Designing "Artificial Brains" for Next-Generation Autonomous Systems

Luca Benini lbenini@iis.ee.ethz.ch, luca.benini@unibo.it

PULP Platform

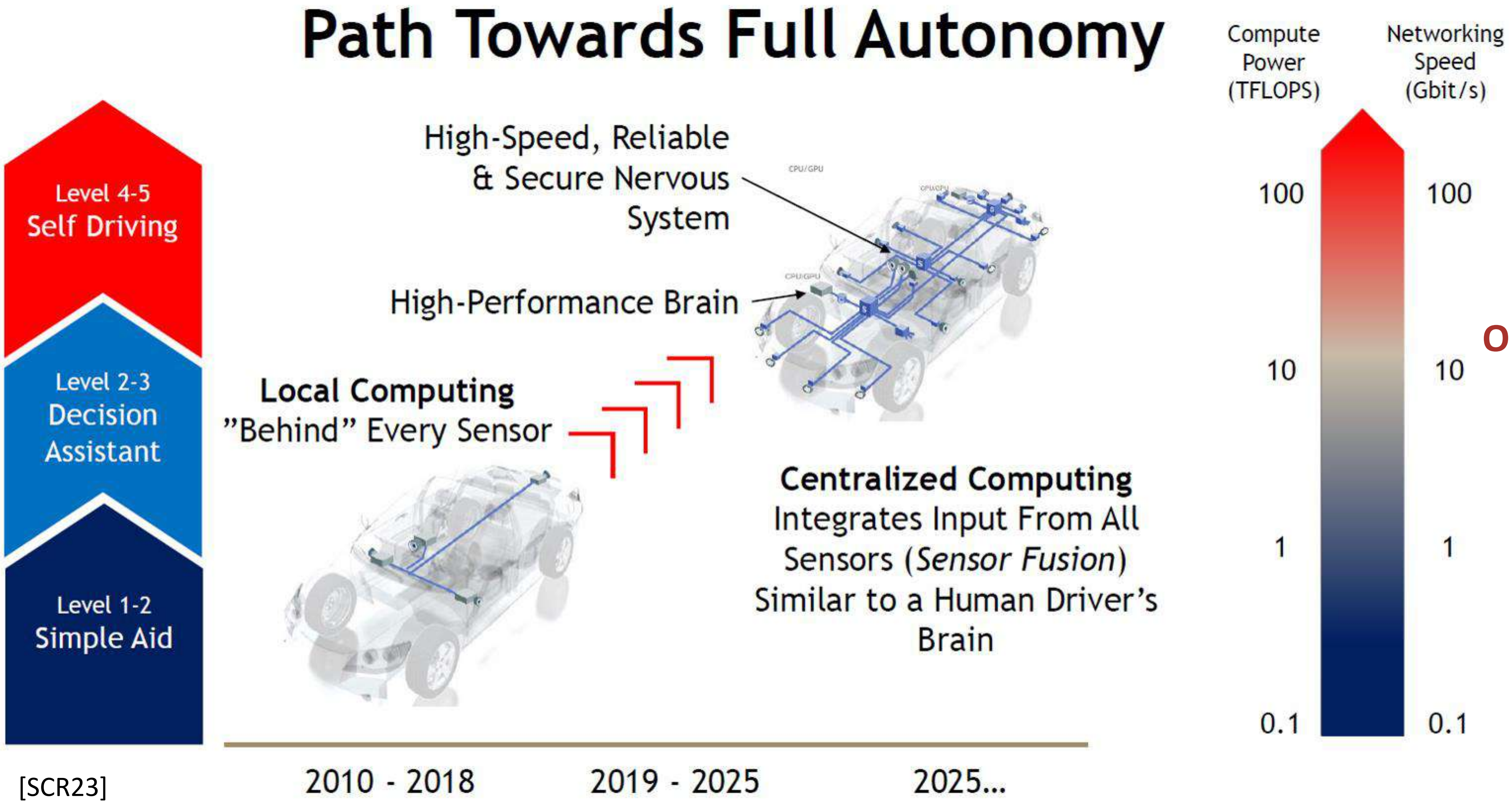
Open Source Hardware, the way it should be!

@pulp_platform

pulp-platform.org

youtube.com/pulp_platform

Autonomous Systems: Roadmap



Path Towards Full Autonomy

Level 4-5
Self Driving

Level 2-3
Decision Assistant

Level 1-2
Simple Aid

High-Speed, Reliable & Secure Nervous System

High-Performance Brain

Local Computing
"Behind" Every Sensor

Centralized Computing
Integrates Input From All Sensors (*Sensor Fusion*)
Similar to a Human Driver's Brain

2010 - 2018 | 2019 - 2025 | 2025...

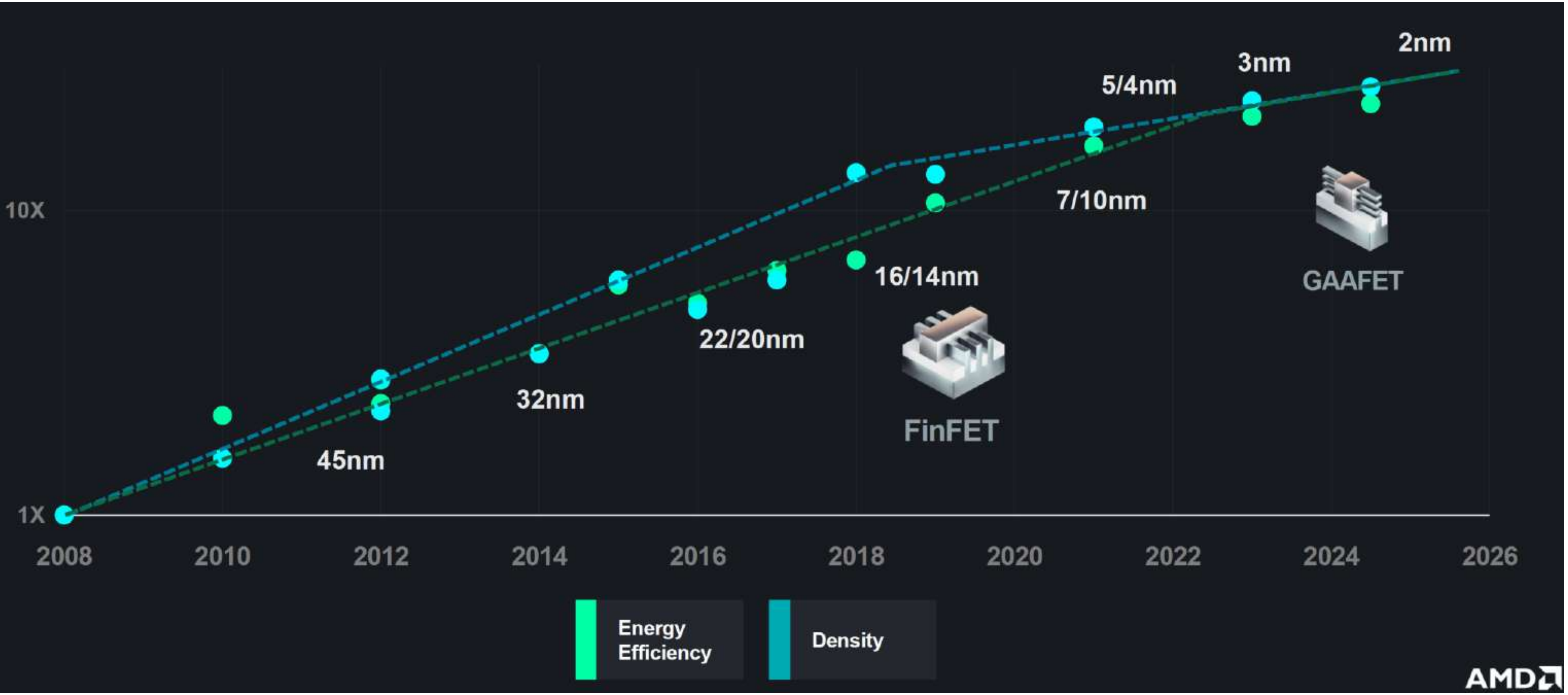
Compute Power (TFLOPS)	Networking Speed (Gbit/s)
100	100
10	10
1	1
0.1	0.1

Efficient

On-car Computing
$P_{MAX} < 1.5$ kW

[SCR23]

Embodied AI



[AMD HotChips24]

Efficient

On-car Computing

$P_{MAX} < 1.5$ kW

Model complexity

10× every ~2.5 years

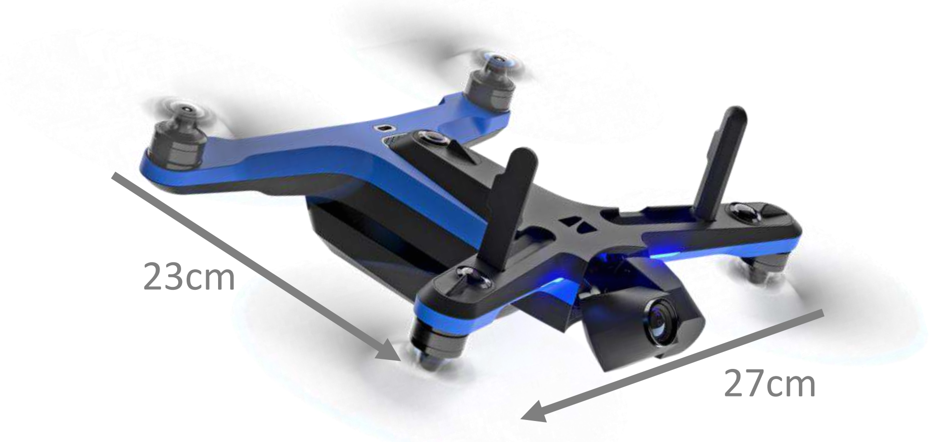
Moore's Law

10x every 12 years!

Autonomous Nano-Drones

Advanced autonomous drone

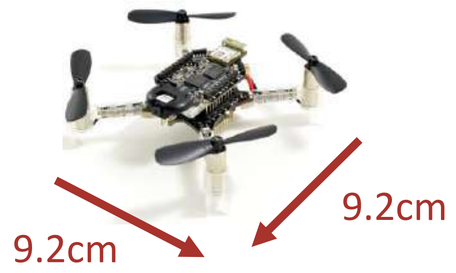
A. Bachrach, “Skydio autonomy engine: Enabling the next generation of autonomous flight,” IEEE Hot Chips 33 Symposium (HCS), 2021



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

- 3D Mapping & Motion Planning
- Object recognition & Avoidance
- 0.06m^2 & **800g of weight**
- Battery Capacity **5410 mAh**

Nano-drone



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

Smaller form factor of 0.008m^2	
Weight:	**27 g (30× lighter)**
Battery capacity:	**250 mAh (20× smaller)**

Intelligence in a 30× smaller payload, 20× lower energy budget?

Achieving True Autonomy on Nano-UAVs

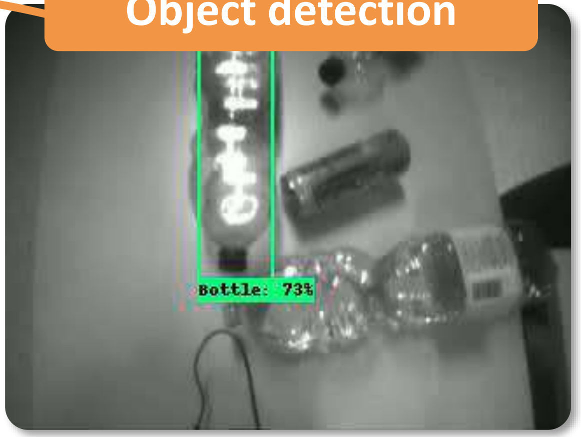
Multiple,

complex,

heterogeneous

tasks at high speed and robustness

fully on board



Object detection

Obstacle avoidance & Navigation

Environment exploration

Multi-GOPS workload at extreme efficiency → P_{max} 100mW

Efficiency through Heterogeneity: Multi-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

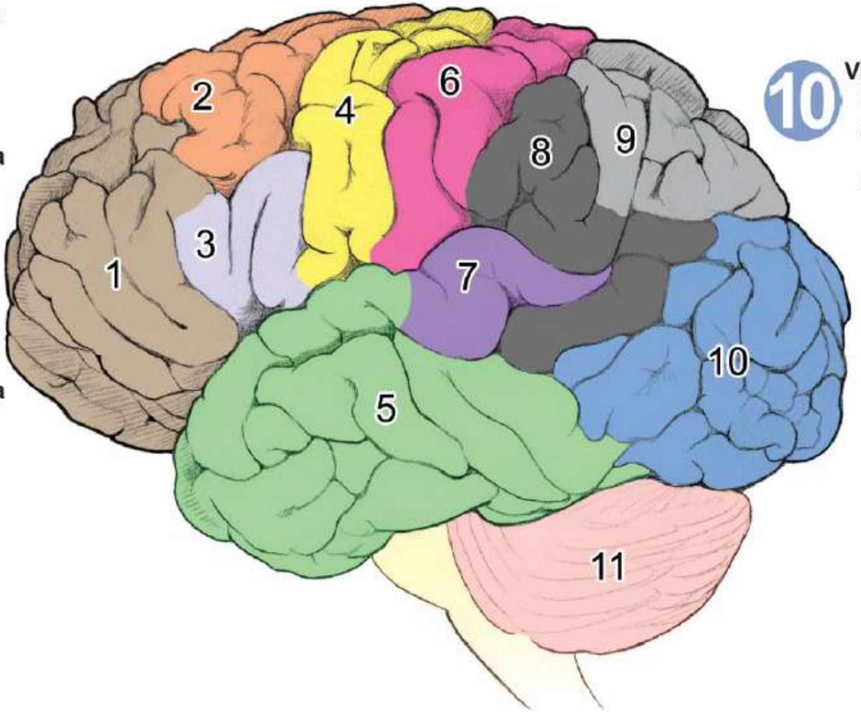
4 Motor Function Area
Ability to move muscles

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

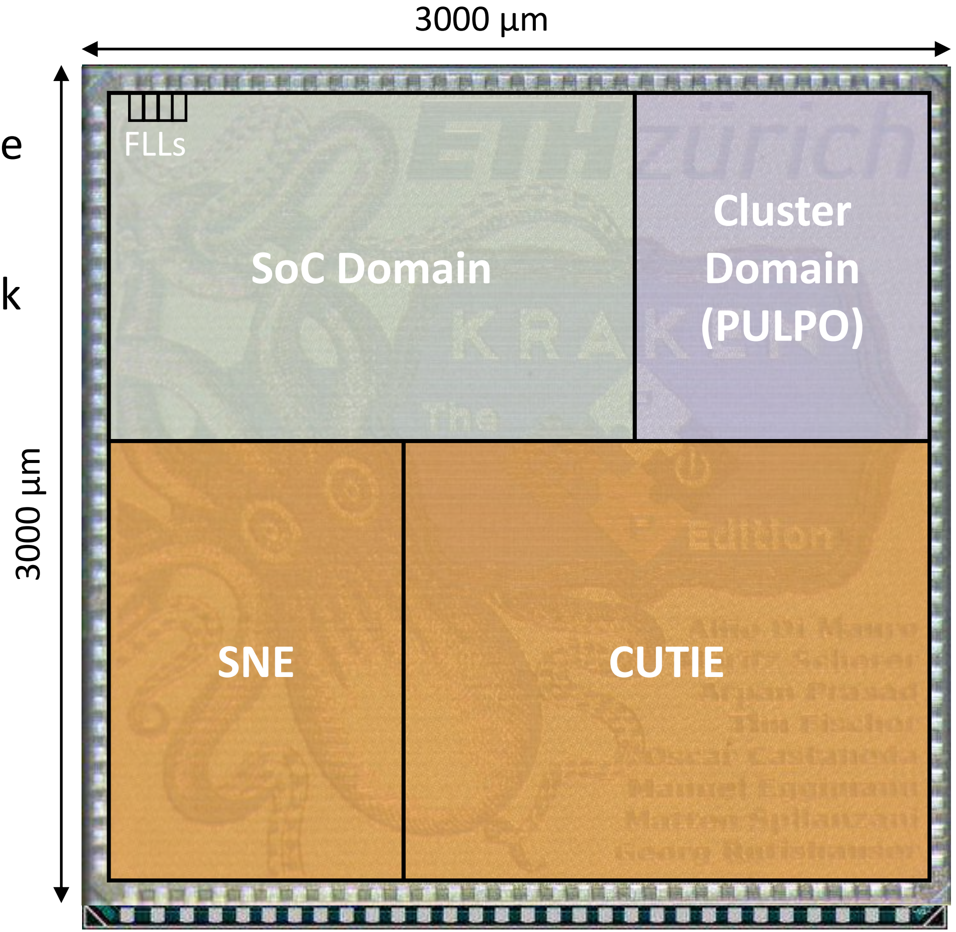
11 Motor Functions
Coordination of movement
Balance
Posture

