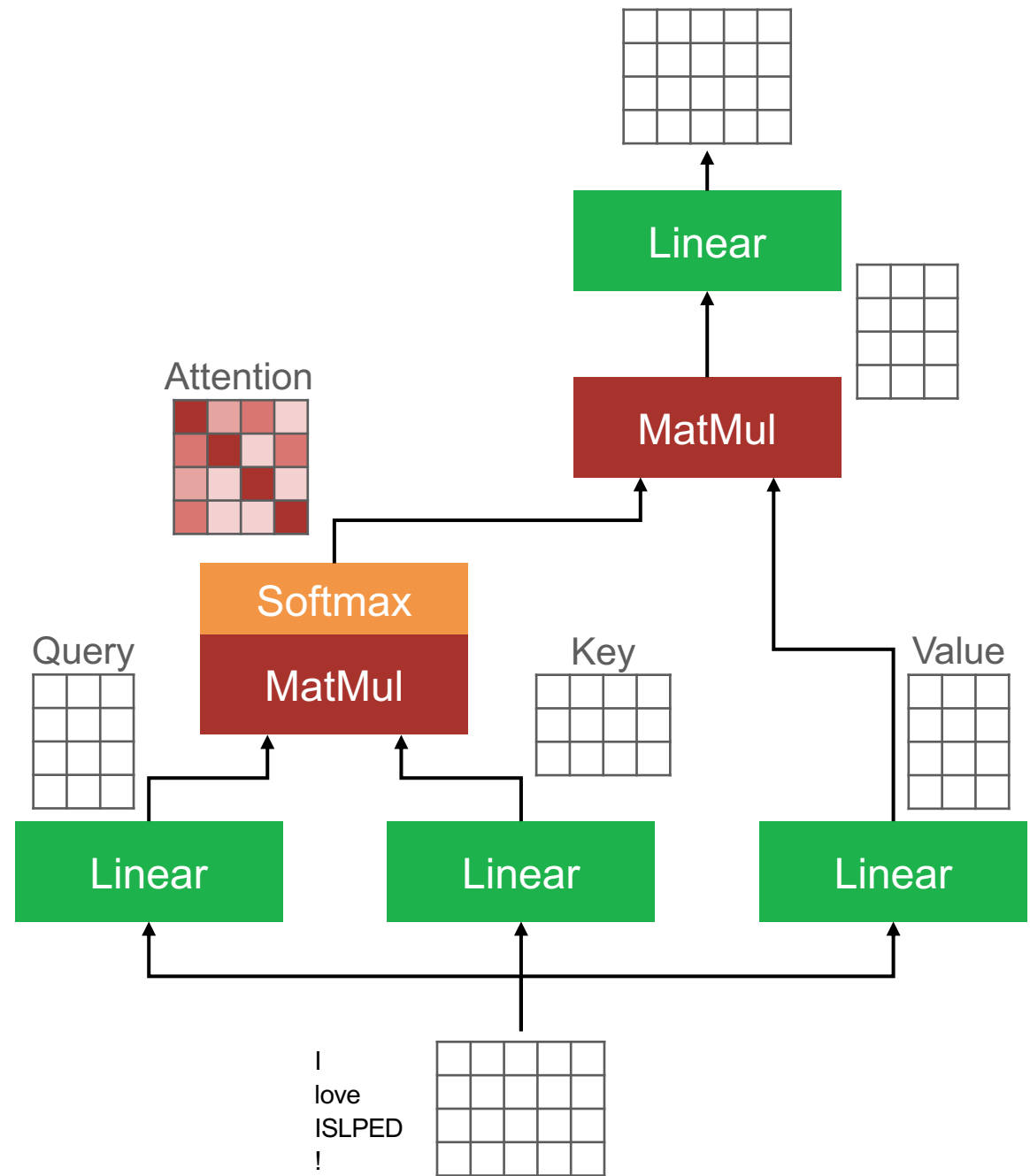
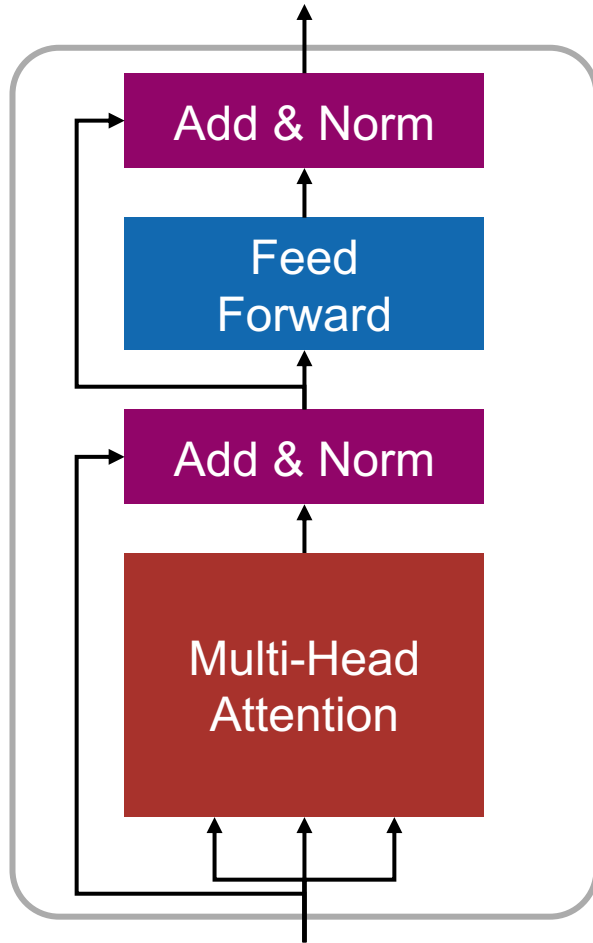
Bommasani, Rishi, et al. "On the Opportunities and Risks of Foundation Models." *Center for Research on Foundation Models (CRFM), Stanford Institute for Human-Centered Artificial Intelligence (HAI)*.



# Attention is all you need!

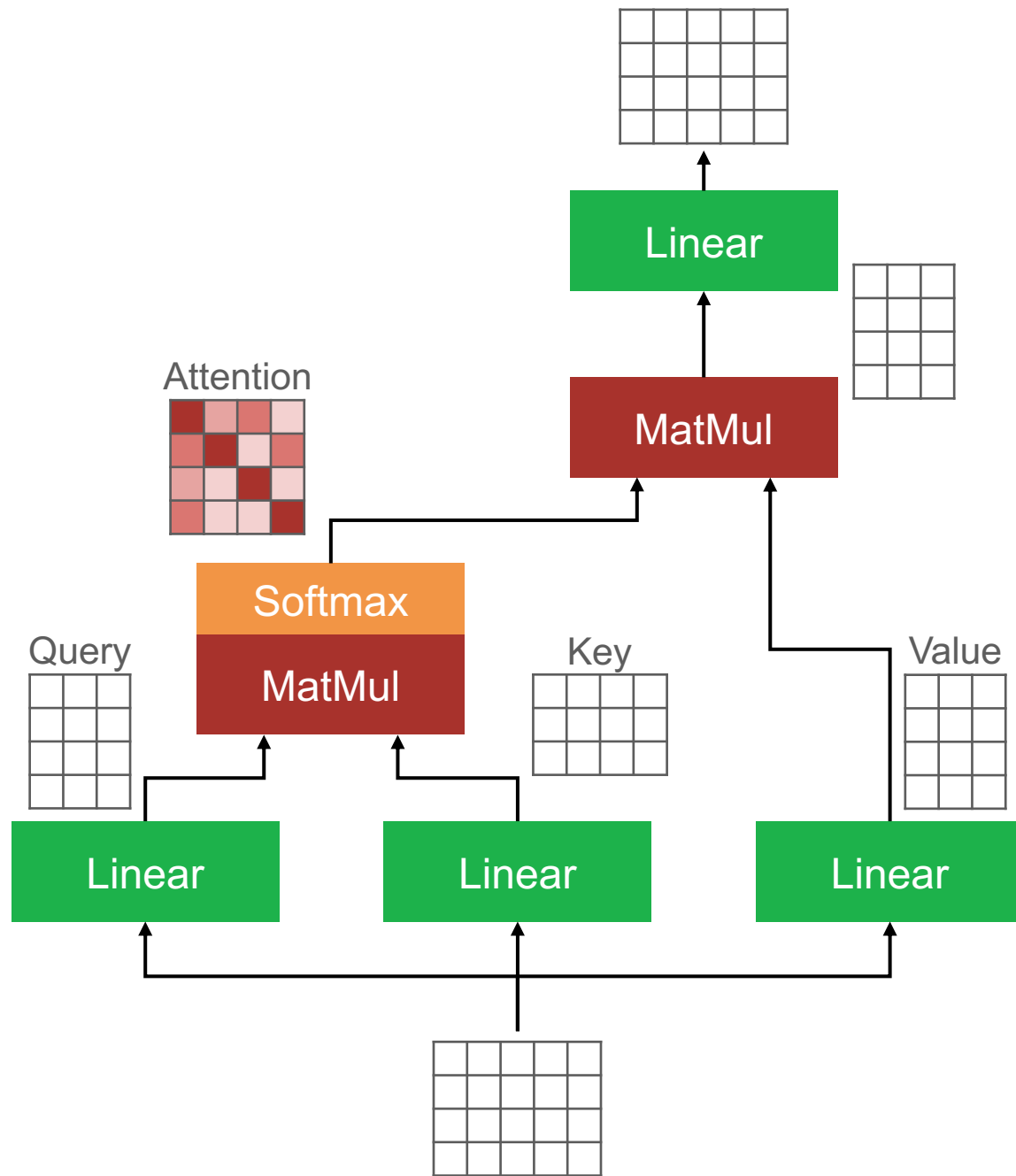


# Attention but how?



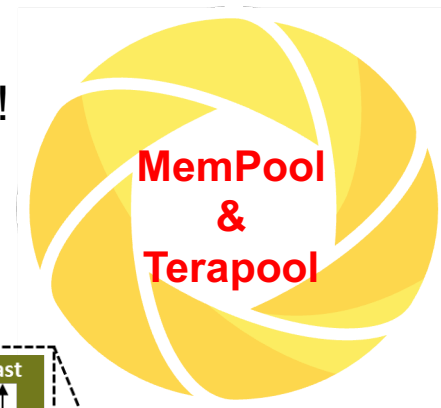
# Challenges in Attention

- Attention matrix is a square matrix of order input length.
  - Computational complexity
  - Memory requirements
- MatMul & Softmax dominate



# Matmul Benefits from Large(r) Shared-L1 clusters

- Why?
  - Better global latency tolerance if  $L1_{size} > 2 * L2_{latency} * L2_{bandwidth}$  (Little's law + double buffer)
  - Smaller data partitioning overhead
  - Larger Compute/Boundary bandwidth ratio:  $N^3/N^2$  for MMUL grows linearly with N!