→ **Multi-sensor frame-based event-based** → **Perception** → **Fusion Reasoning**

Kraken: 22nm SoC, Multiple Heterogeneous Accelerators

The ***Kraken***: an "Extreme Edge" Brain

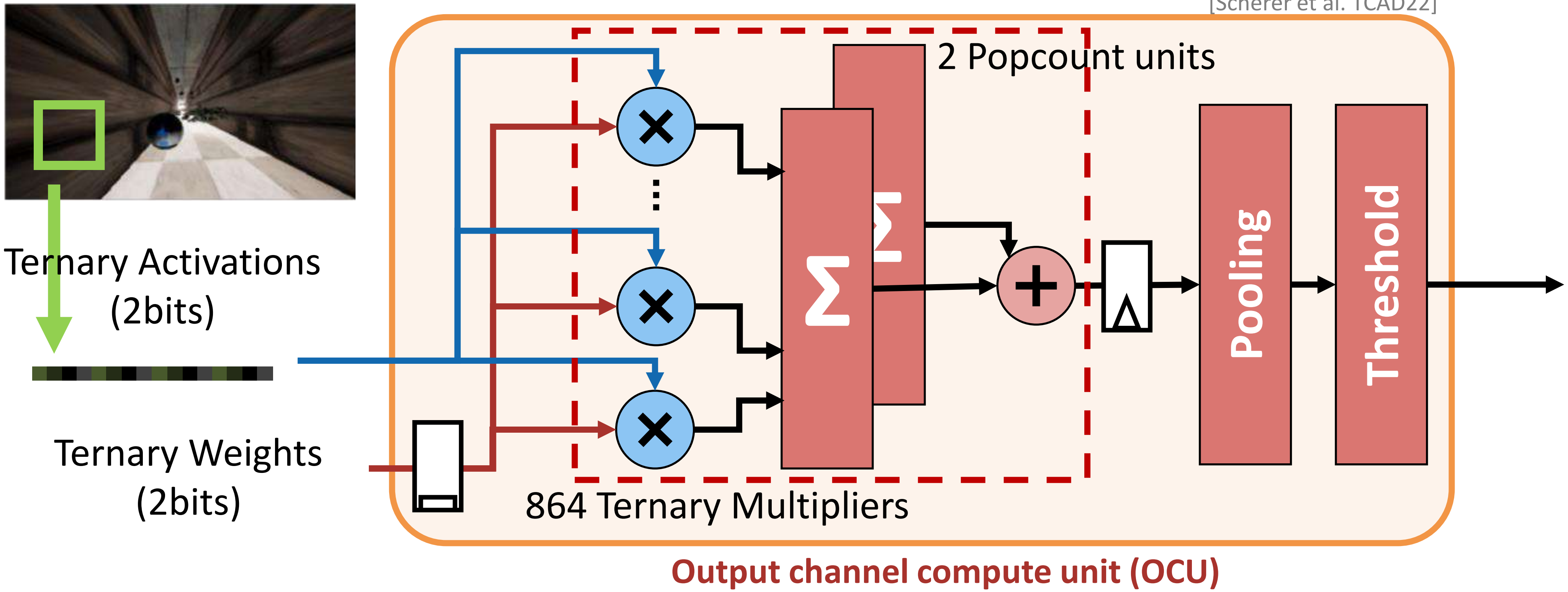
- RISC-V Cluster
  8 Compute cores +1 DMA core
- CUTIE
  Dense ternary-neural-network accelerator
- SNE
  Energy-proportional spiking-neural-network accelerator



Technology	22 nm FDSOI
Chip Area	9 mm^2
SRAM SoC	1 MiB
SRAM Cluster	128 KiB
VDD range	0.55 V - **0.8 V**
Cluster Freq	**~370 MHz**
SNE Freq	**~250 MHz**
CUTIE Freq	**~140 MHz**

CUTIE: Perception from Frame Sensors

[Scherer et al. TCAD22]

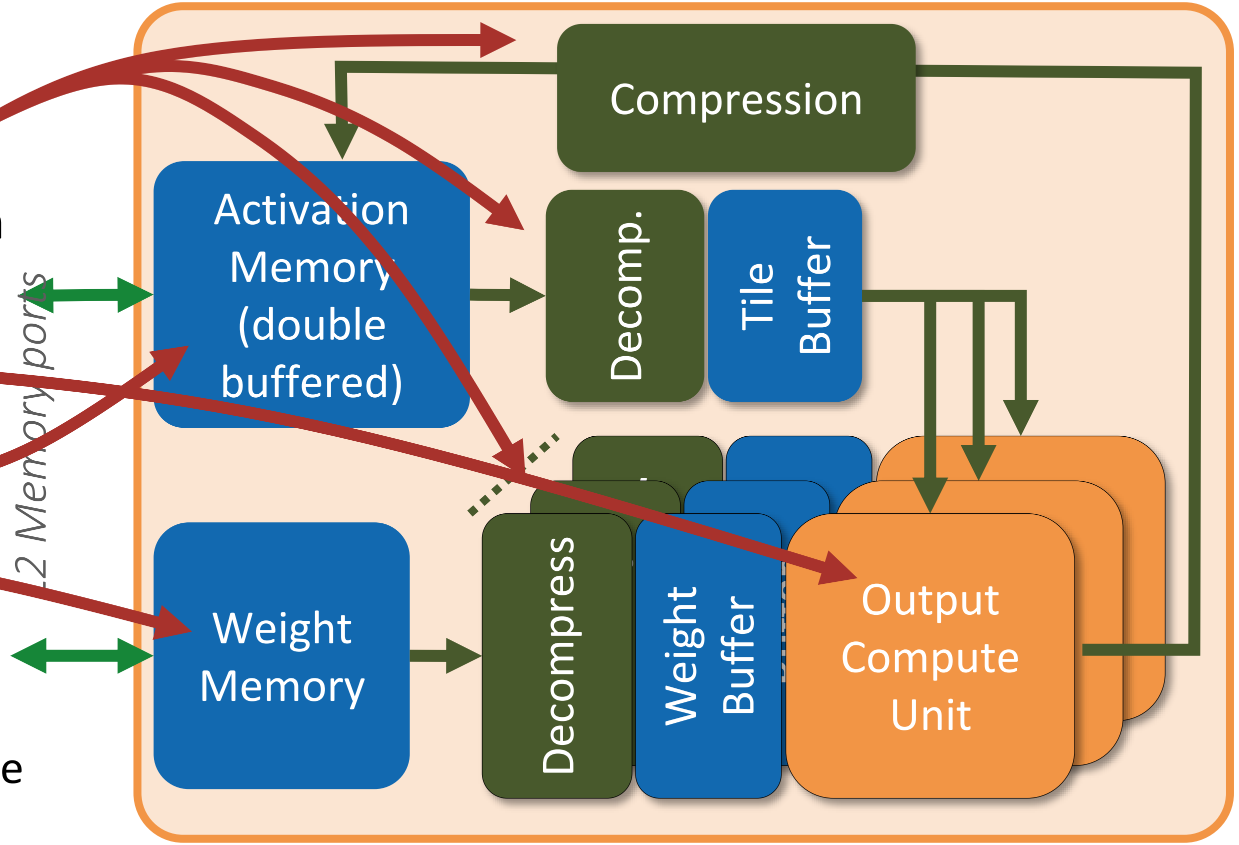


- **Completely Unrolled Ternary Neural Inference Engine**: K × K window, all input channels, cycle-by-cycle sliding
- One *Output Compute Unit* (OCU) computes one output activation per cycle!
- Zeros in weights and activations, spatial smoothness of activations reduce switching activity

Aggressive quantization and full specialization

Kraken`s CUTIE Implementation

- Data in 1.6 bits (Ternary value) with On-the-fly Compression/Decompression
- Configuration in Kraken
  - 96 channels (Output compute units)
  - 3 × 3 kernels
  - 64 × 64 pixels feature maps (158 KiB)
  - 9 layers of weights (117 KiB)
- Lots of TMAC/cycle
  - 96 OCUs, 96 Input channels, 3 × 3 kernels:
  - 96 × 96 × 3 × 3 = 82'944 Ternary-MAC/cycle



1fJ/MAC (1POPS/W) – Ternary OPS

General Purpose: Domain-Specialized RV32 Core (PE)

RISC-V 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
```

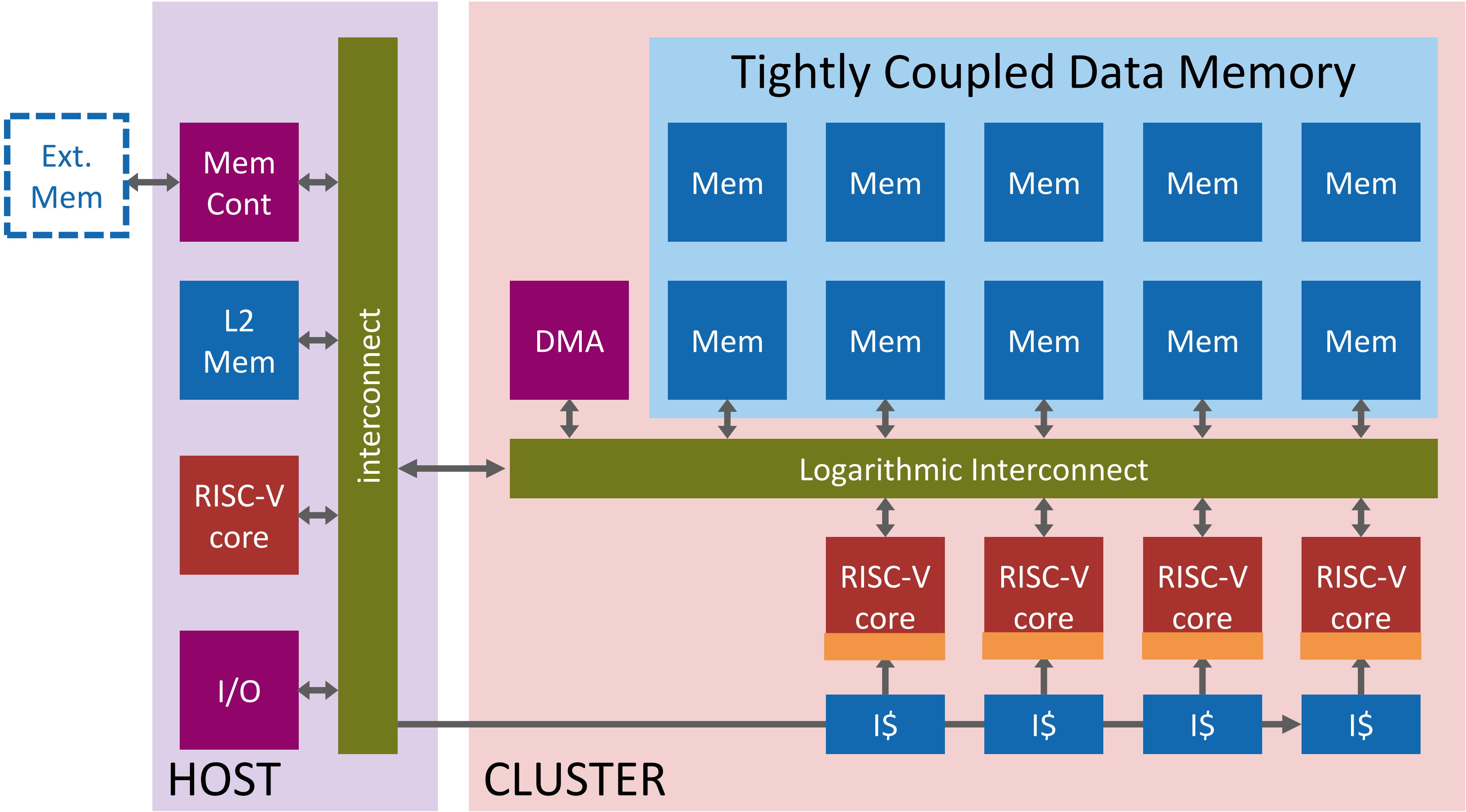
RISC-V core

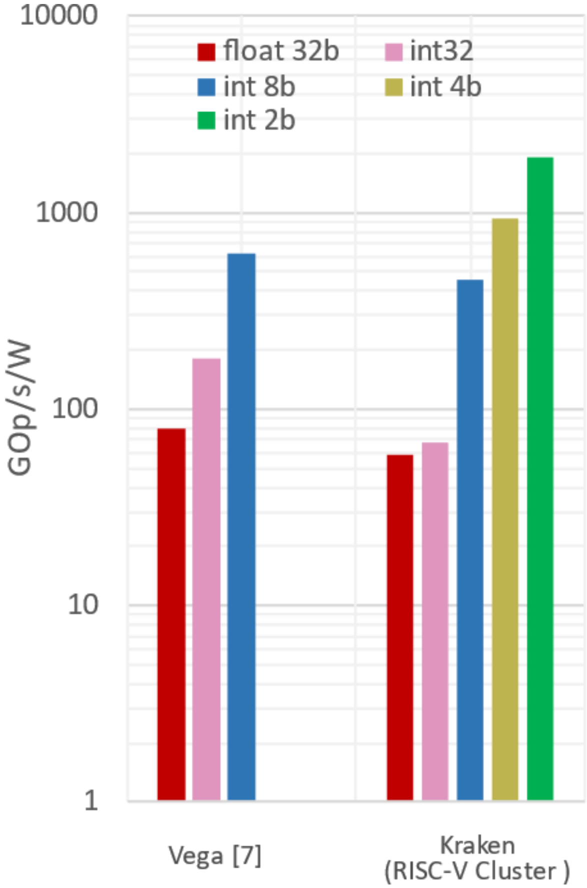
15x less instructions than Vanilla

90%+ ALU Utilization

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

PULP Paradigm: A PE **cluster** accelerates a host system





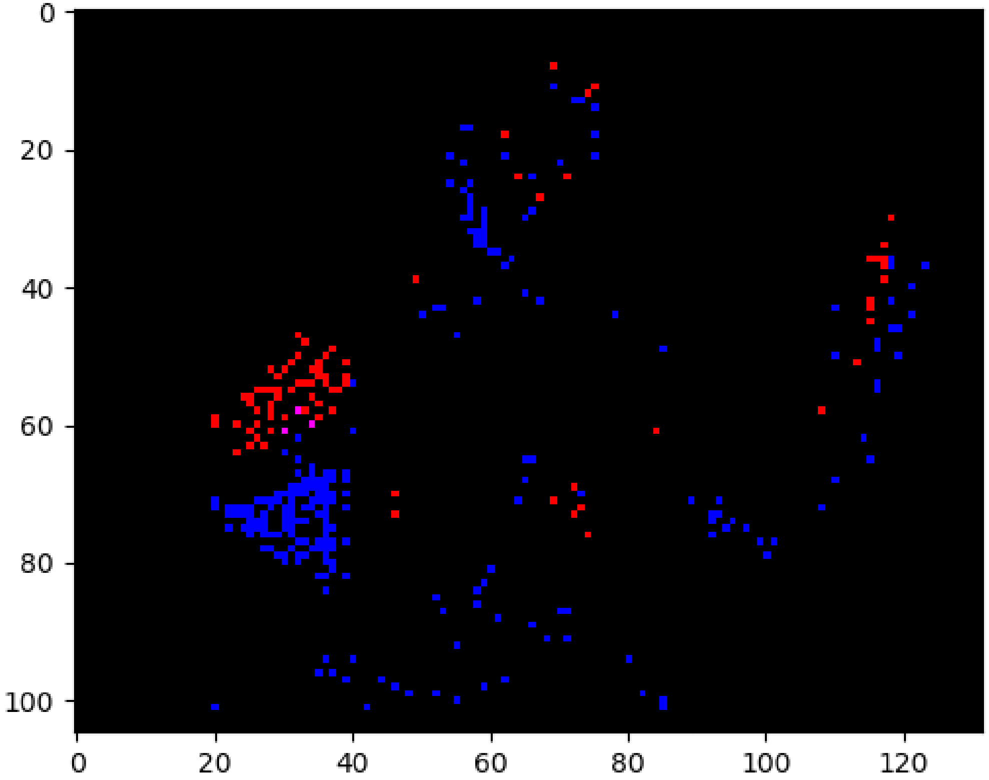
1TOPs/W - 2b/4b Operands

SNE: Perception on Event Sensors

Event Sensors – DVS camera

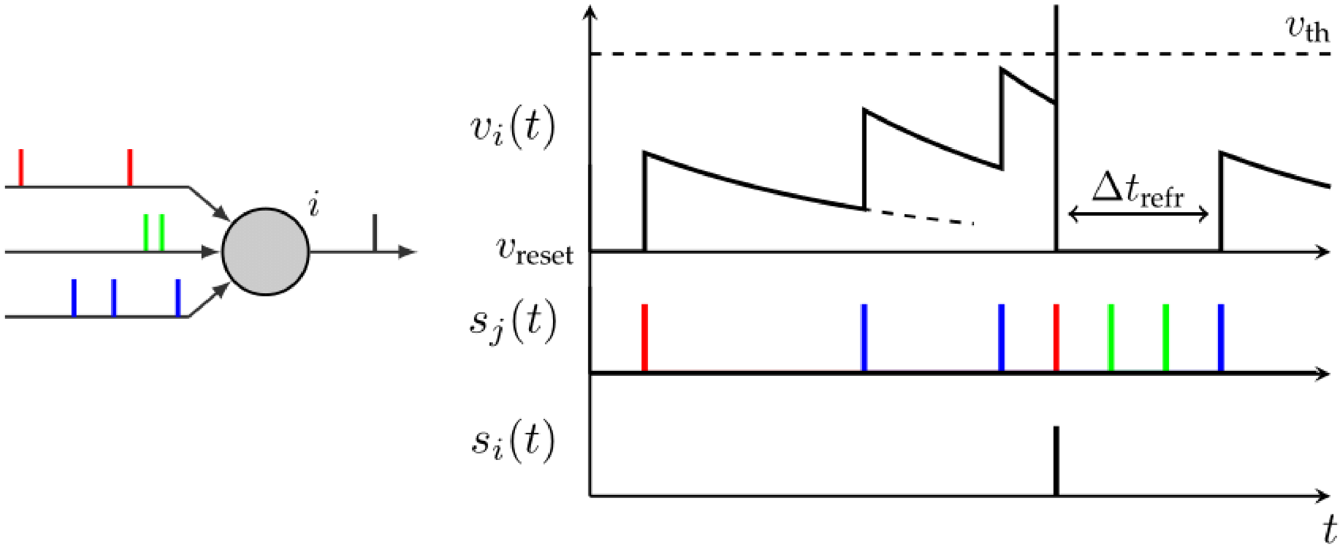
Ultra-low latency

Energy- proportional interface



Spiking Neural Engine (SNE)

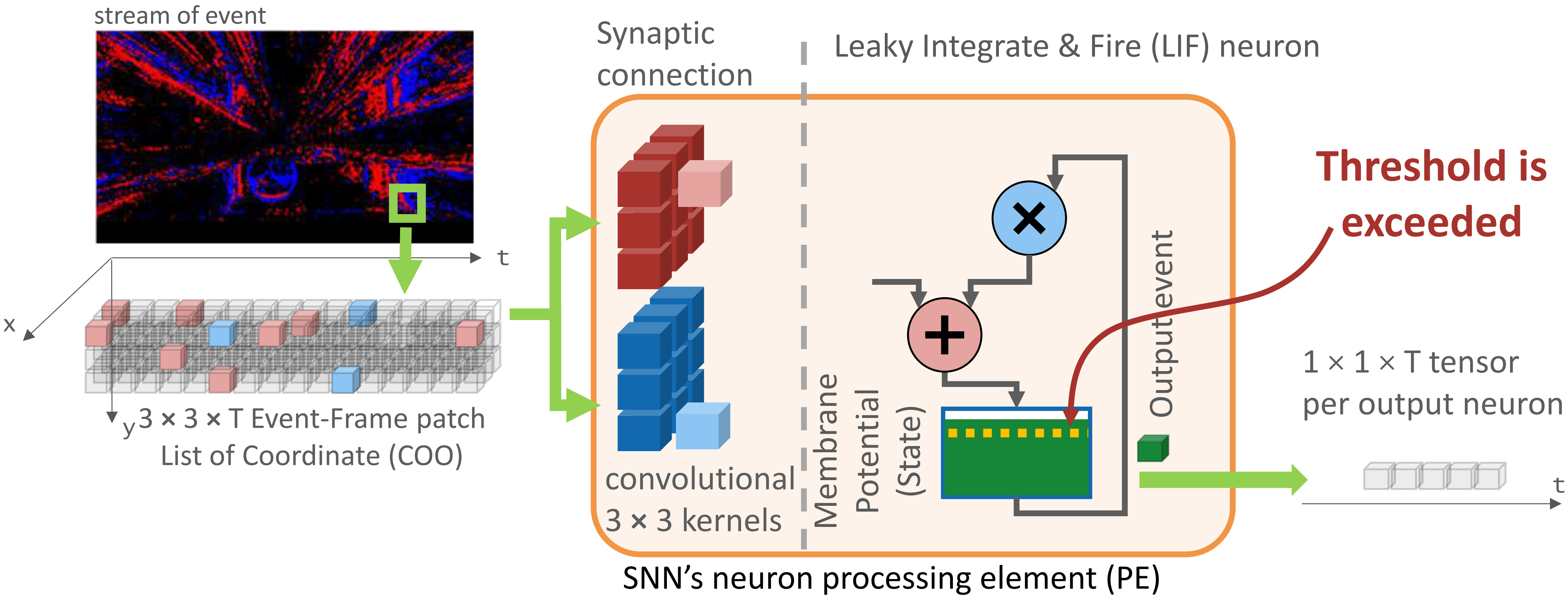
Leaky Integrate & Fire (LIF) neurons



[Di Mauro et al. DATE22]

SNE works seamlessly with DVS (event-based) sensors

Event consumption, and output spikes generation



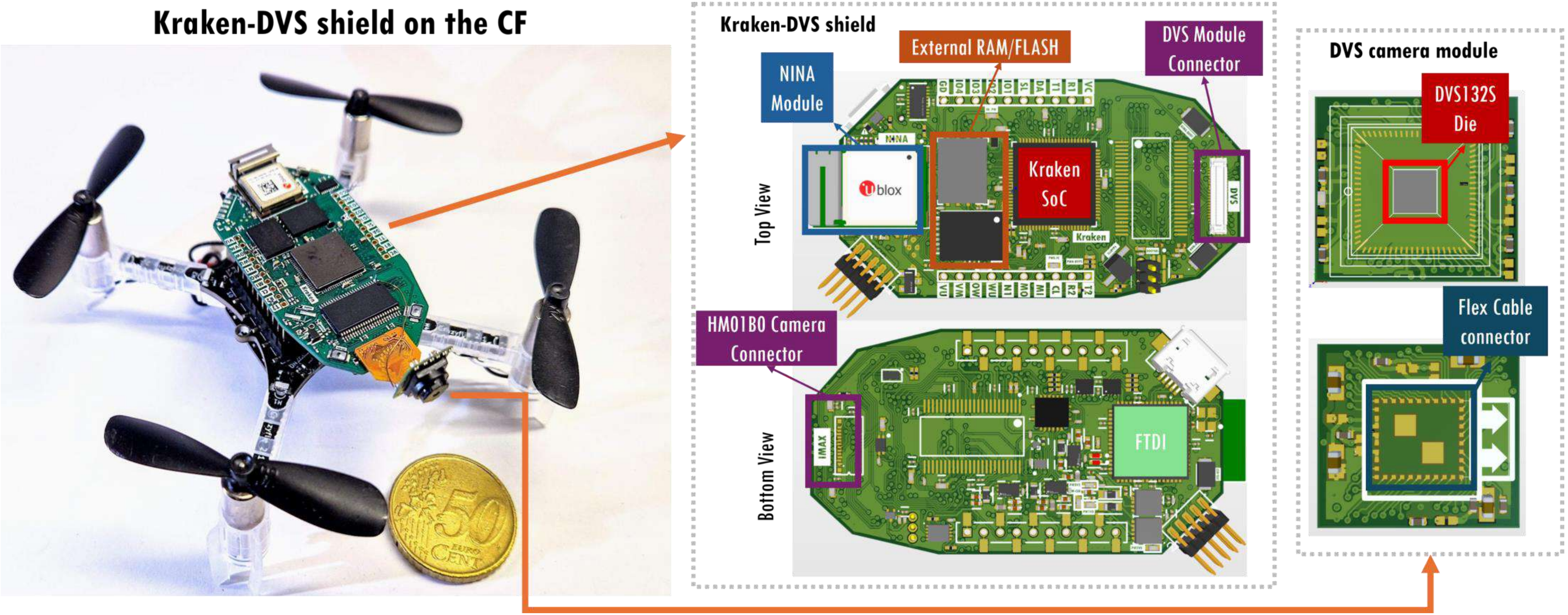
stream of event
t
x
y
3 × 3 × T Event-Frame patch
List of Coordinate (COO)
Synaptic connection
Leaky Integrate & Fire (LIF) neuron
Threshold is exceeded
convolutional 3 × 3 kernels
Membrane Potential (State)
Output event
1 × 1 × T tensor per output neuron
t
SNN's neuron processing element (PE)

A more complex dynamic than conventional DNNs neurons:

- Membrane Potential Accumulation/Activation 1× SynAcc = 1× 4b-ADD + 1× 8b-COMPARE
- Membrane Potential decay 1× SynDec = (1× 8b-MUL) + (1× 8b-MUL + 1× 8b-ADD)

Kraken Shield and System Architecture

- 7g payload
- DVS and frame-based cameras → real-time multi-modal perception.
- Designed for integration into nano-UAV platforms



Kraken Power Consumption (all Included)

Combined power consumption of SNE, CUTIE, PULP cluster

Model	Inference/s	µJ/inf	Power (mW)
SNE	1.02k	18	98
CUTIE	10k	6	110