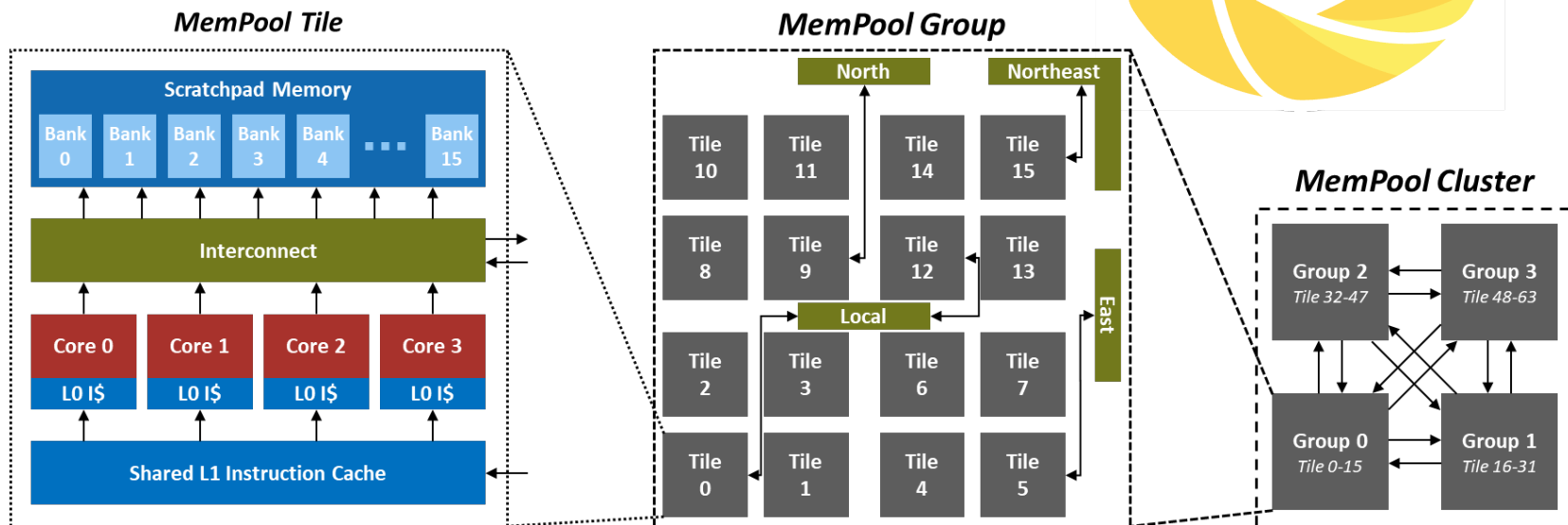


- A large “MemPool”

- 256+ cores
- 1+ MiB of shared L1 data memory
- $\leq 10$  cycle latency (Snitch can handle it)

- Physical-aware design

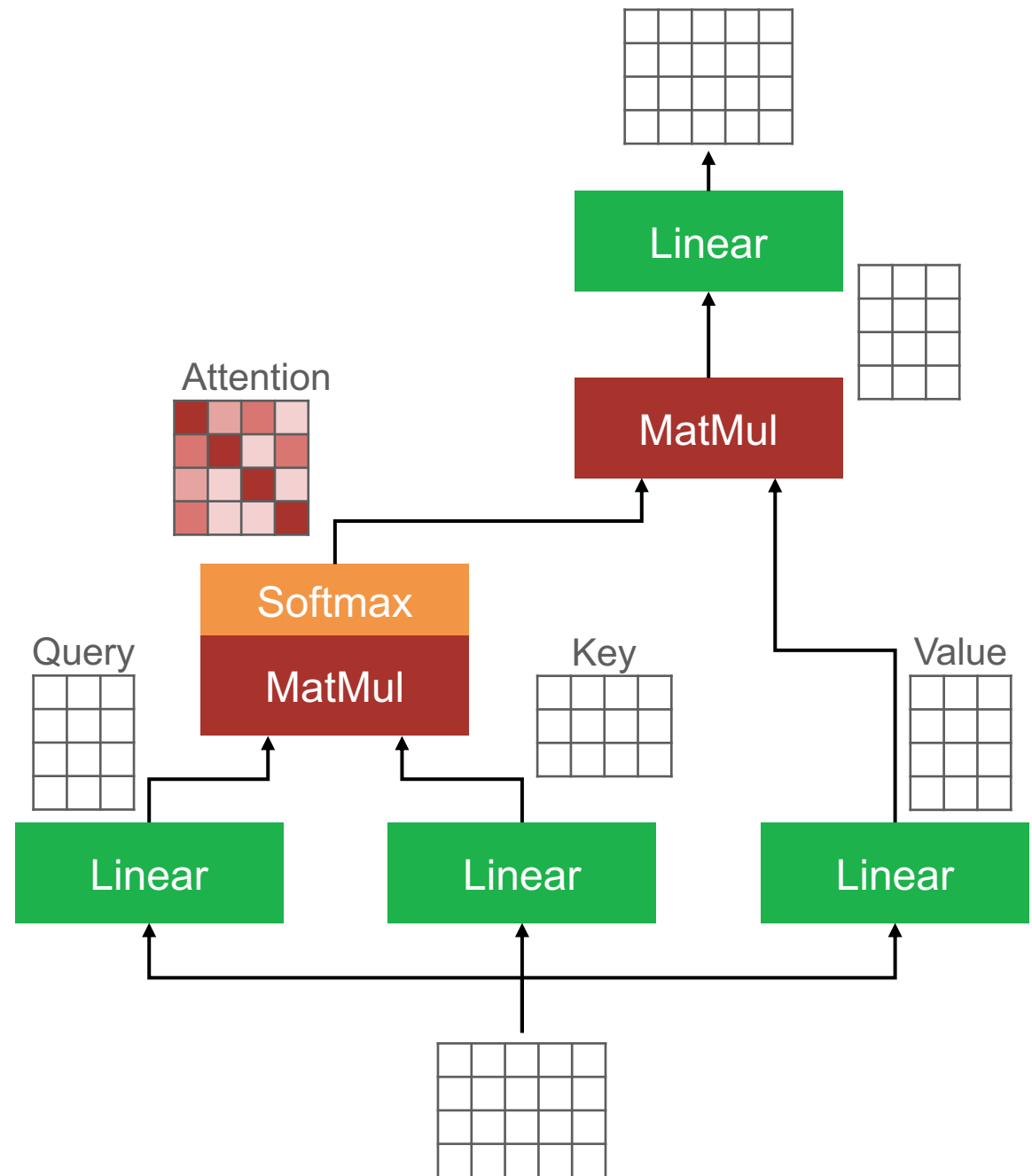
- WC Frequency  $> 700$ +Mhz
- Targeting iso-frequency with small cluster



**Butterfly Multi-stage Interconnect 0.3req/core/cycle, 5 cycles**

# Challenges in Attention

- **Attention matrix is a square matrix of order input length.**
  - Computational complexity
  - Memory requirements
- **Every attention layer applies *Softmax* to attention matrix!**

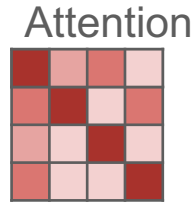


# Challenges in *Attention*

- **Attention matrix is a square matrix of order input length.**
  - Computational complexity
  - Memory requirements

- **Every attention layer applies *Softmax* to attention matrix!**

- 3 passes over a row.
- Quantization is problematic.



Softmax

$$\text{Softmax}(\mathbf{x})_i = \frac{e^{x_i - \max(\mathbf{x})}}{\sum_j^n e^{x_j - \max(\mathbf{x})}}$$