PULP	221	750	165

P=373mW, representing just 5% of the UAV's power budget

Heterogeneous, Multiscale Accelerated Computing

Multiple Scales of acceleration

Extensions to processor cores

- Explore new extensions
- Efficient implementations

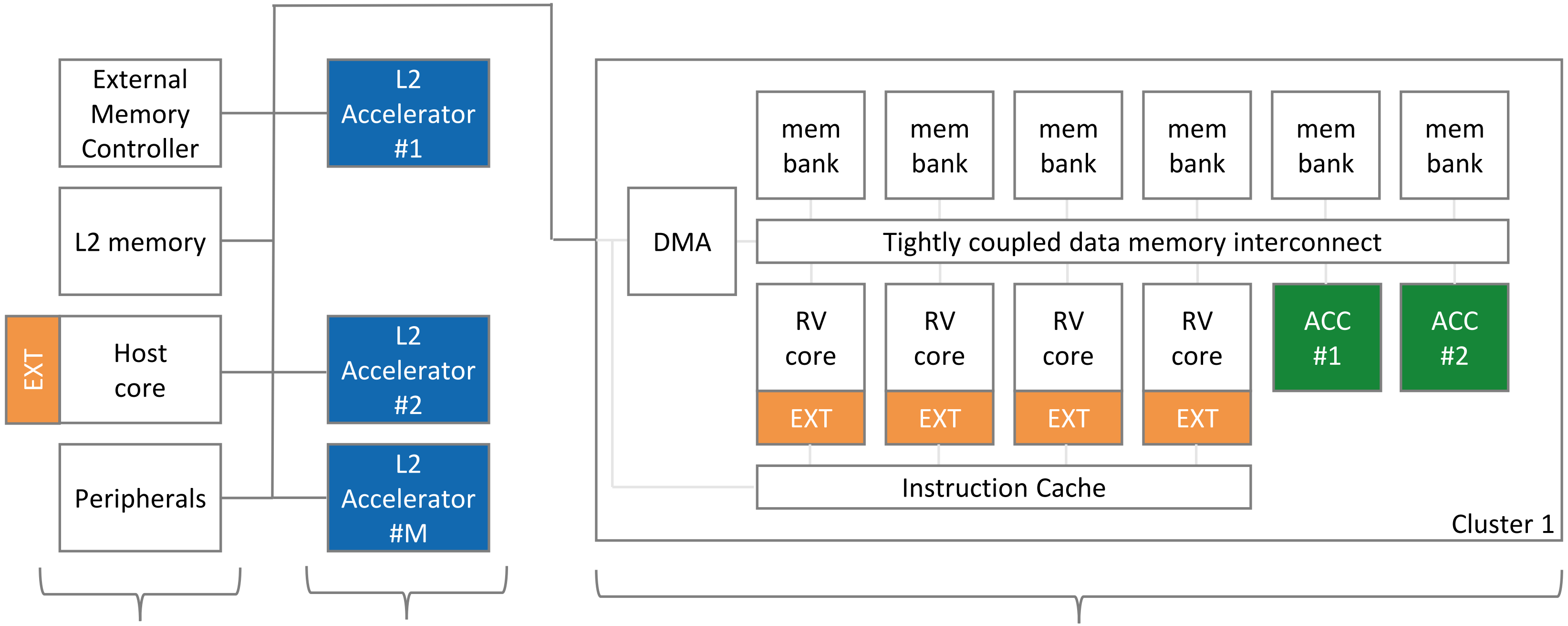
Shared-memory Accelerators

- Domain specific
- Local memory

Multiple Decoupled Accelerators

- Communication
- Synchronization

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



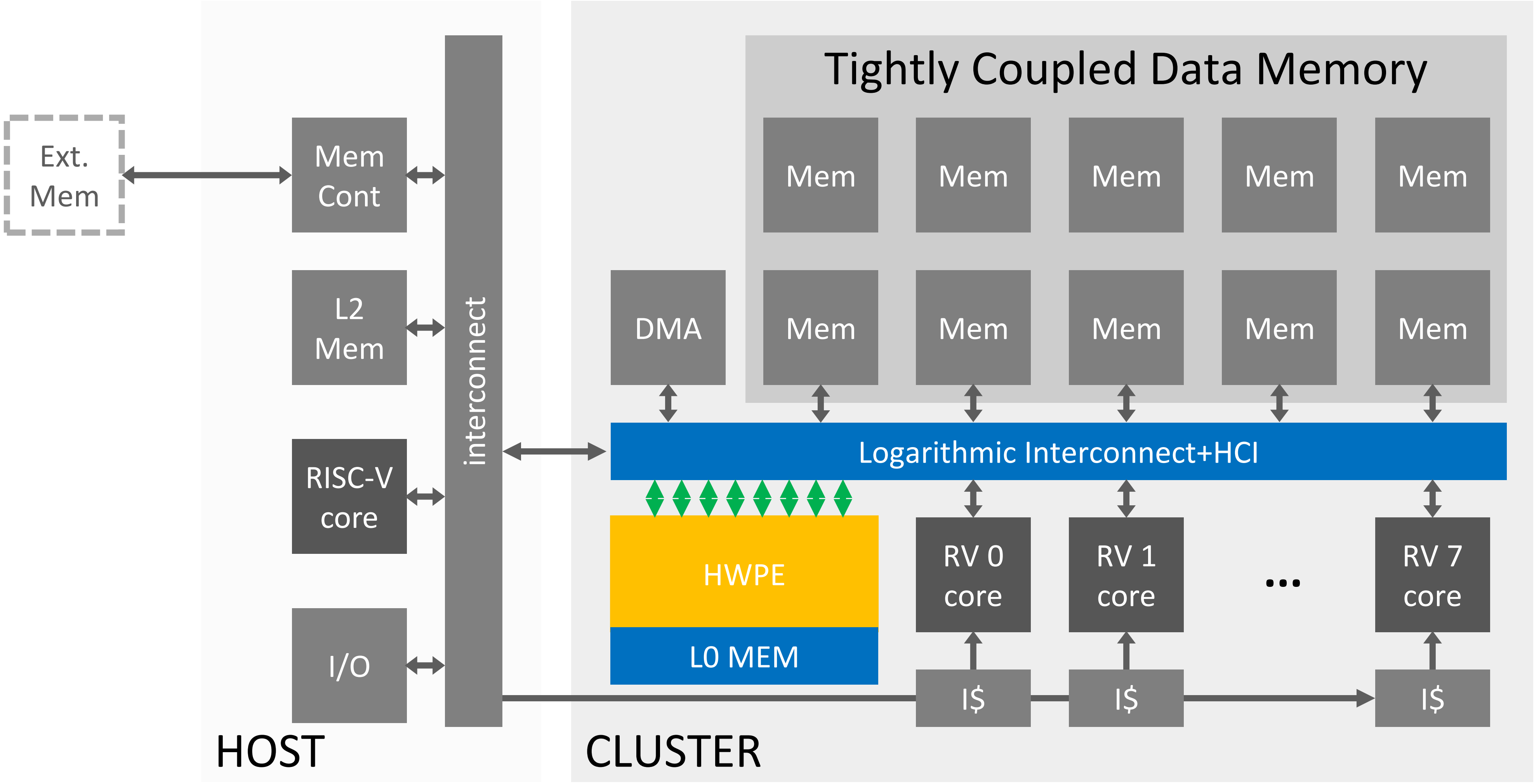
Host, L2, L3 IOs

Decoupled accelerators

Computing cluster with tightly coupled accelerators

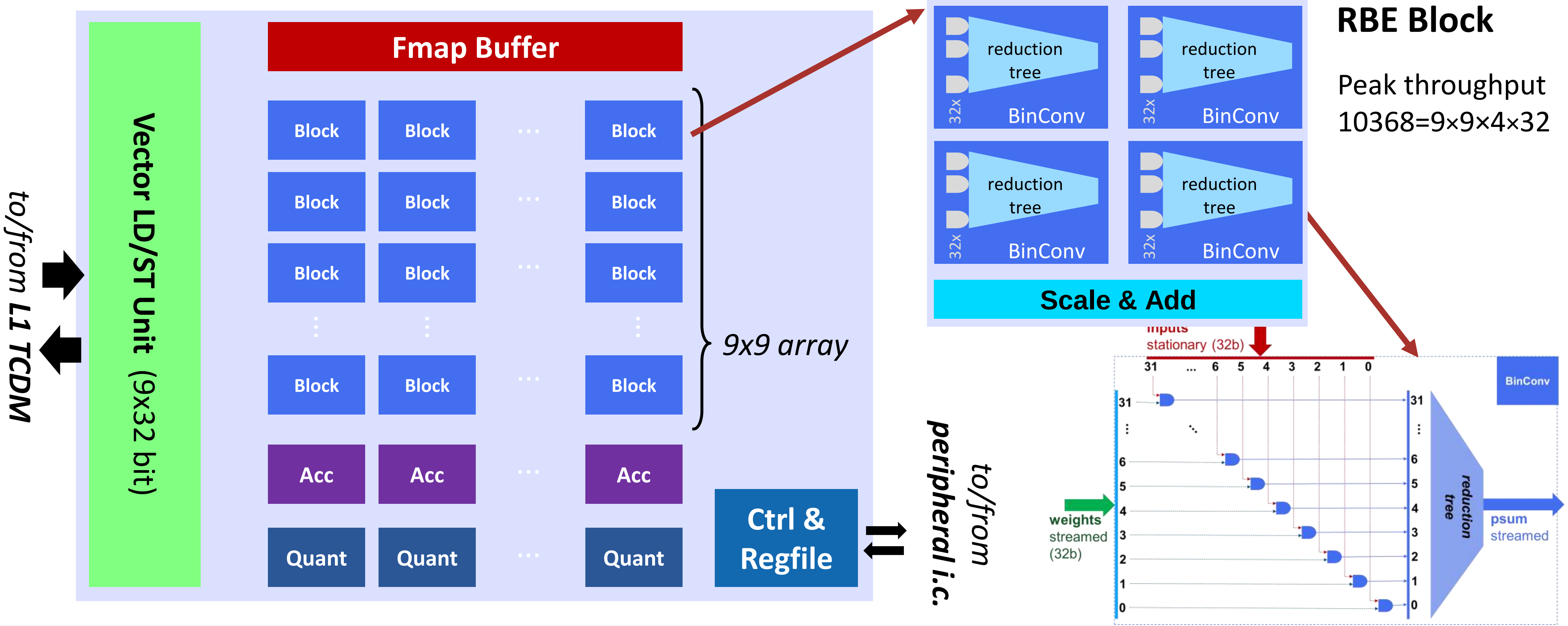
RISC-V is a key enabler → max agility, enabling SW build-up, without vendor lock-in

Tightly-coupled Accelerators



HWPE: Reconfigurable Binary Engine

$$\mathbf{y}(k_{out}) = \boldsymbol{quant}\left(\sum_{i=0..\boldsymbol{M}}\sum_{j=0..\boldsymbol{N}}\sum_{k_{in}} 2^i 2^j \left(\mathbf{W_{bin}}(k_{out},k_{in}) \otimes \mathbf{x_{bin}}(k_{in})\right)\right)$$



RBE Block

Peak throughput
10368=9×9×4×32

Energy efficiency 10-20× (0.1pJ/OP) w.r.t. SW on cluster @same accuracy

Specialization in perspective

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

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

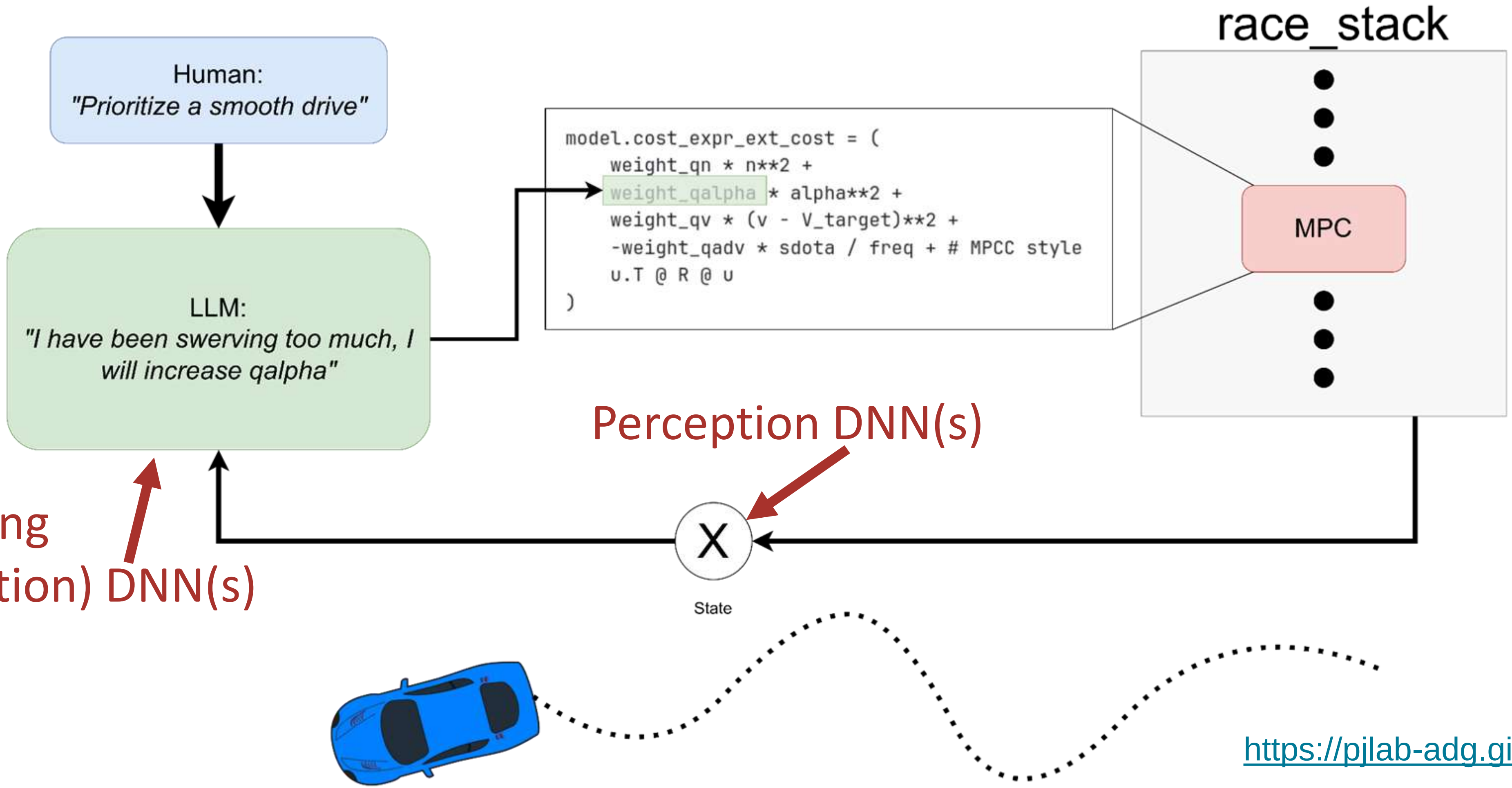
ISA-based 10-20x →**1-5pJ (8bit)** → **XPULP**

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

Highly specialized DP 10-20x →**1-5fJ (ternary)** → **CUTIE, SNN**

Beyond Perception: Reasoning with Gen.AI

LLM Reasoning on Human Commands & Robot Observations


Human:
"Prioritize a smooth drive"
LLM:
"I have been swerving too much, I will increase qalpha"
race_stack
MPC
X
State

```
model.cost_expr_ext_cost = (
    weight_qn * n**2 +
    weight_qalpha * alpha**2 +
    weight_qv * (v - V_target)**2 +
    -weight_qadv * sdota / freq + # MPCC style
    u.T @ R @ u
)
```

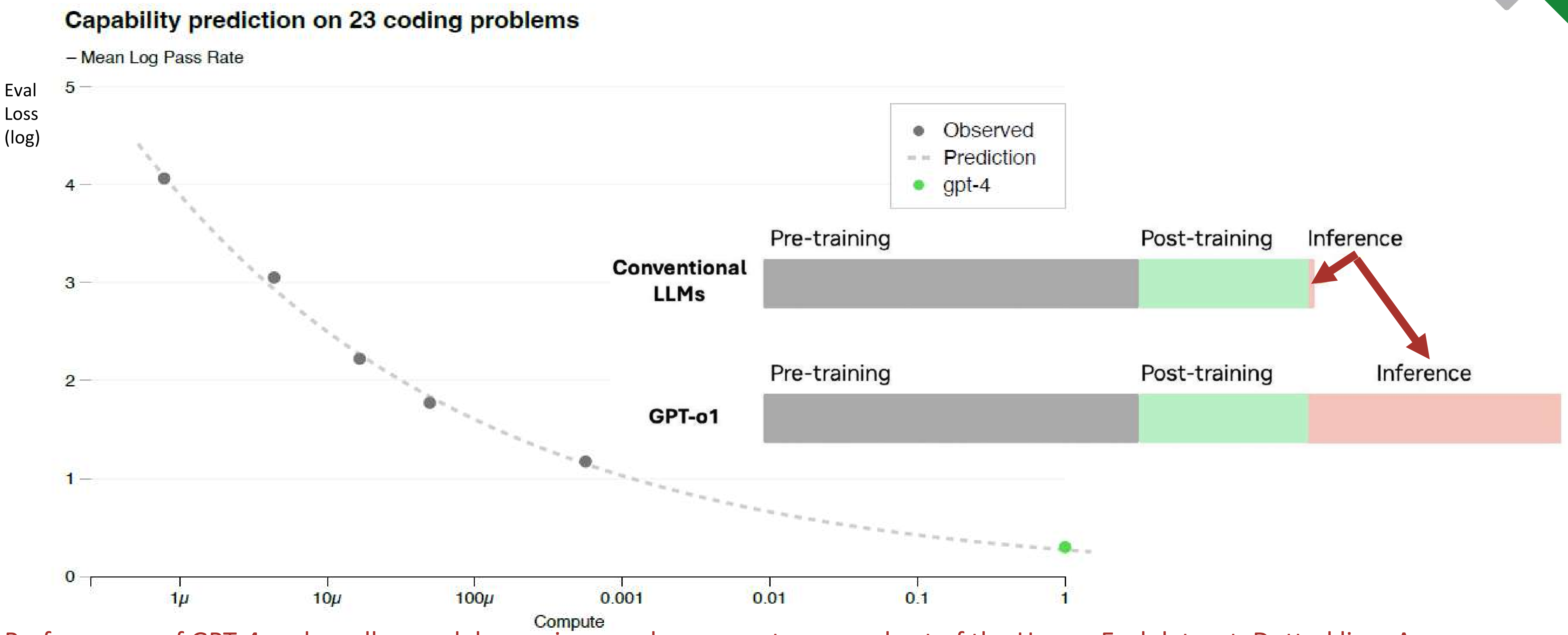
Perception DNN(s)

Reasoning (generation) DNN(s)

https://pjlab-adg.github.io/DiLu/

Pervasive Gen.AI Challenge

OpenAI’23 arXiv:2303.08774



Performance of GPT-4 and smaller models: y-axis mean log pass rate on a subset of the HumanEval dataset. Dotted line: A power law fit to smaller models (excluding GPT-4) → Accurately predicts GPT-4’s performance. x-axis is training compute (log)

There is no Othe Way to Go, but UP

Multiple Scales of acceleration

Extensions to processor cores

- Explore new extensions
- Efficient implementations

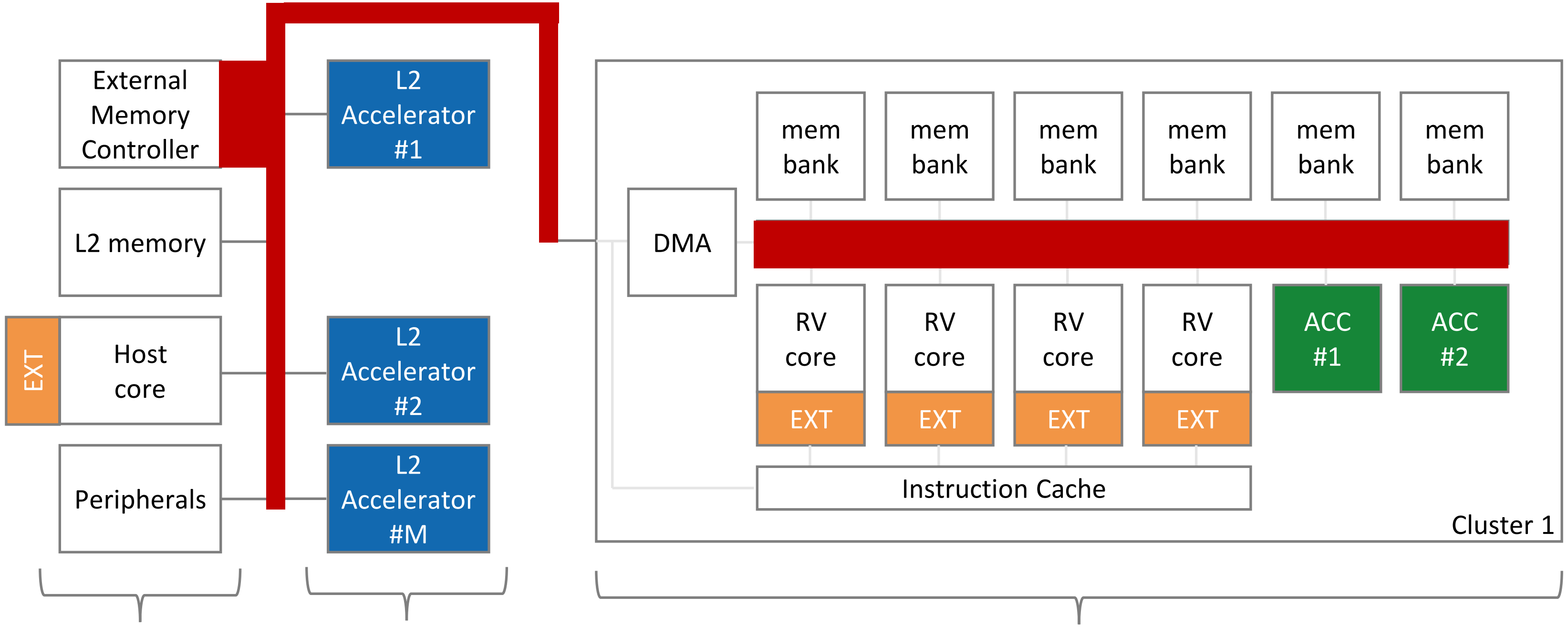
Shared-memory Accelerators

- Domain specific
- Local memory

Multiple Decoupled Accelerators

- Communication
- Synchronization

Local, Global, Off-Chip Interconnect



Host, L2, L3 IOs

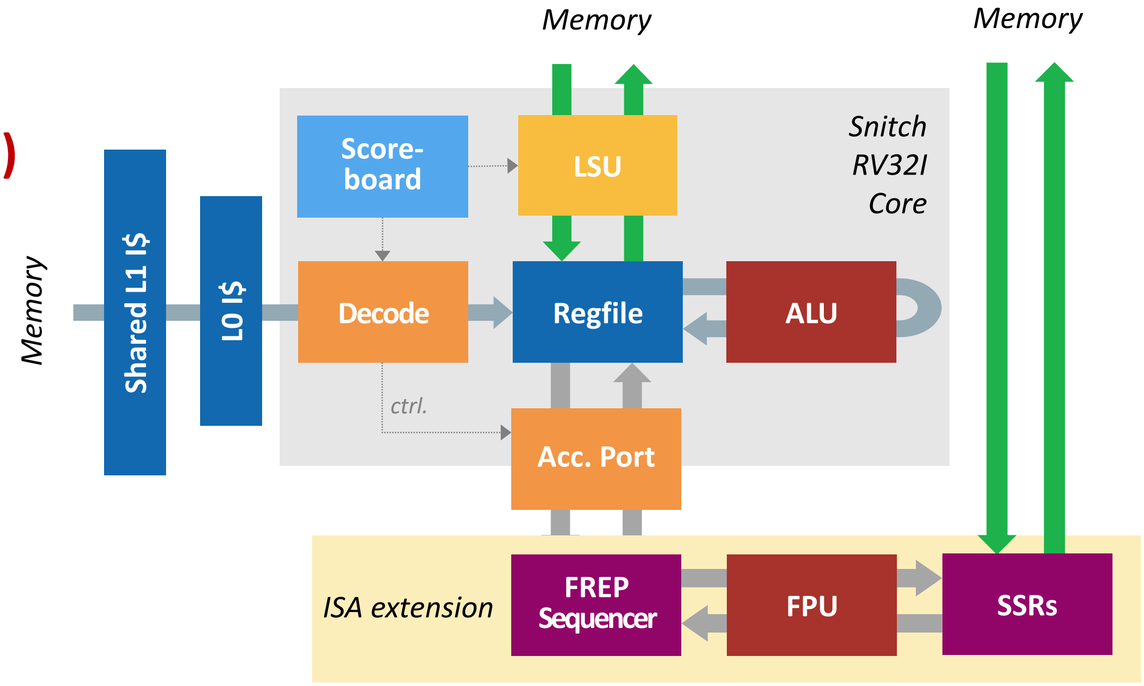
Decoupled accelerators

Computing cluster with tightly coupled accelerators

Specialize interconnects too! Local, global, package, system

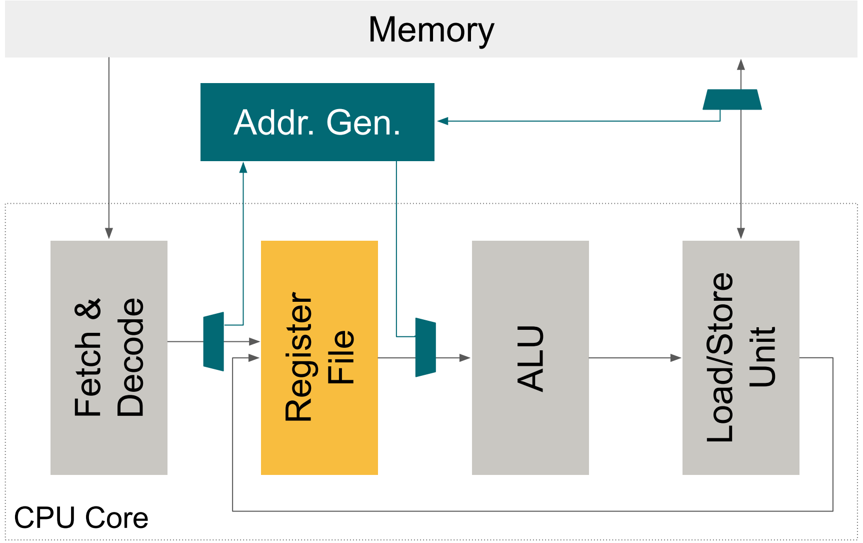
Snitch Core: Latency Tolerant, Extensible RV PE

- ***Snitch*: tiny (20KGE), extensible RV core**
  - *Extensible* through **accelerator port**
  - *Latency-tolerant* through scoreboard+ld/st queue
    → can issue ~10 non-blocking memOPs
  - **Tolerates 10 cycles of memory latency (Little's law)**
- **Paired with *ISA extension subsystem***
- ***Native streaming support***
  - Load/store elision
  - Reduction of I$ pressure



SSR & FREP: Streaming Extension

- **SSR**: Link register read/writes into implicit LD/ST
  - Extension around the core's register file
  - Address generators (2-3KGE/SSR)
  - Configured out of inner loop (LD/ST elision)
  - Staggering: generators prefetch from memory (latency tolerant!)
- **FREP**: L0 instruction buffer (no I$ access)
  - Pseudo-dual issue (Int pipeline can proceed in parallel)
  - No boundary checking for loop (similar HW loop in DSPs)
- **Boost FPU utilization → 100% (once setup is amortized)**

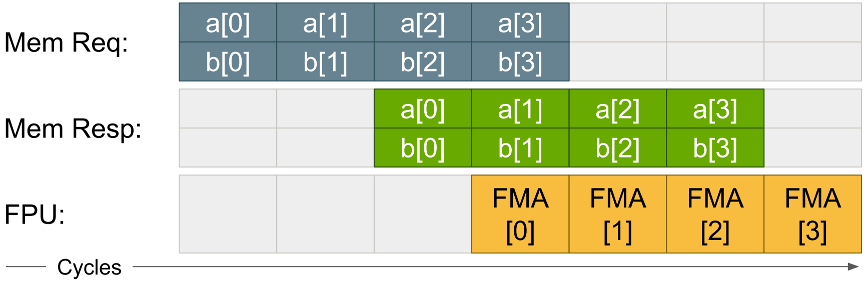


dotp: 30% FPU