# ITA: Integer Transformer Accelerator



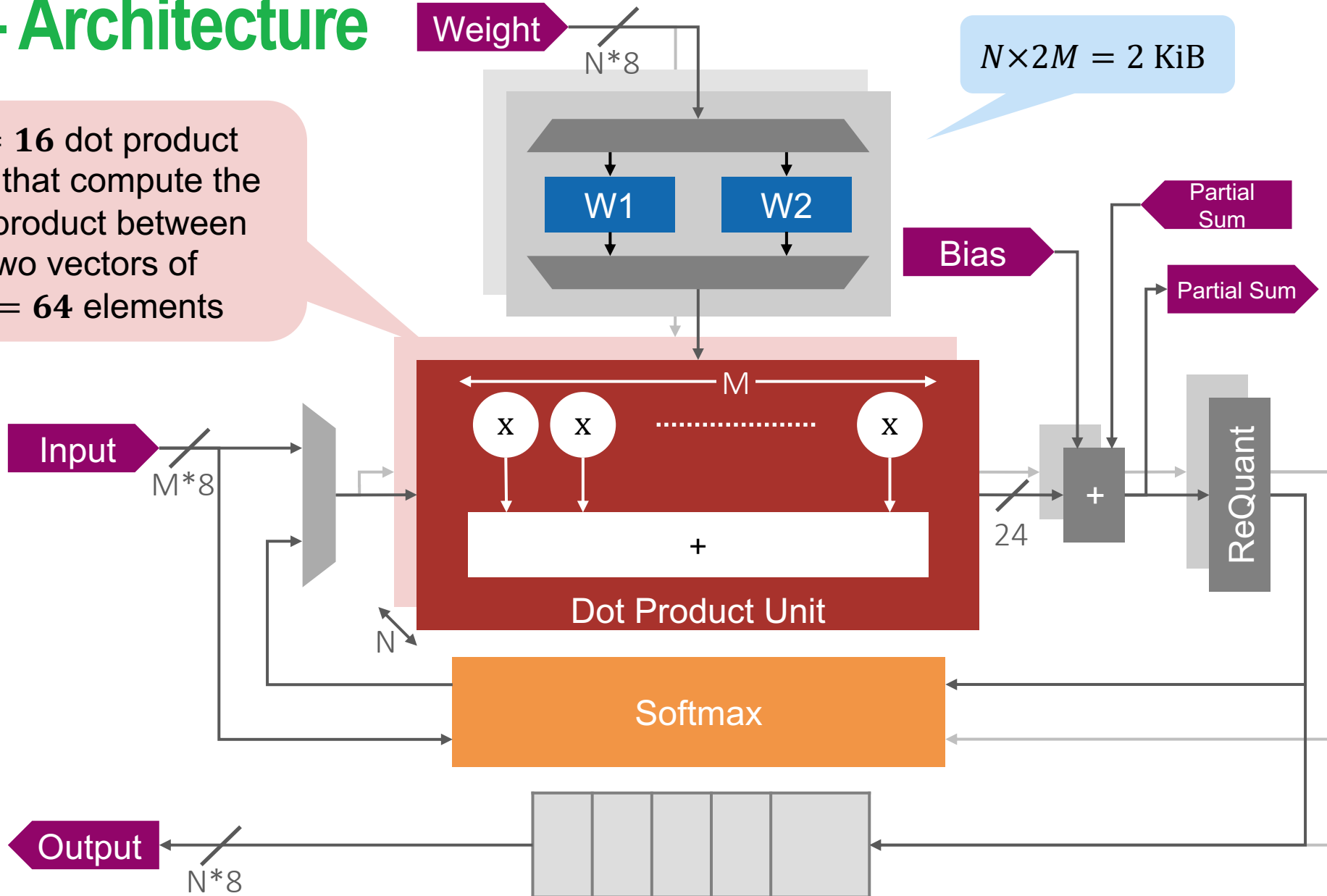
- **Attention** accelerator for transformers!
- INT8 quantized networks
- Output stationary - Local weight stationary
  - Spatial input reuse
  - Spatial output partial sum reuse
- Fused  $Q.K^T$  and  $A.V$  computation
- Special *Softmax* unit!



[Islamoglu et al. ISLPED23]

# ITA – Architecture

$N = 16$  dot product units that compute the dot product between two vectors of  $M = 64$  elements





# Hardware-friendly Softmax

$$\text{Softmax}(\mathbf{x})_i = \frac{e^{x_i - \max(\mathbf{x})}}{\sum_j^n e^{x_j - \max(\mathbf{x})}}$$

Softmax

# Hardware-friendly Softmax

$$\text{Softmax}(\mathbf{x})_i = \frac{e^{x_i - \max(\mathbf{x})}}{\sum_j^n e^{x_j - \max(\mathbf{x})}}$$

$$\text{Softmax}(\mathbf{x})_i = \frac{1}{\sum_j^n 2^{(x_{qj} - \max(\mathbf{x}_q)) \gg 5}} 2^{(x_{qi} - \max(\mathbf{x}_q)) \gg 5}$$

Softmax

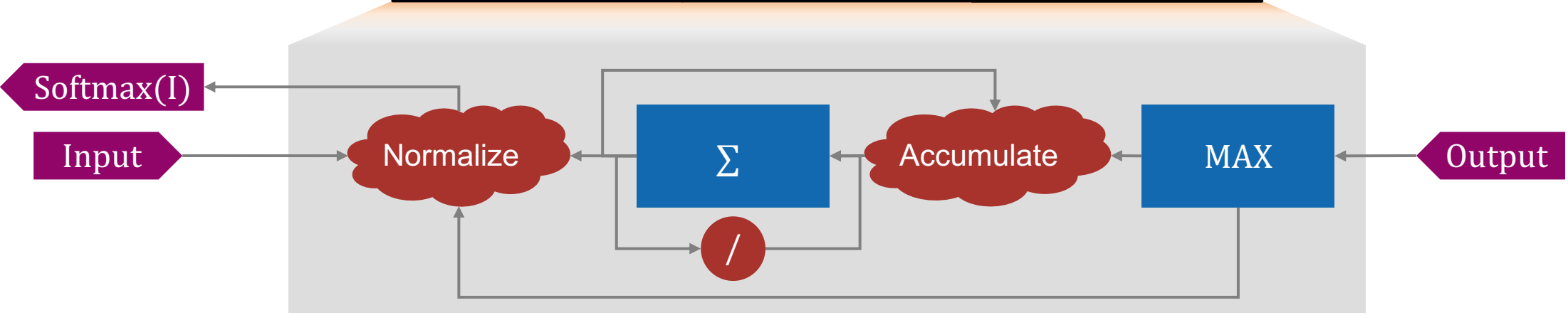
Directly operates on quantized values.

No exponentiation modules and multipliers.

Computes softmax on streaming data.

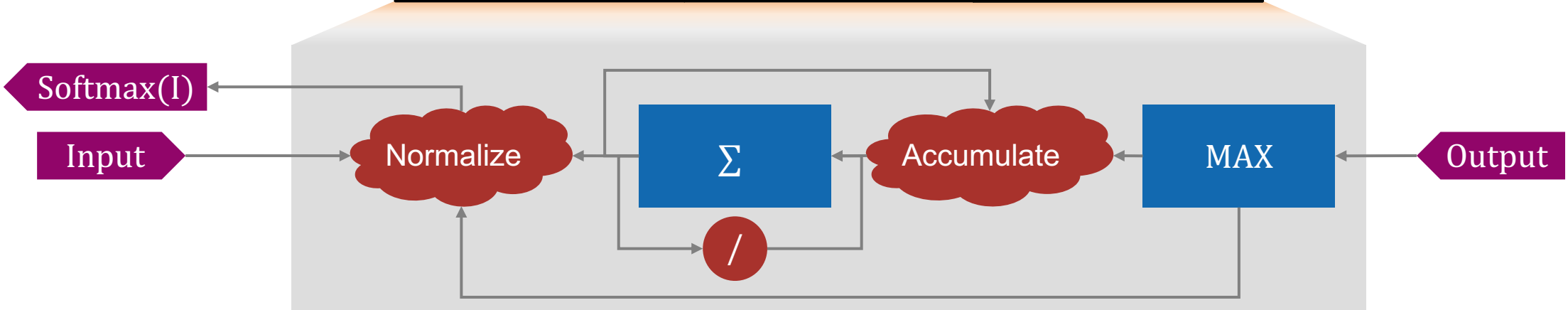
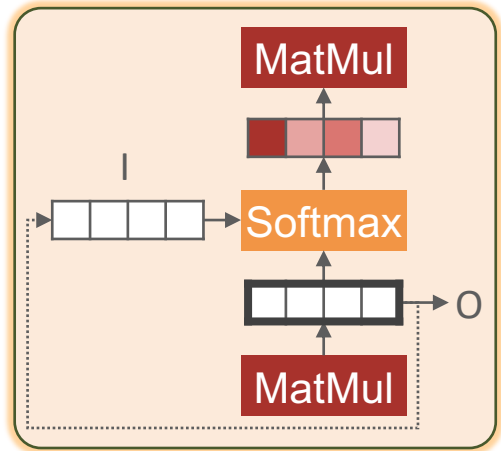
# Hardware-friendly *Softmax*

$$\text{Softmax}(\mathbf{x})_i = \frac{1}{\sum_j^n 2^{(x_{qj} - \max(\mathbf{x}_q)) \gg 5}} 2^{(x_{qi} - \max(\mathbf{x}_q)) \gg 5}$$