```
loop:
fld r0, %[a]
fld r1, %[b]
fmadd r2, r0, r1
```

dotp: 90% FPU

```
scfg 0, %[a], ldA
scfg 1, %[b], ldB
loop:
fmadd r2, ssr0, ssr1
```

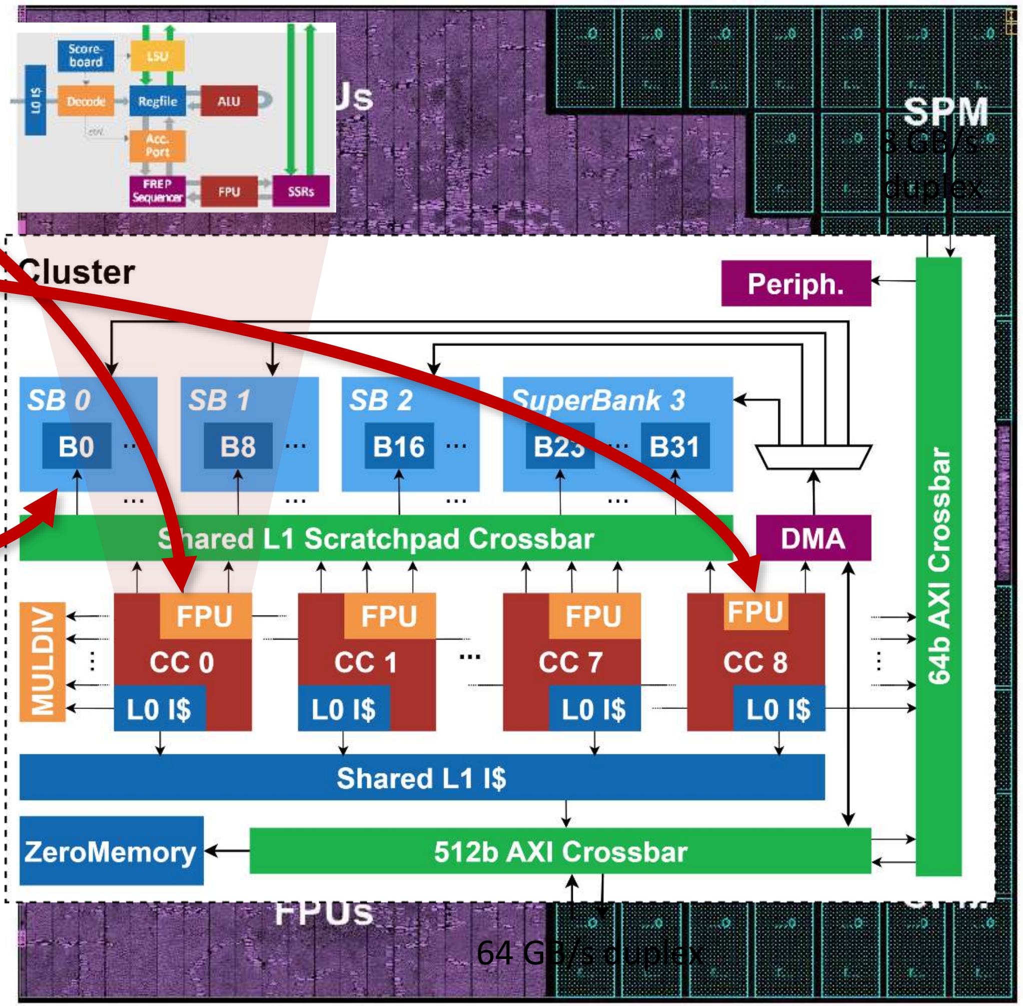


Mem Req:	a[0] b[0]	a[1] b[1]	a[2] b[2]	a[3] b[3]			
Mem Resp:			a[0] b[0]	a[1] b[1]	a[2] b[2]	a[3] b[3]	
FPU:				FMA [0]	FMA [1]	FMA [2]	FMA [3]

Latency Tolerance: Less expensive than OoO (CPU) and Multi-threading (GPU)

Snitch Cluster: The Fundamental Compute Block

- **8 Snitch compute cores**
  - SIMD 64b FPU with SSRs & FREP
- **9th Core: DMA engine**
  - 512b interface to interconnect
  - HW support for autonomous ≤ 2D transfers, higher dimensions through SW
  - *Latency-tolerance block transfers* (100s of cycles)
- **128 KiB TCDM**
  - 32-bank, low-latency shared scratchpad
  - Double-buffer large chunks (KBs) with DMA
- **Shared TCDM, I-cache and peripherals**

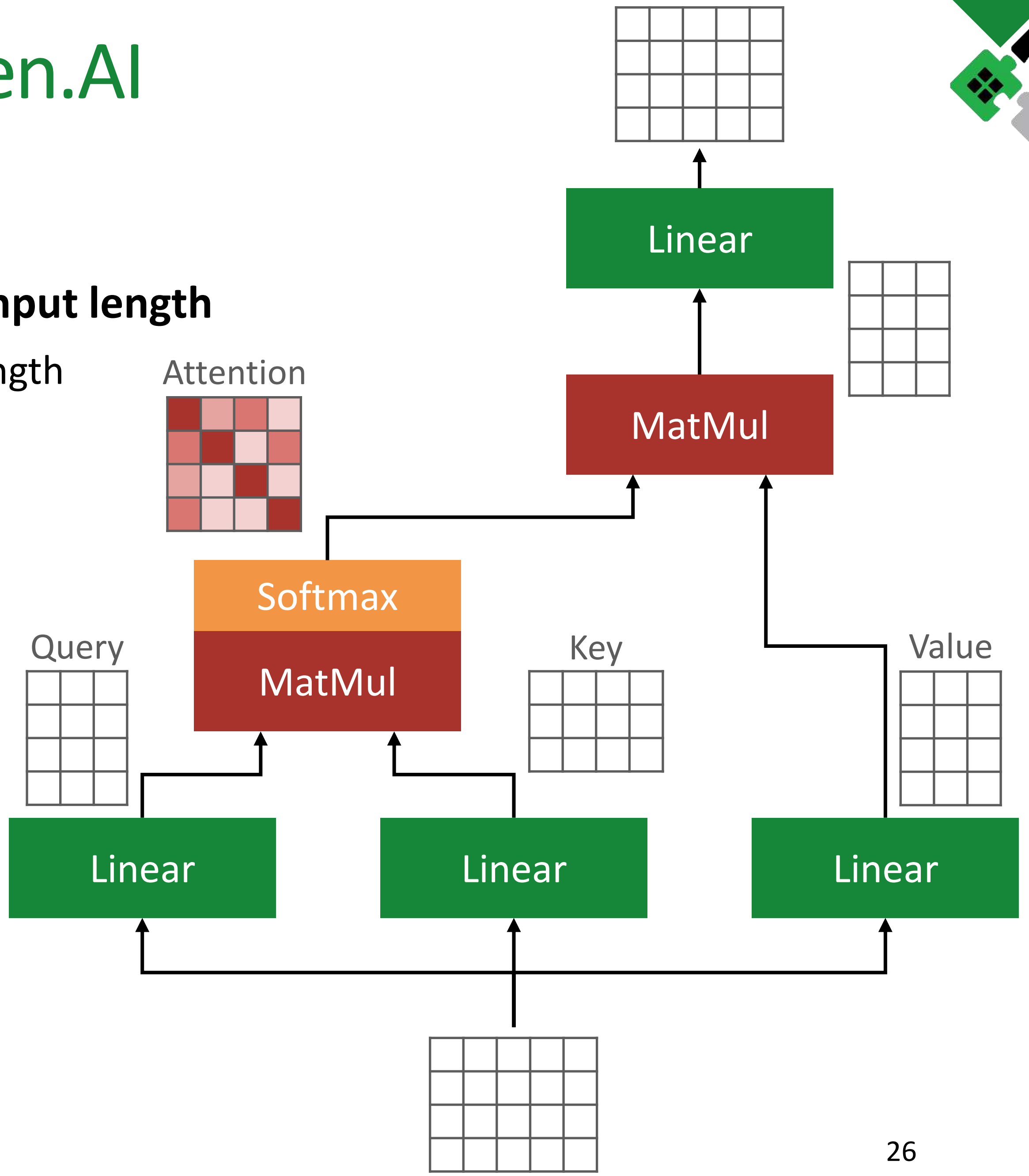


- **Shared DMA (10% overhead) for latency tolerance of L2+→L1: 100s vs 10s cycles**

Specializing the Cluster for Gen.AI

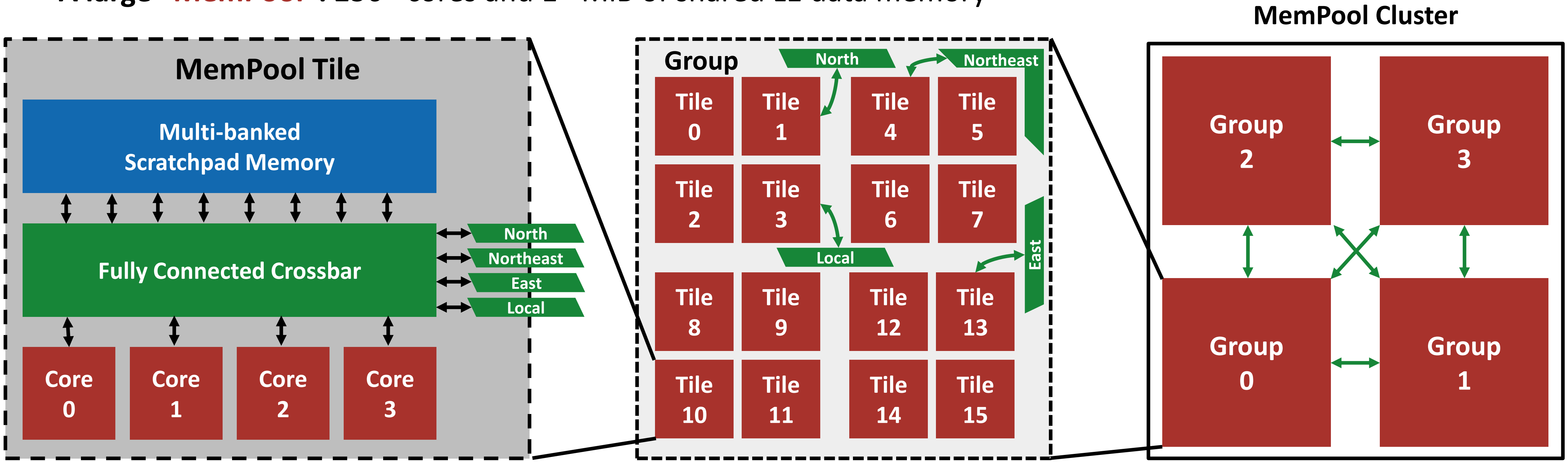
- **Attention is key**
- **Attention matrix is a square matrix of order input length**
  - Quadratic memory requirement vs. sequence length
  - No asymmetry between operands ("weightless")
- **MatMul & Softmax dominate**

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



Matmul Benefits from Large Shared-L1 clusters

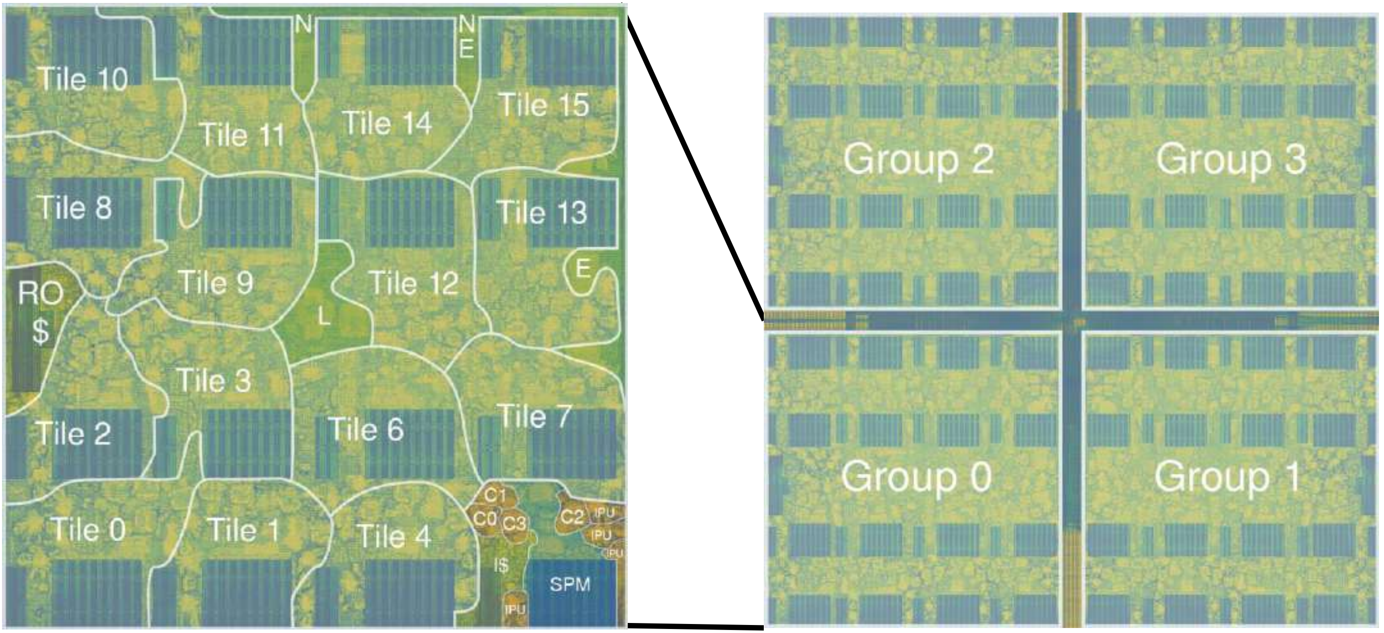
- **Why?**
  - Better global latency tolerance if **$L1_{size} > 2\times L2_{latency} \times L2_{bandwidth}$** (Little's law + double buffer)
  - Smaller data partitioning overhead
  - Larger Compute/Boundary bandwidth ratio: **N^3/N^2** for MMUL grows linearly with N!
- **A large "MemPool":** 256+ cores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



MemPool Cluster: A physical-aware design

MemPool
&
Terapool

- **A Scalable Manycore Architecture with Low-Latency Shared L1 Memory**
  - 256+ cores
  - 1+ MiB of shared L1 data memory
  - ≤ 8 cycle latency (Snitch can handle it)
- **Hierarchical design**
- **Implemented in GF22**
  - Targeting 500 MHz (SS/0.72V/125°C)
  - Reaching 600 MHz (TT/0.80V/25°C)
  - Targeting iso-frequency with PULP
- **Cluster area of 13 mm²**
  - 5 mm diagonal
  - Round trip in 5 cycles
- **Terapool: 1024 Cores!**



MemPool Group

MemPool Cluster

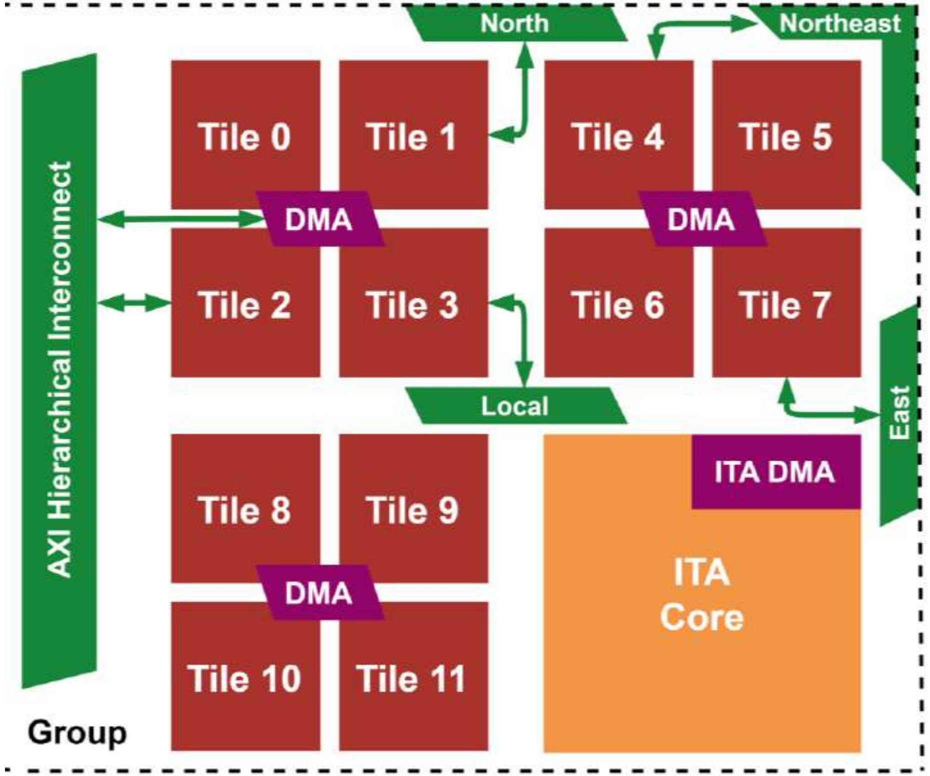
MemPool + Integer Transformer Accelerator (ITA)

Tightly coupled Acceleration Enginee

- Matmul & Softmax
- Reduce pressure on memory and interconnect

Collaborative Execution

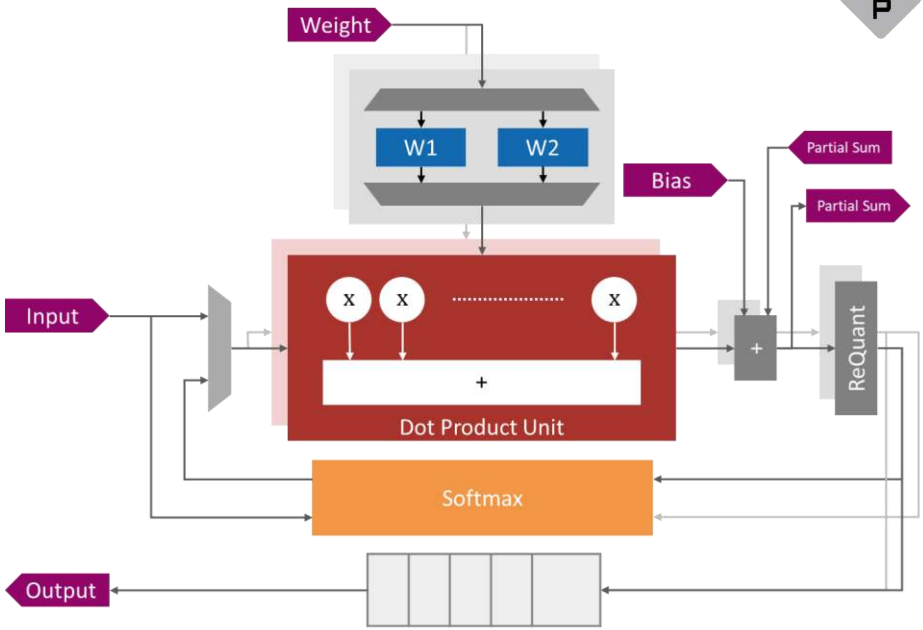
- Cores prepare activations for the next attention head
- Final head accumulation computed in cores
- Nonlinearity in cores (PACE)

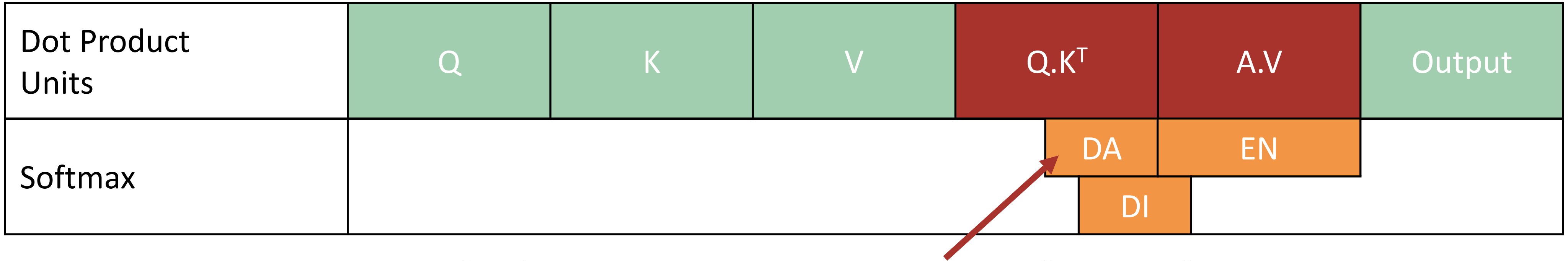


MemPool + Integer Transformer Accelerator (ITA)

Integer Attention Accelerator

- 8-bit inputs, weights & outputs
- Builtin data marshaling & pipelined operation
- Streaming partial Softmax **adding no additional latency**
- Fused Q× K^T, Softmax and A × V computation
- Support for hardware-aware Softmax approximation in QuantLib





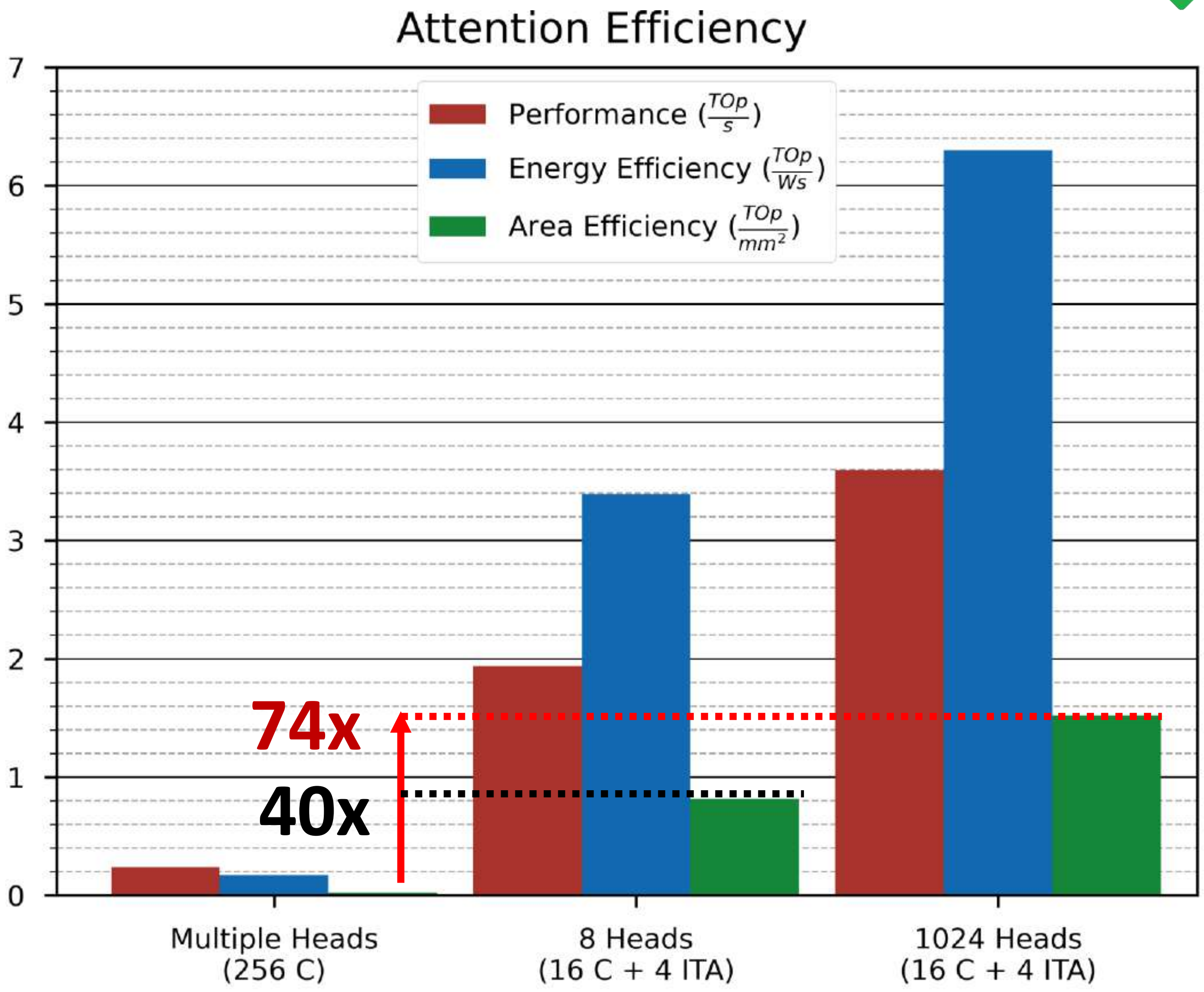
$$e^{a_i - a_{max_{n+1}}} = e^{a_i - a_{max_n}} \cdot e^{a_{max_n} - a_{max_{n+1}}}$$

Attention on ITA

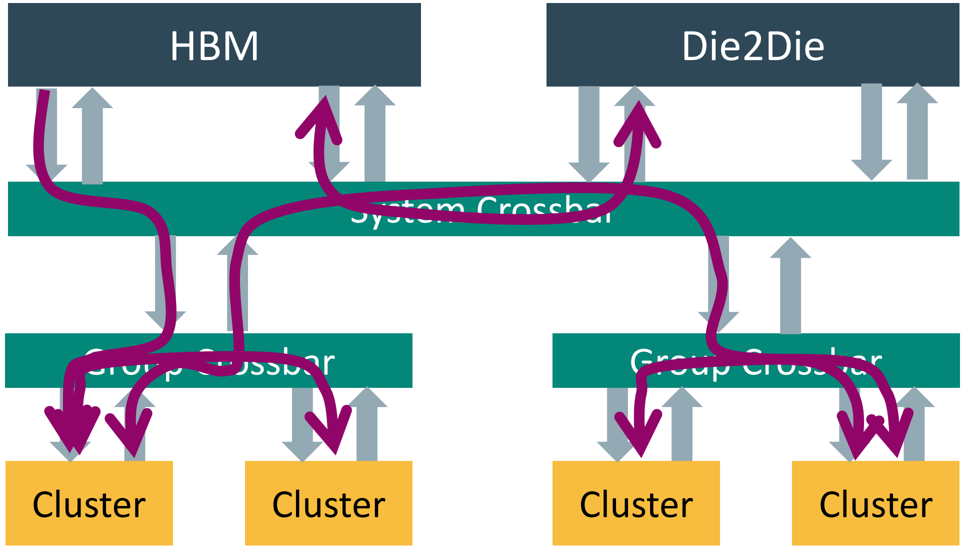
Performance
increase of **15x**

Energy Efficiency
increase of **36x**

Area Efficiency
increase of **74x**



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

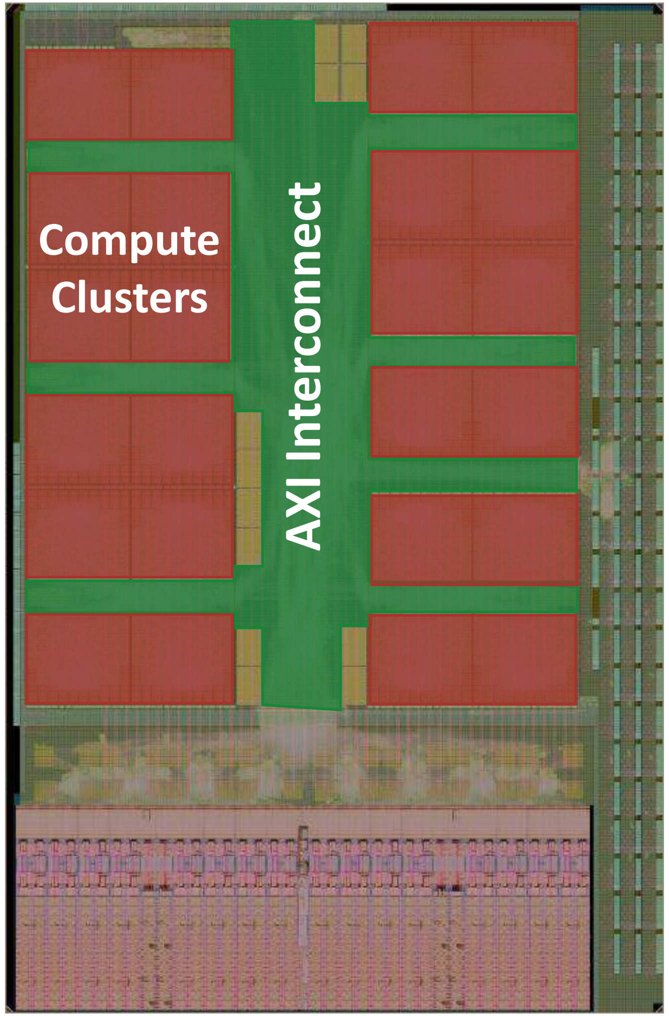
- Between **clusters**
- Across **groups**

Smartly distribute workload

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

Big trend!

Addressing interconnect scalability



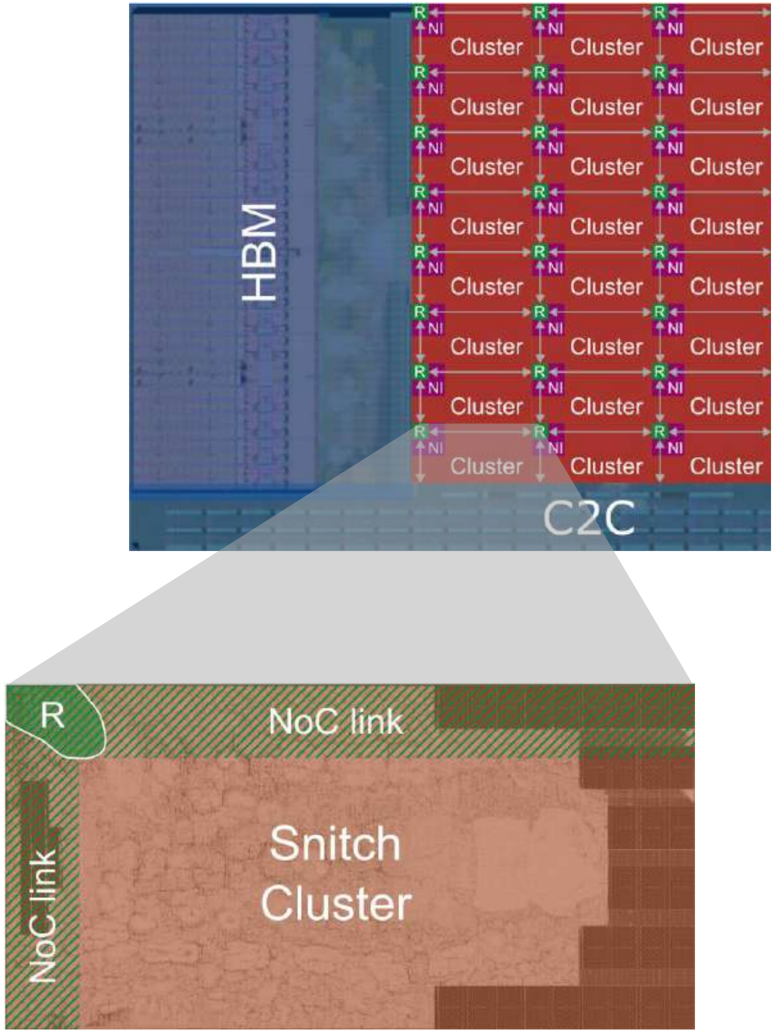
- **Fat-tree was very challenging in Implementation**
  - AXI has severe scalability issues
  - Top-level Xbar had to be split up
  - Still, interconnect takes up almost **40%***

- **Working on NoC solution, *FlooNoC***
  - Fully AXI4 compatible
  - Solves AXI4 **scalability issues**
  - Designed with awareness of physical design
  - **Wide** & **physical** channels