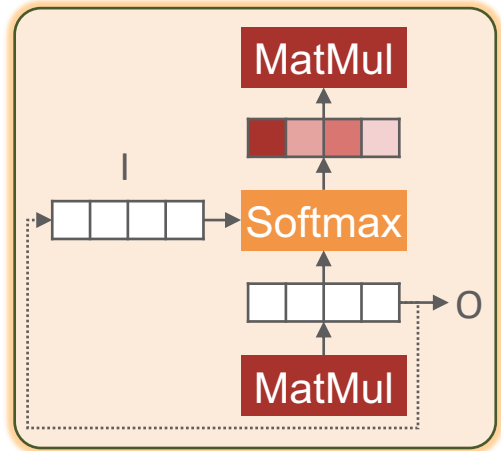


# Hardware-friendly Softmax

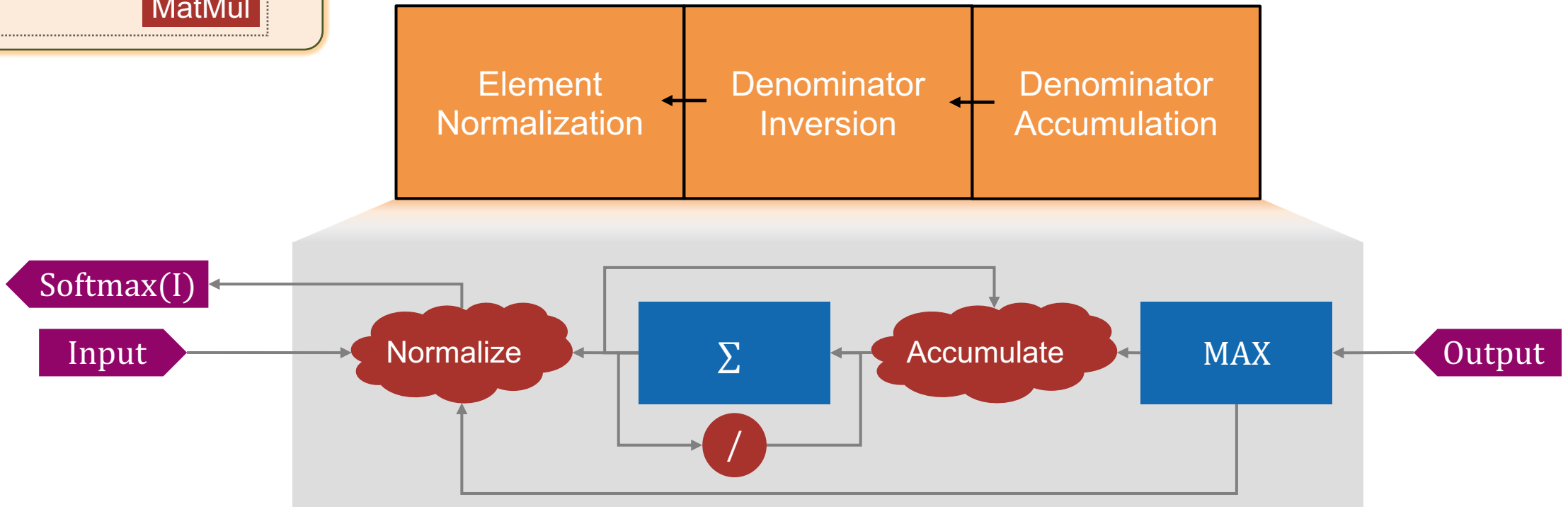
$$\text{Softmax}(\mathbf{x})_i = \frac{1}{\sum_j^n 2^{(x_{qj} - \max(\mathbf{x}_q)) \gg 5}} 2^{(x_{qi} - \max(\mathbf{x}_q)) \gg 5}$$



# Hardware-friendly Softmax

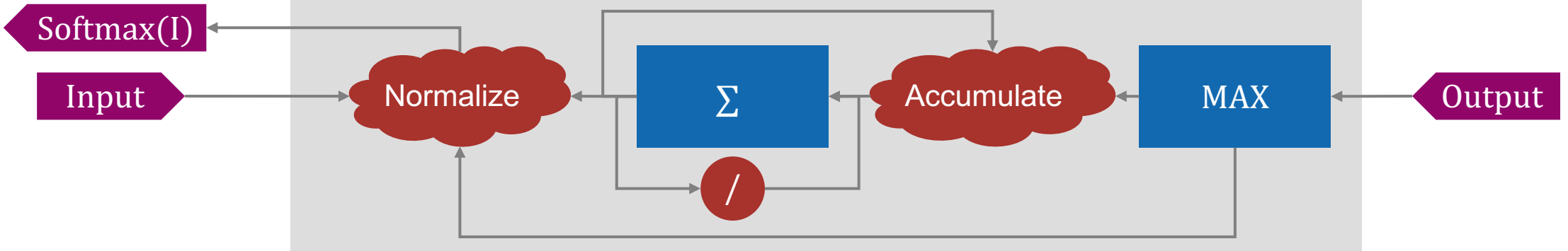
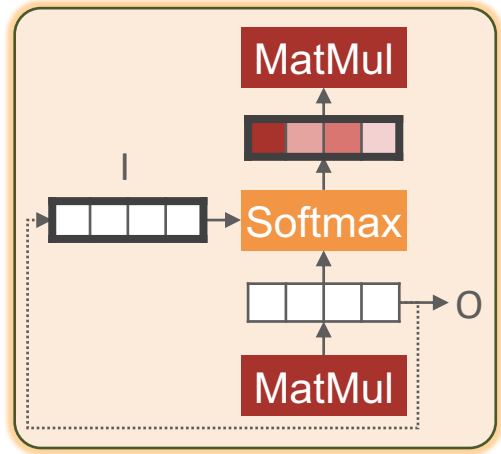


$$\text{Softmax}(\mathbf{x})_i = \frac{1}{\sum_j^n 2^{(x_{qj} - \max(\mathbf{x}_q)) \gg 5}} 2^{(x_{qi} - \max(\mathbf{x}_q)) \gg 5}$$



# Hardware-friendly Softmax

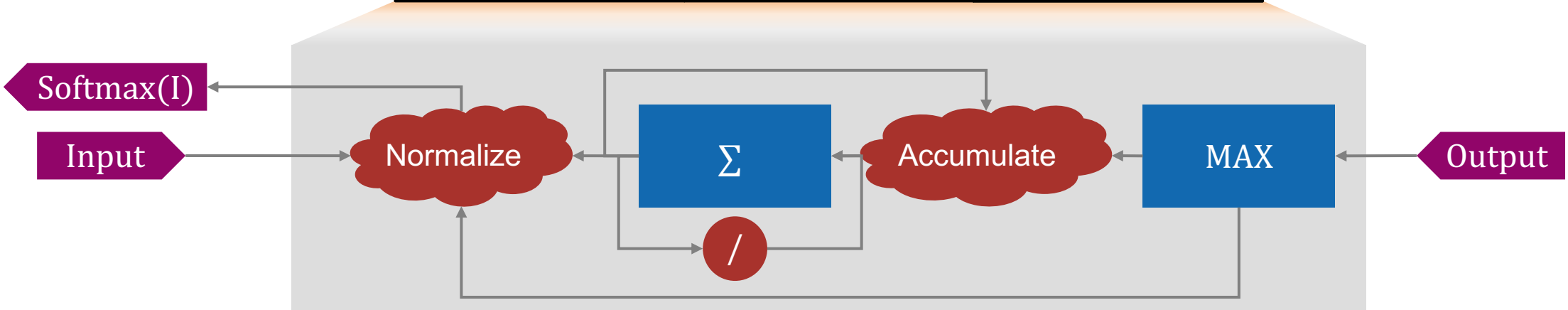
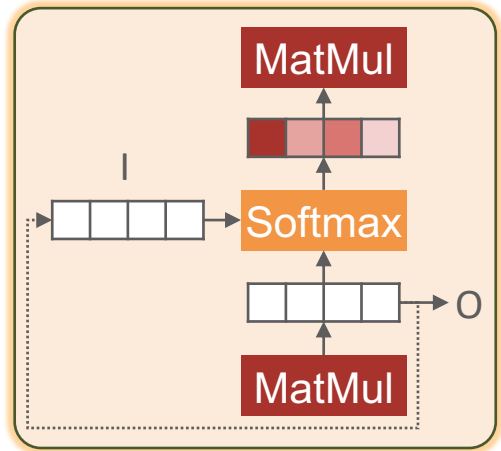
$$\text{Softmax}(\mathbf{x})_i = \frac{1}{\sum_j^n 2^{(x_{qj} - \max(\mathbf{x}_q)) \gg 5}} 2^{(x_{qi} - \max(\mathbf{x}_q)) \gg 5}$$



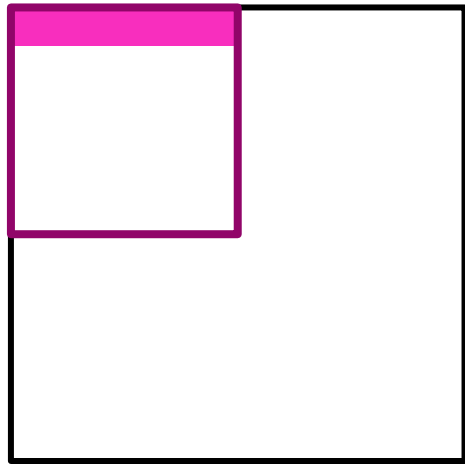
# Hardware-friendly Softmax

*MAE = 0.46%*

$$\text{Softmax}(\mathbf{x})_i = \frac{1}{\sum_j^n 2^{(x_{qj} - \max(\mathbf{x}_q)) \gg 5}} 2^{(x_{qi} - \max(\mathbf{x}_q)) \gg 5}$$

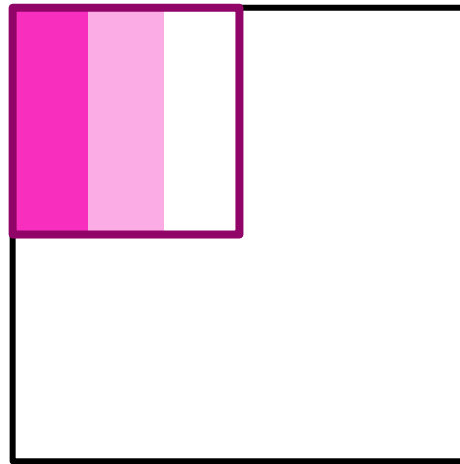


# Output stationary - Local weight stationary



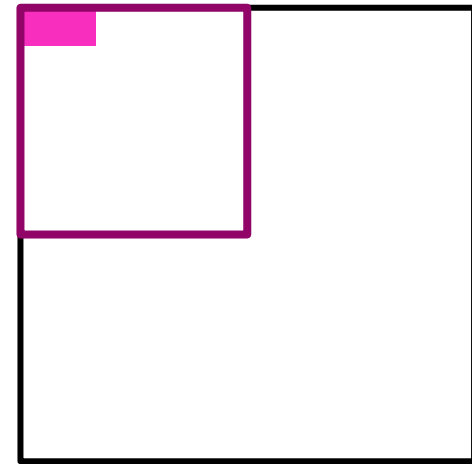
Input

×



Weight

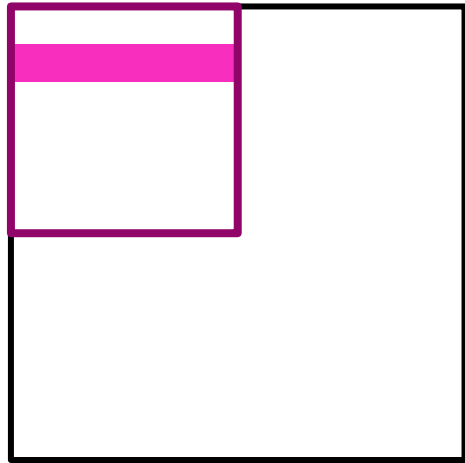
=



Output

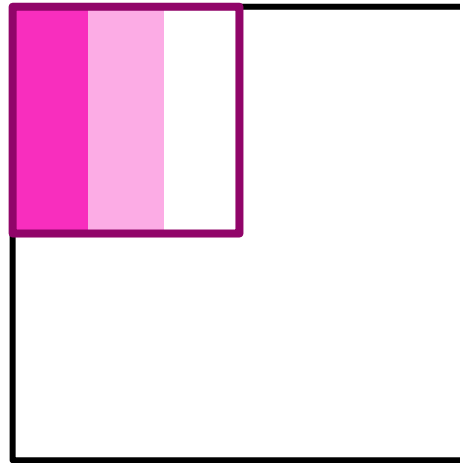


# Output stationary - Local weight stationary



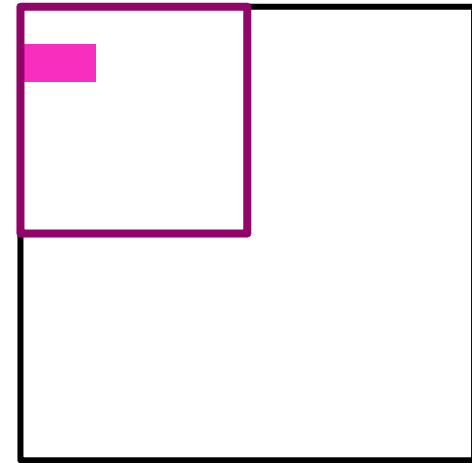
Input

×



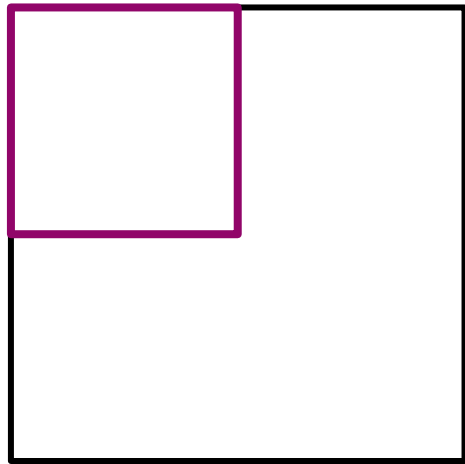
Weight

=



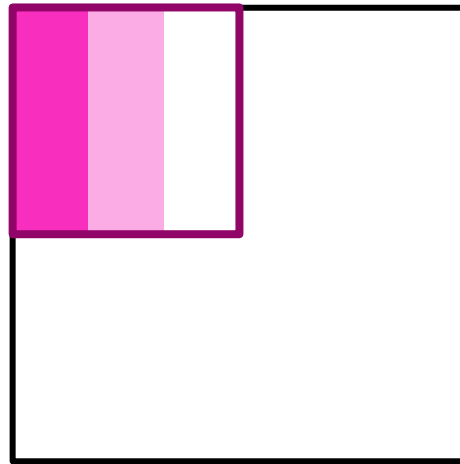
Output

# Output stationary - Local weight stationary



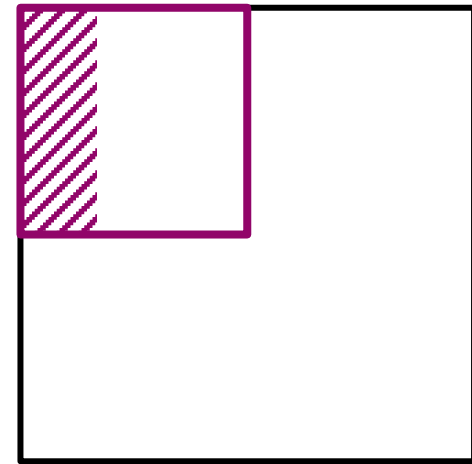
Input

×



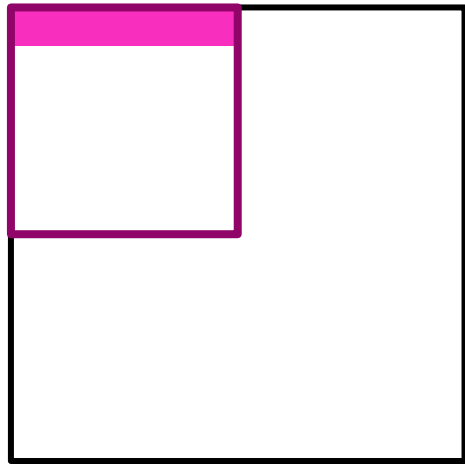
Weight

=



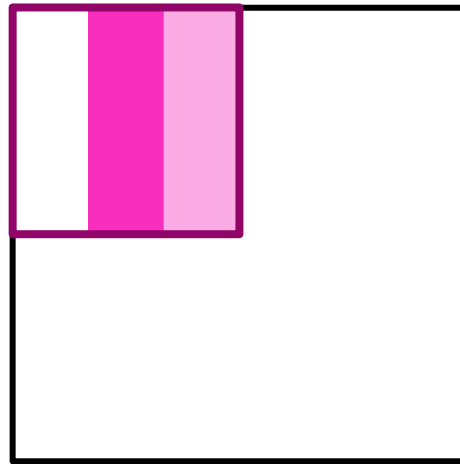
Output

# Output stationary - Local weight stationary



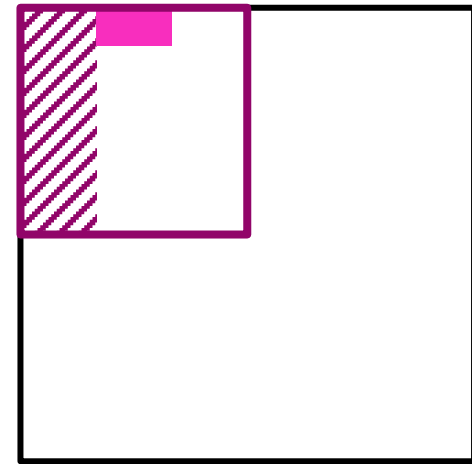
Input

×



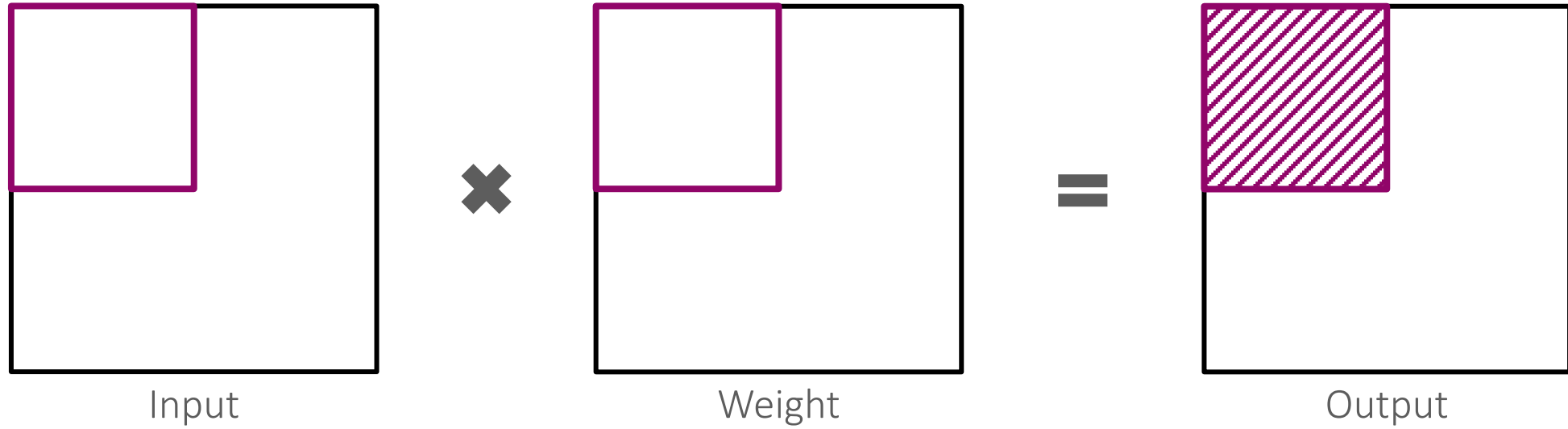
Weight

=

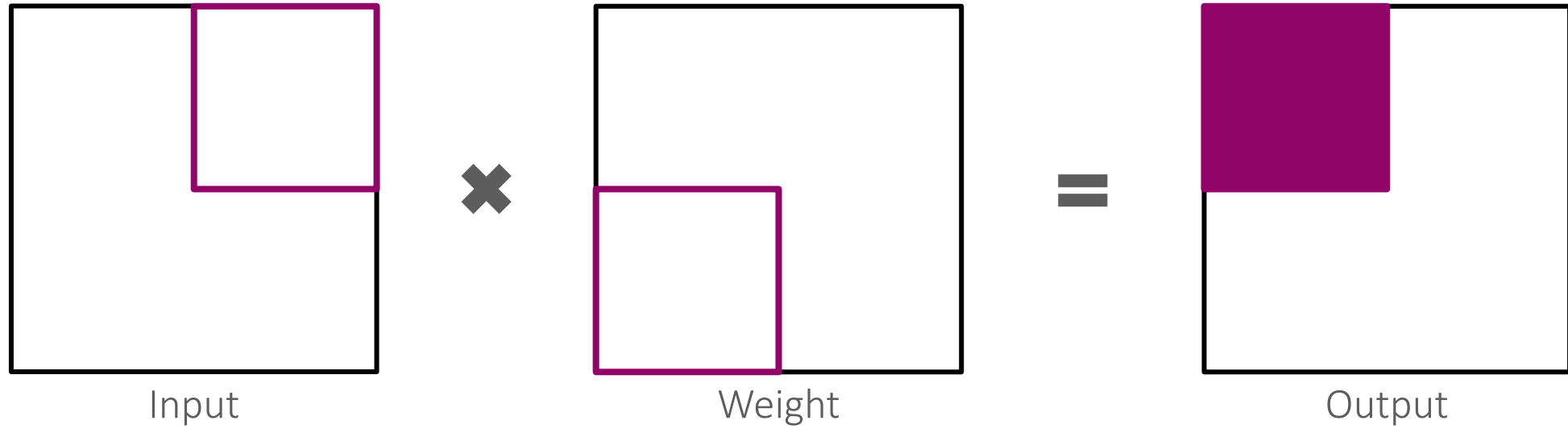


Output

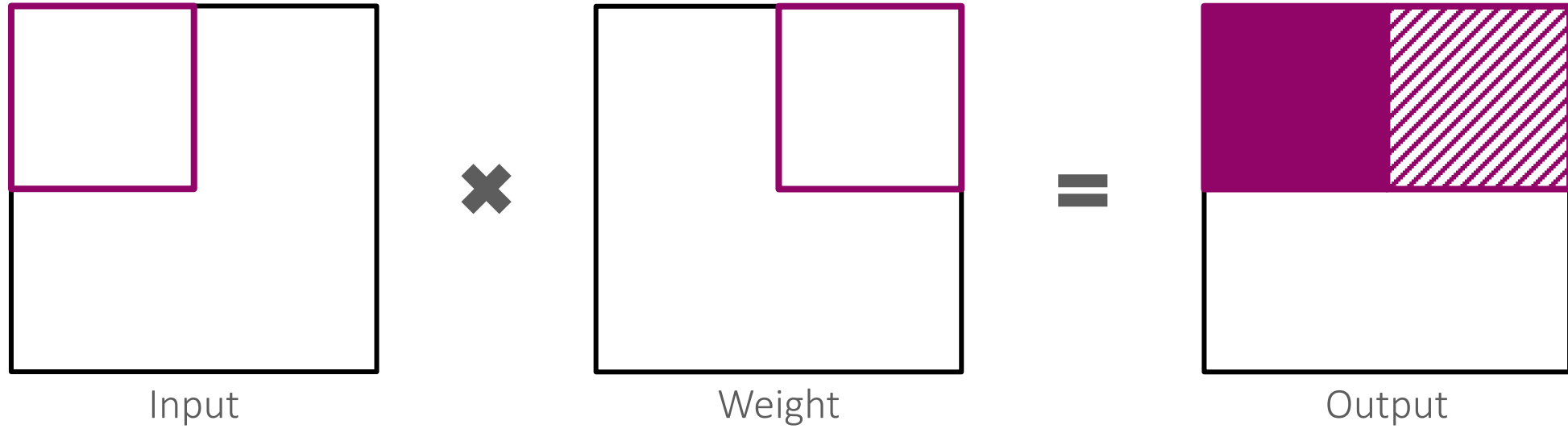
# Output stationary - Local weight stationary



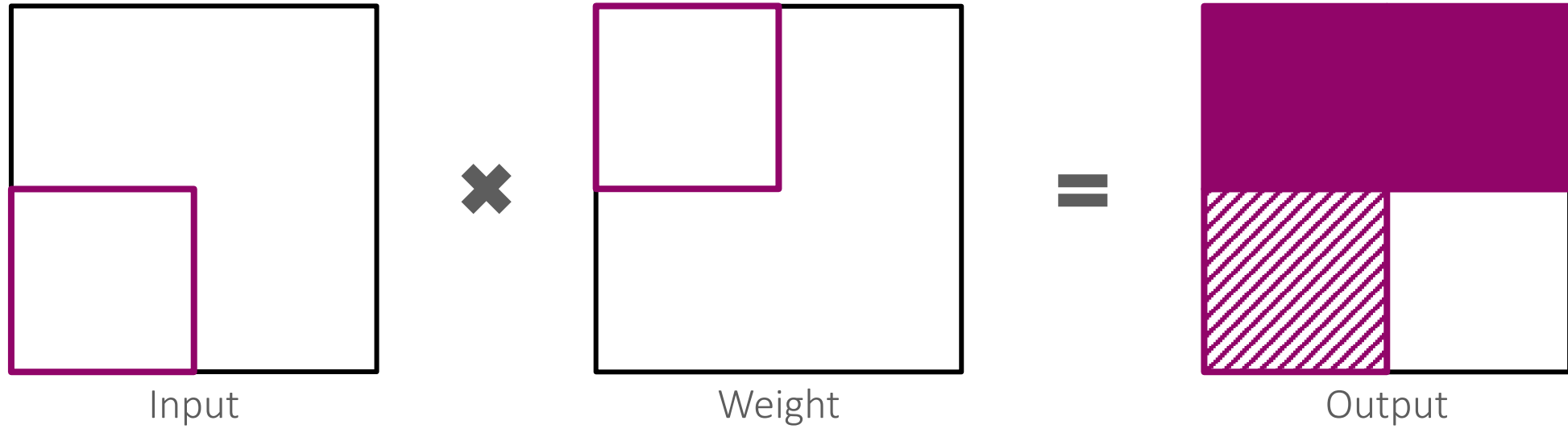
# Output stationary - Local weight stationary



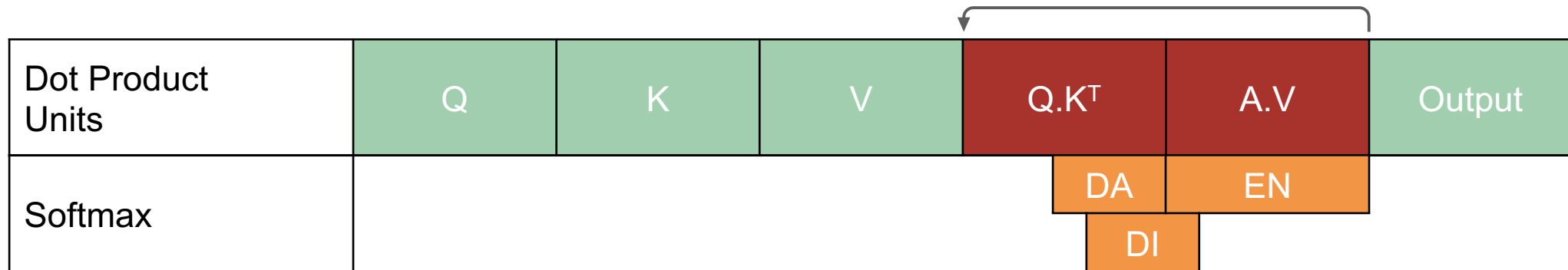
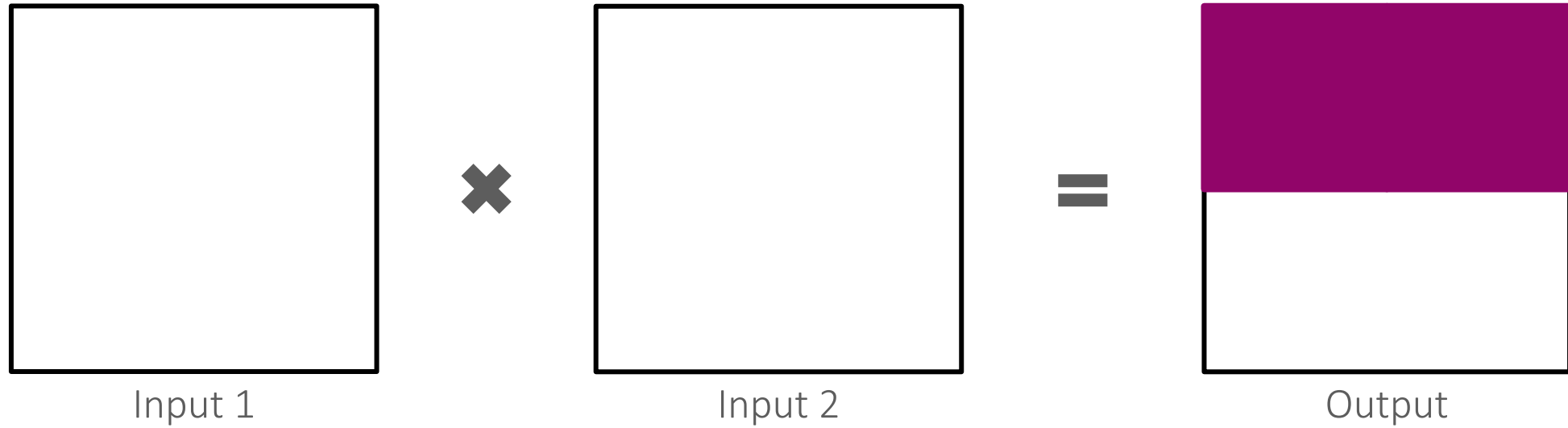
# Output stationary - Local weight stationary



# Output stationary - Local weight stationary

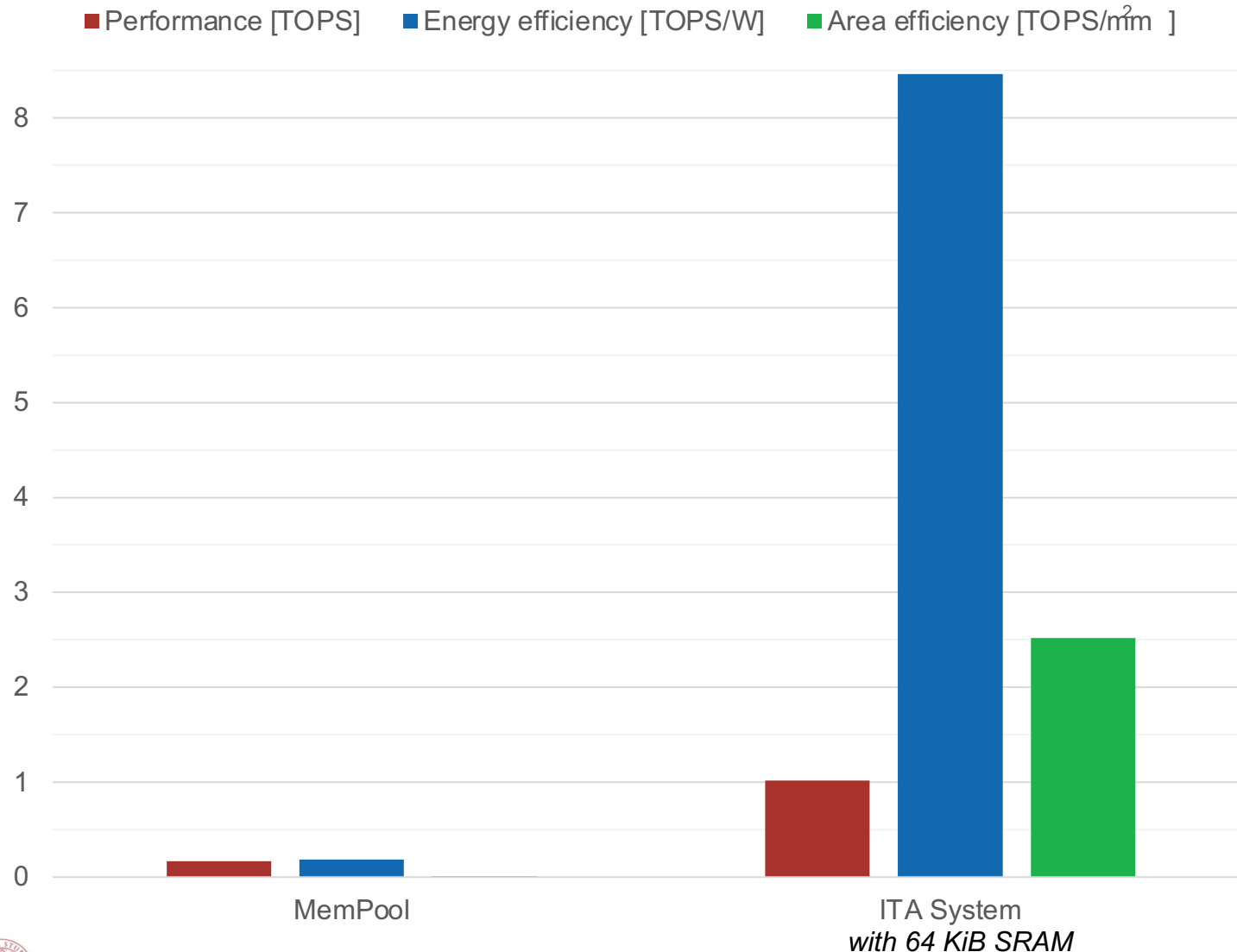


# Fused Q.K<sup>T</sup> and A.V computation





# Comparison to a software baseline on MemPool



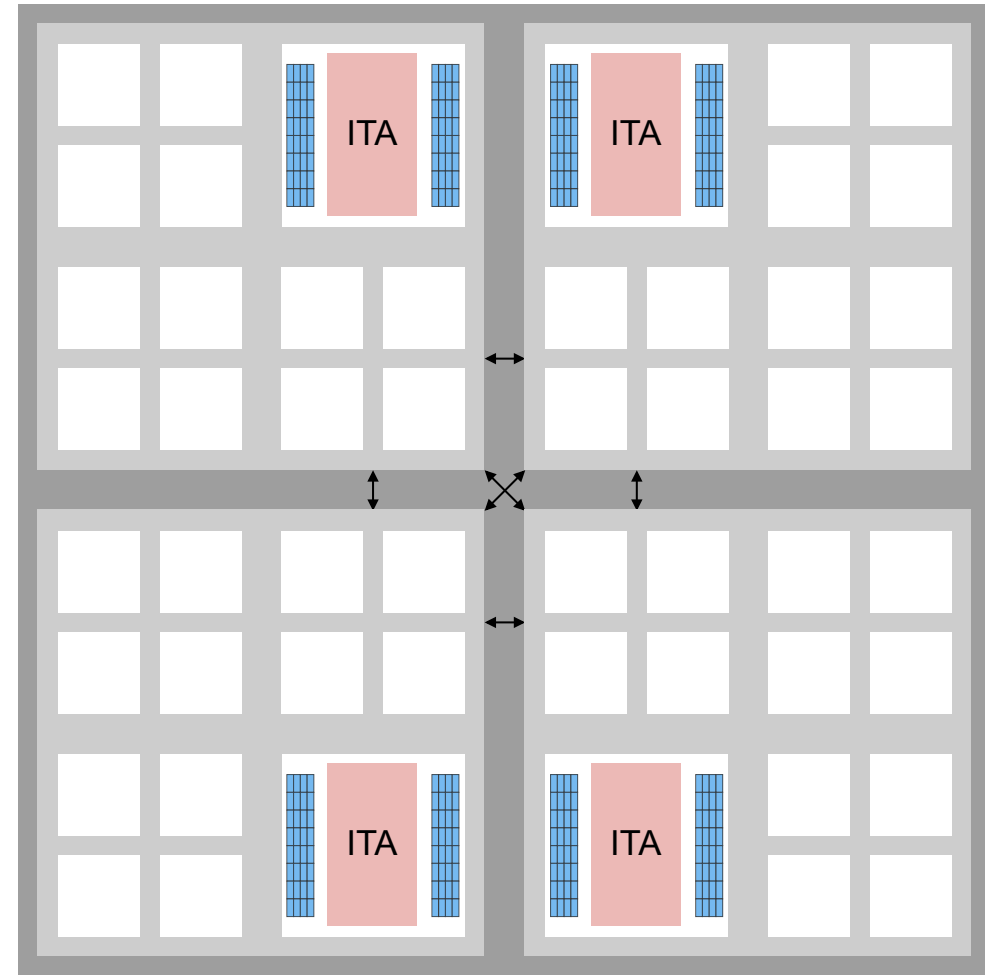
**Performance**  
increase of **6x**

**Energy Efficiency**  
increase of **45x**

**Area Efficiency**  
increase of **220x**

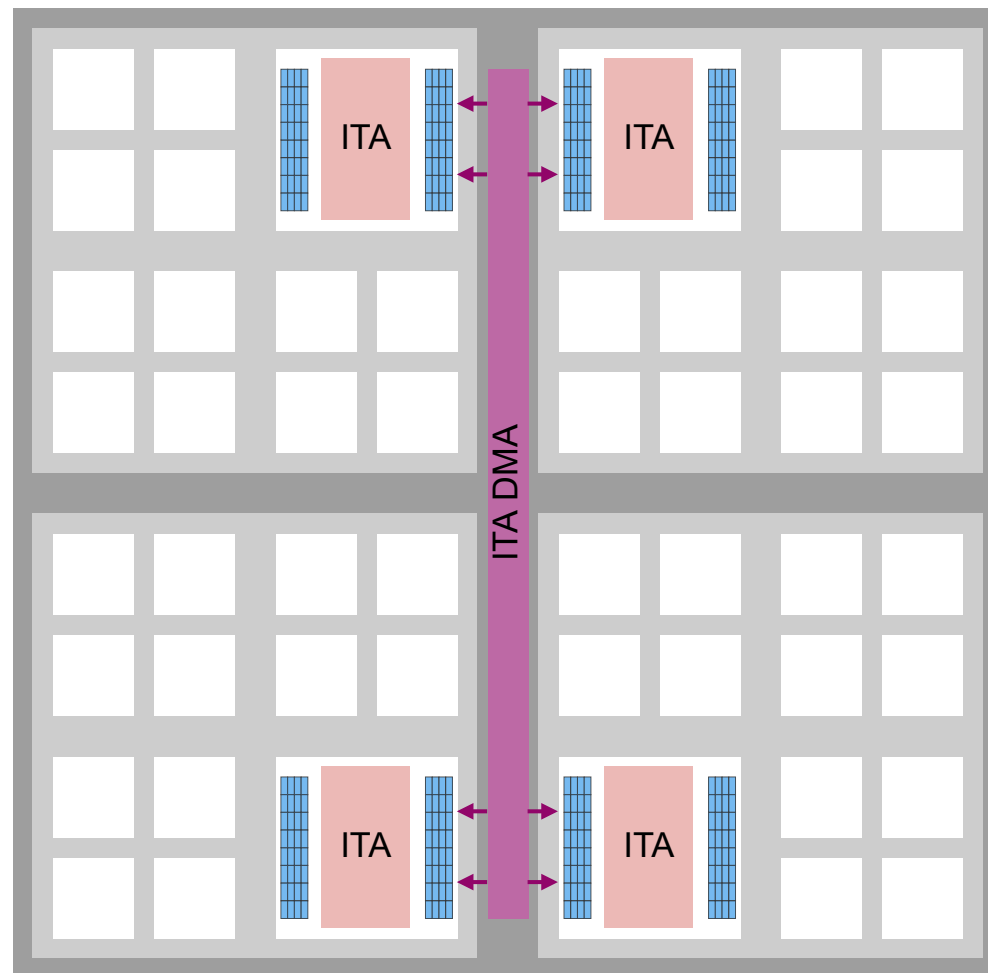
# Integrating ITA into MemPool

- ✓ Where to put ITA?
- ✓ How to connect ITA to L1 memory?
- How to refill L1 from L2 memory for ITA?

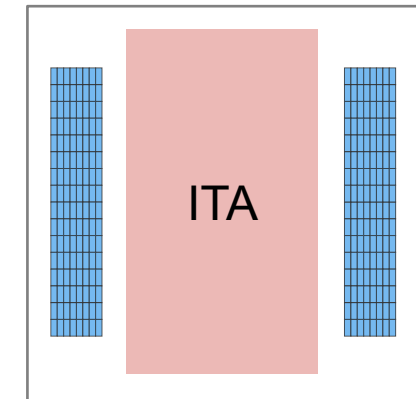
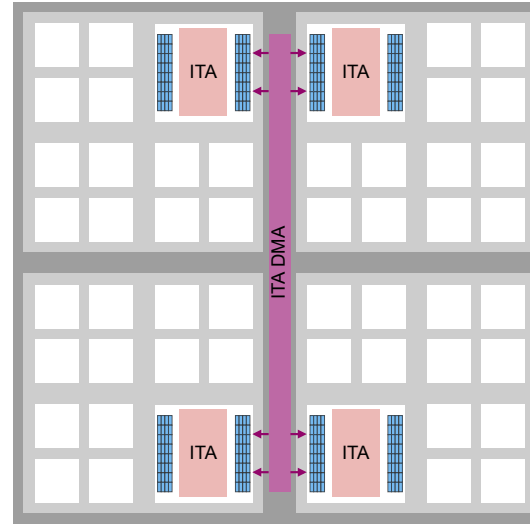
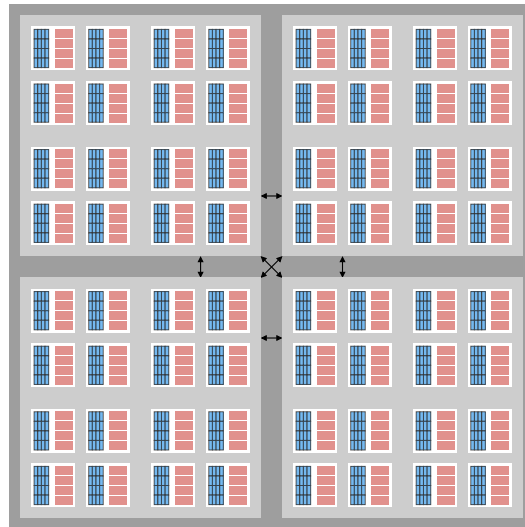


# Adding a special DMA for ITA

- Moves transformer data from L2 to L1 memory
- Inputs are broadcasted to all groups
- Two 16 bytes/cycle ports per group



# Comparison to MemPool and ITA System



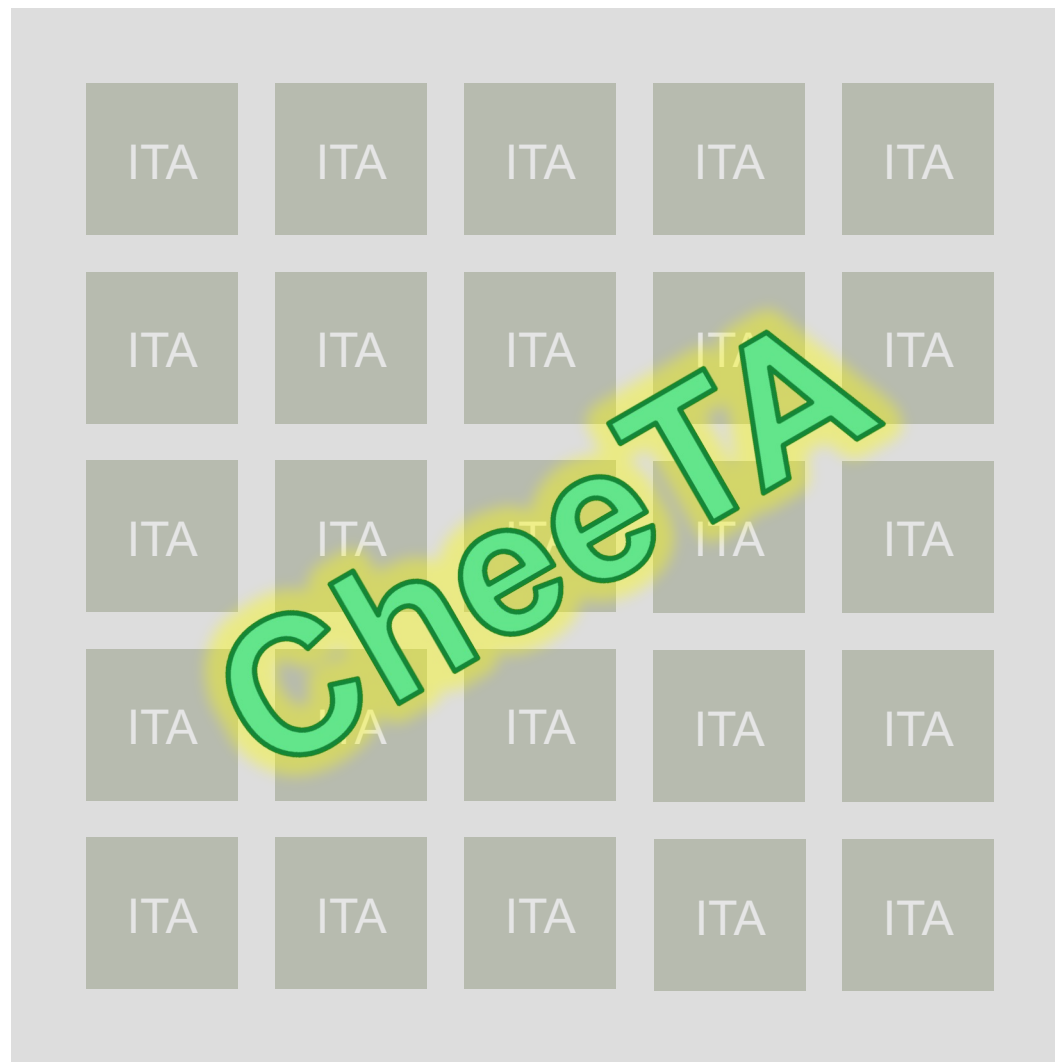
	MemPool	ITA & Banks	ITA only	ITA System
Throughput [TOPS]	0.135	3.43	3.43	1.02
Energy efficiency [TOPS/W]	0.159	7.09	12.3	8.46
Area efficiency [TOPS/mm <sup>2</sup> ]	0.0114	2.10	5.02	2.52

Red arrows and text indicate performance improvements: 25x increase in throughput, 45x increase in energy efficiency, and 2x increase in area efficiency for the ITA System compared to MemPool.

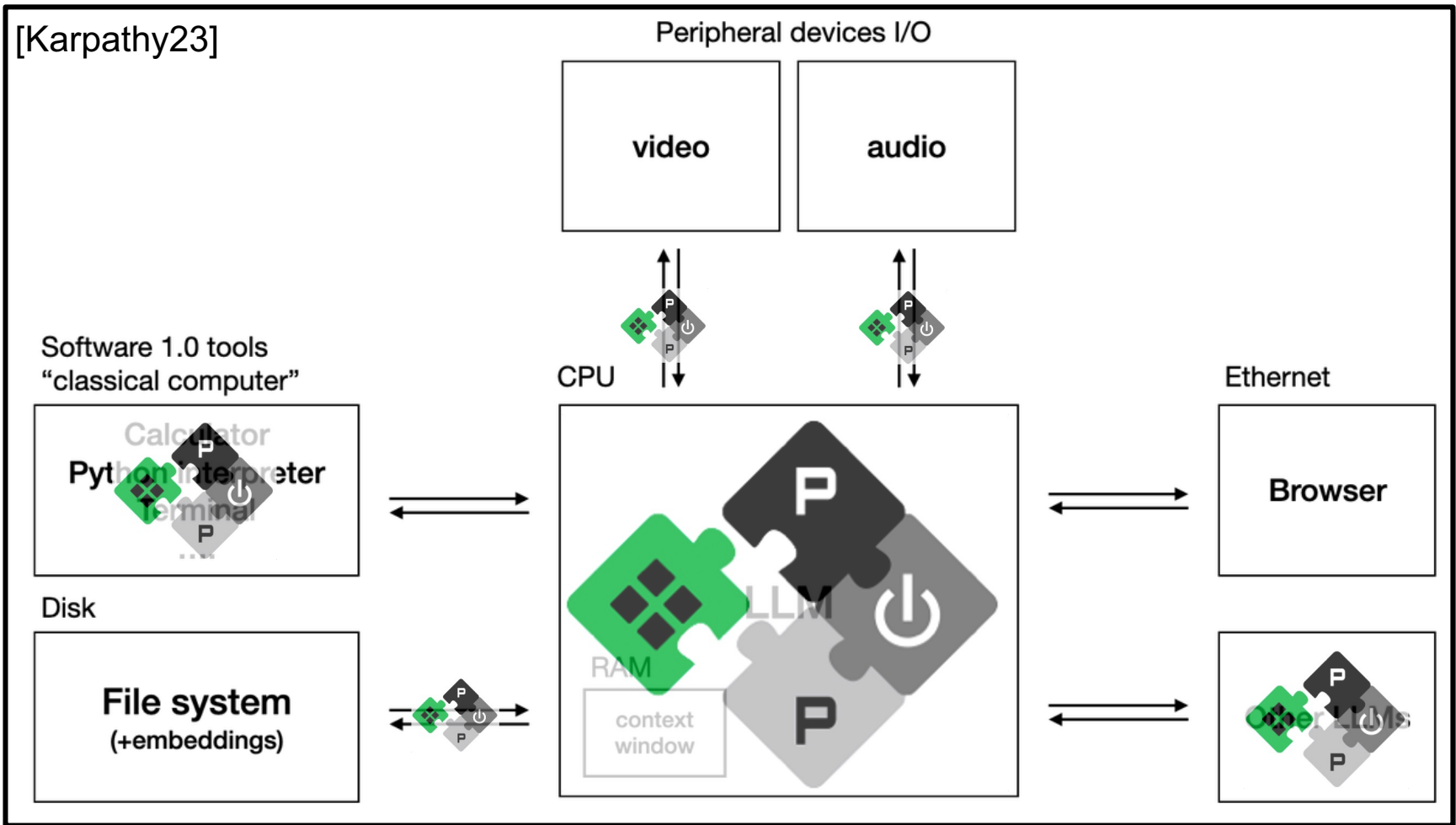
# Future of Mempool+ITA: Scaling up further

- 10B+ models (LLAMA2)
- Block-FP capability ( $<8b/w,act$ )
- Sparsity handling
- Multi-chiplet *terapool*
- 3D memory

**Accelerate LLMs  
and reach 100  
TFLOPS or higher  
in a few W**



# Embodied AI vision: LLM everywhere?



-  **Efficient**
-  **Safe**
-  **Real-time**
-  **Secure**



**Thank You!**