**HBM & C2C excluded*

Replacing the AXI interconnect with a NoC

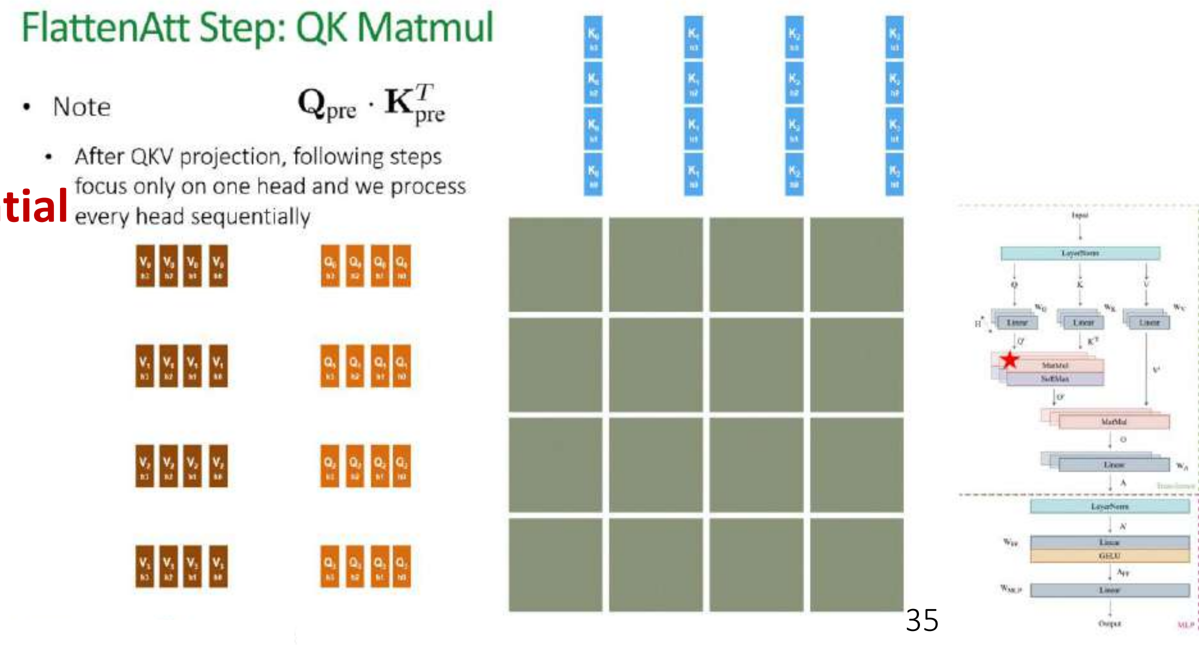
- **Potential for big area/performance gains**
  - Only **~10%** interconnect area
  - **66%** more clusters, same floorplan
  - *High Bandwidth*: **629Gbps/link**
  - *High Energy-Efficiency*: **0.19pj/B/hop**



MHA Mapping on NoC: FlattenAttention

- Proposed Dataflow Schedule of MHA
  - We leverage all-cluster L1 for single head attention – Minimize I/O complexity
  - Gen.AI specialized NoC
    - Matrix transpose engine for transposition of (K -> K^T)
    - Collective operations on NoC
- Benchmark & Results
  - 16x16 Clusters (8TFLOPS, 256kB L1), 2TB/s HBM
  - One layer MHA of Llama3-70B (seq=4K, batch=8)
  - **Efficient collective operation support on NoC is essential**
    - **3x speedup to baseline**

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

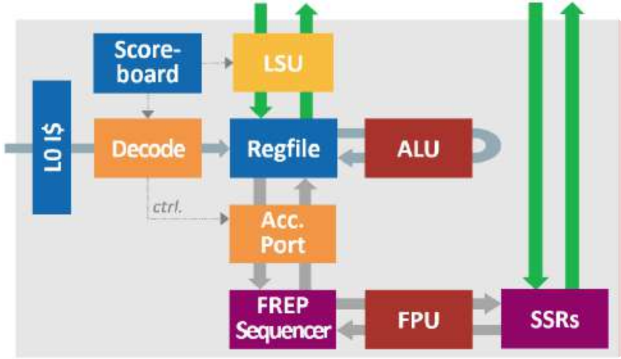


FlattenAtt Step: QK Matmul

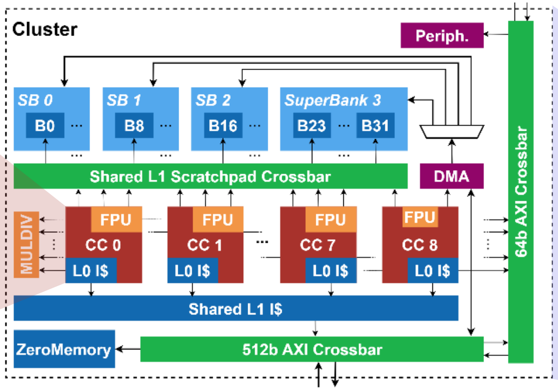
- Note $\mathbf{Q}_{\text{pre}} \cdot \mathbf{K}_{\text{pre}}^T$
  - After QKV projection, following steps focus only on one head and we process every head sequentially

Scaling UP: From Chip to chiplets

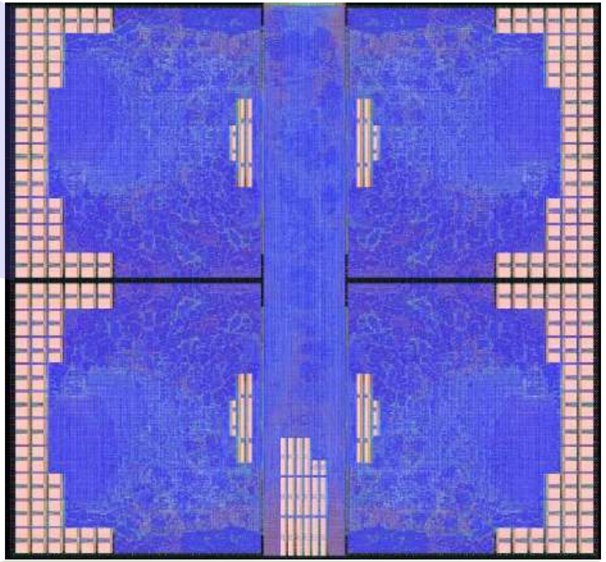
Snitch Core



Snitch Cluster

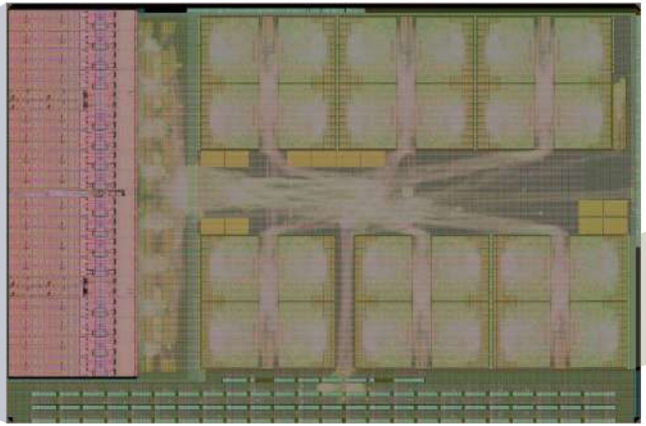


Occamy Group

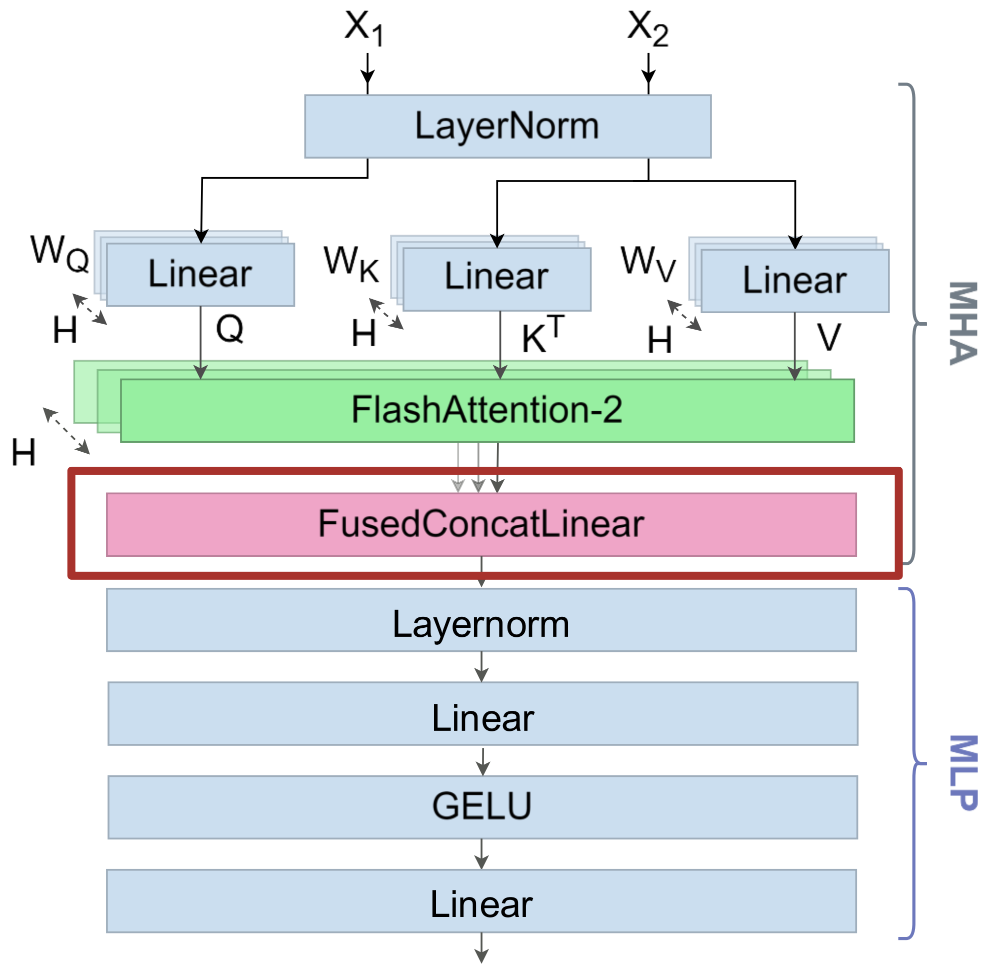


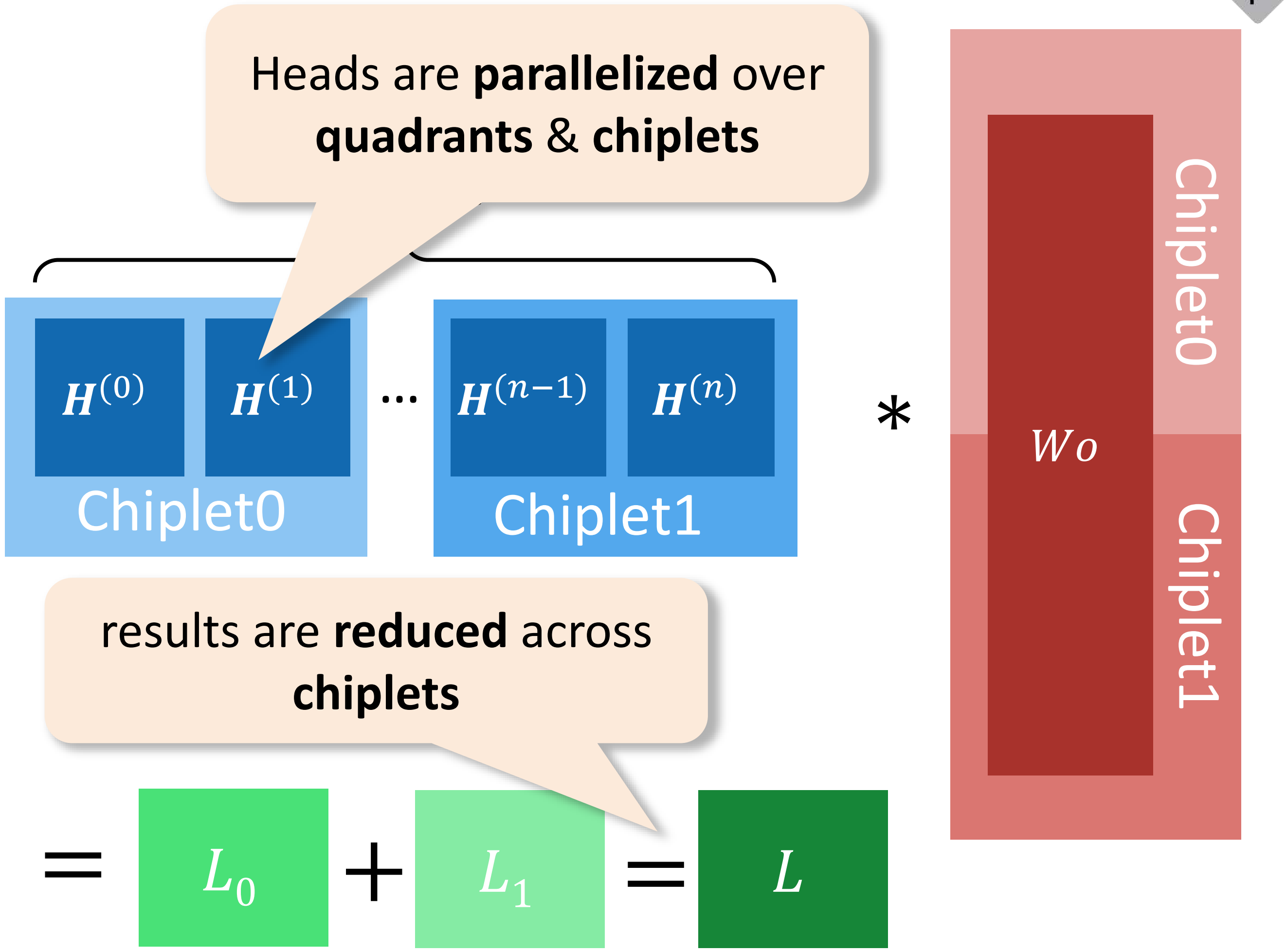
Occamy System

Occamy Chiplet



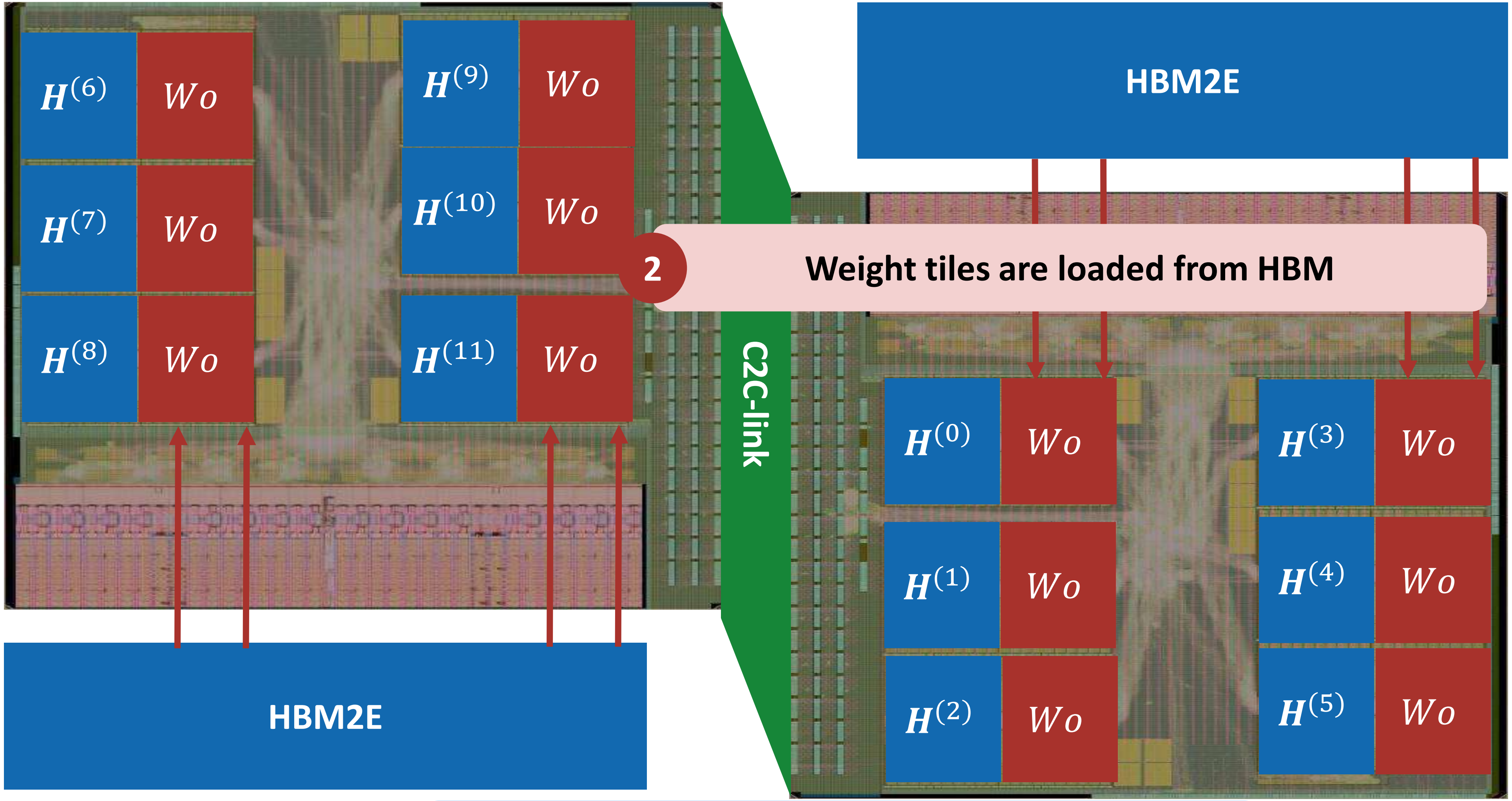
Not Only Layer-by-Layer distribution across Chiplets!




Heads are **parallelized** over **quadrants** & **chiplets**

results are **reduced** across **chiplets**

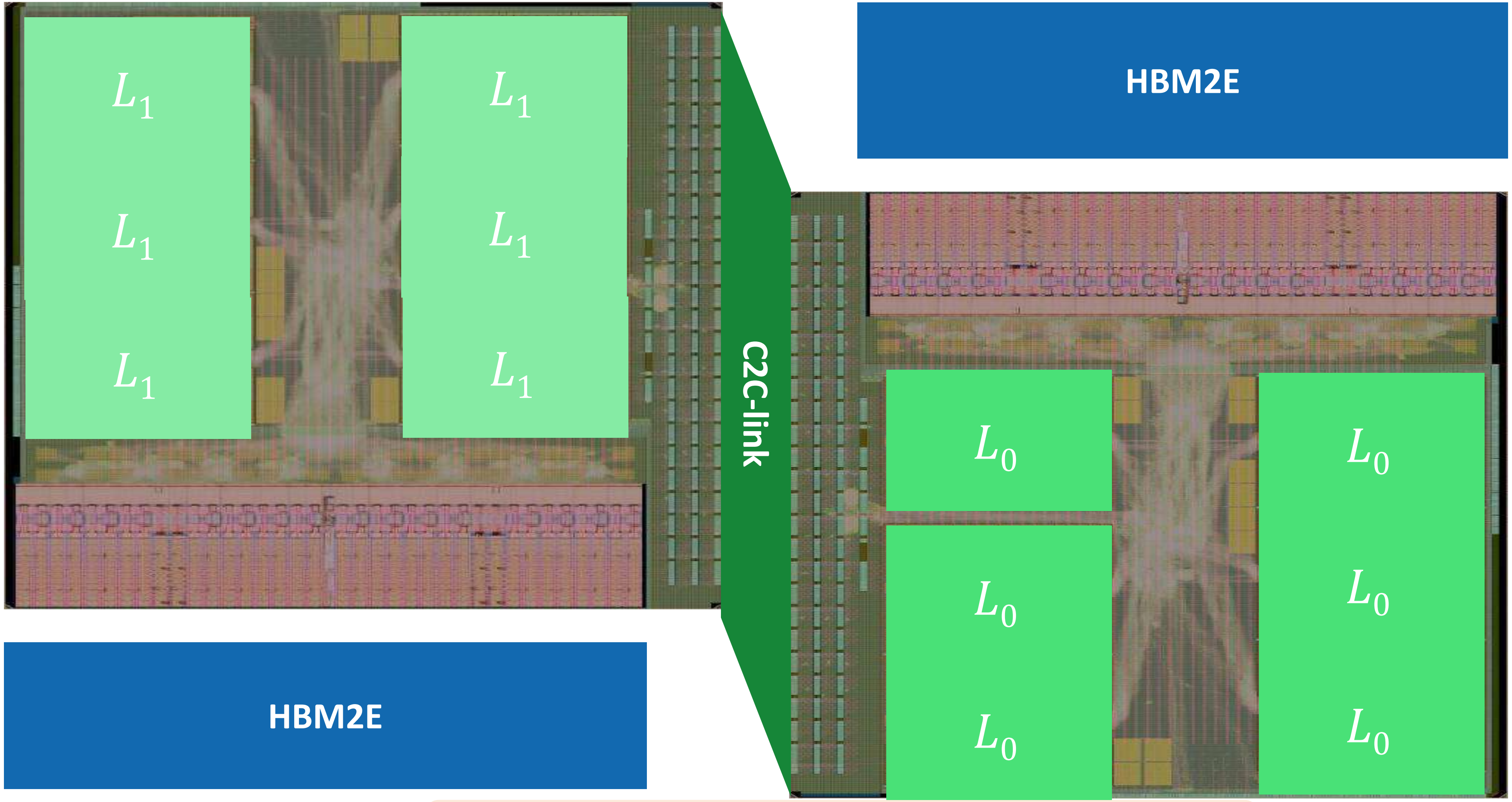
Linear Projection & Head Concatenation Fusion



2 Weight tiles are loaded from HBM

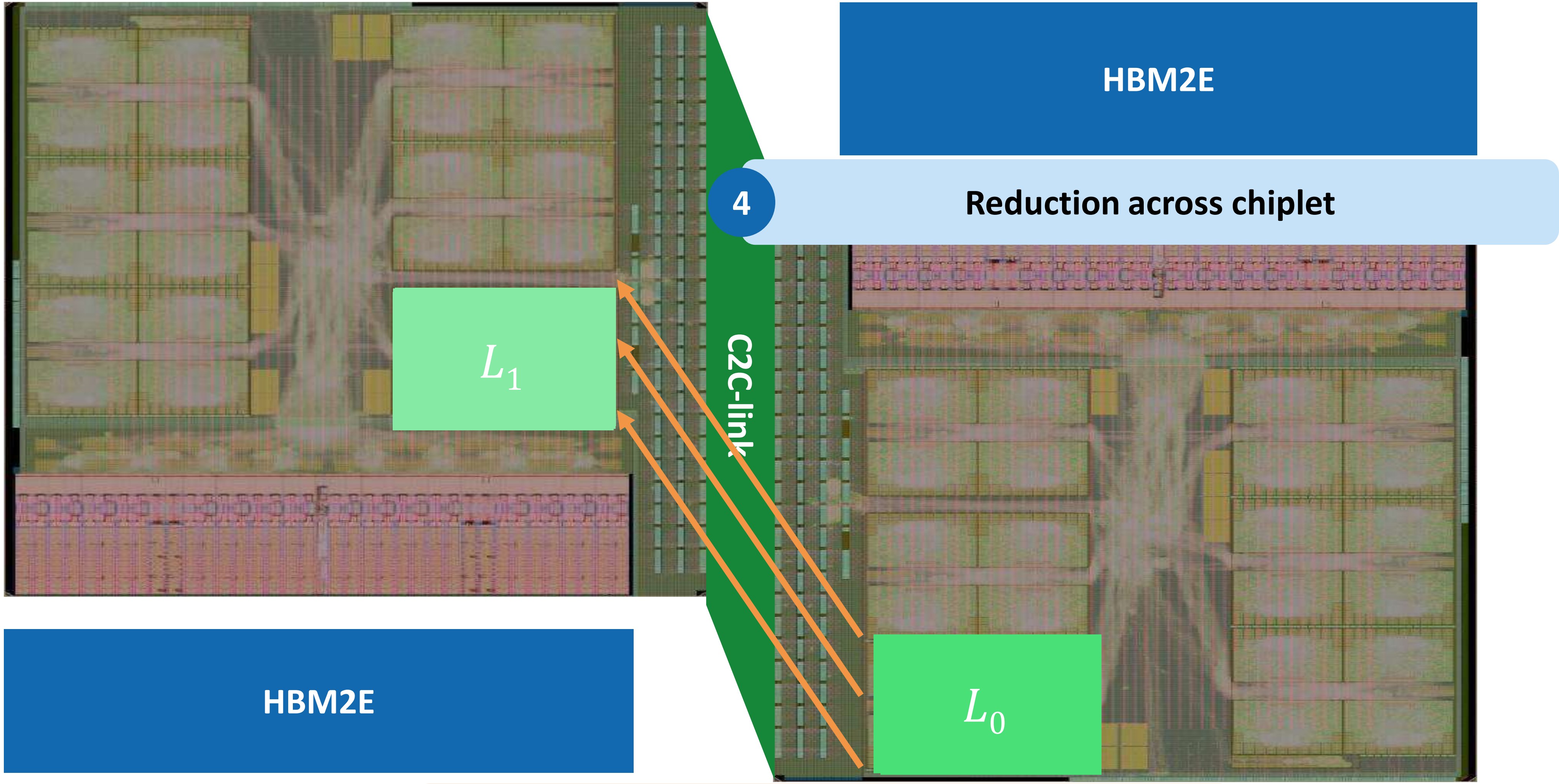
1 Heads are already stored in clusters

Linear Projection & Head Concatenation Fusion



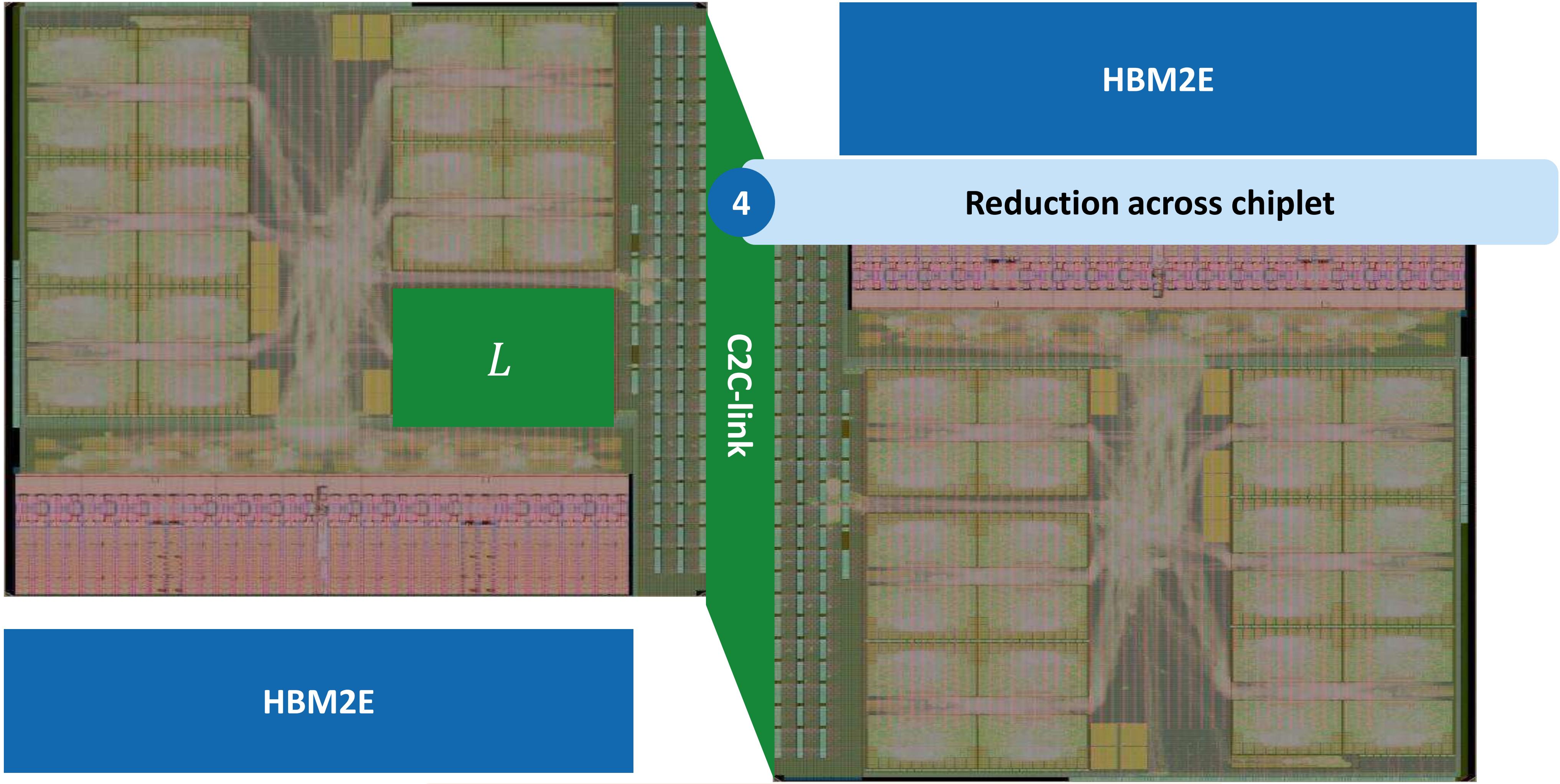
3 **Logarithmic-tree result reduction on-chiplet**

Linear Projection & Head Concatenation Fusion



3 Logarithmic-tree result reduction on-chiplet

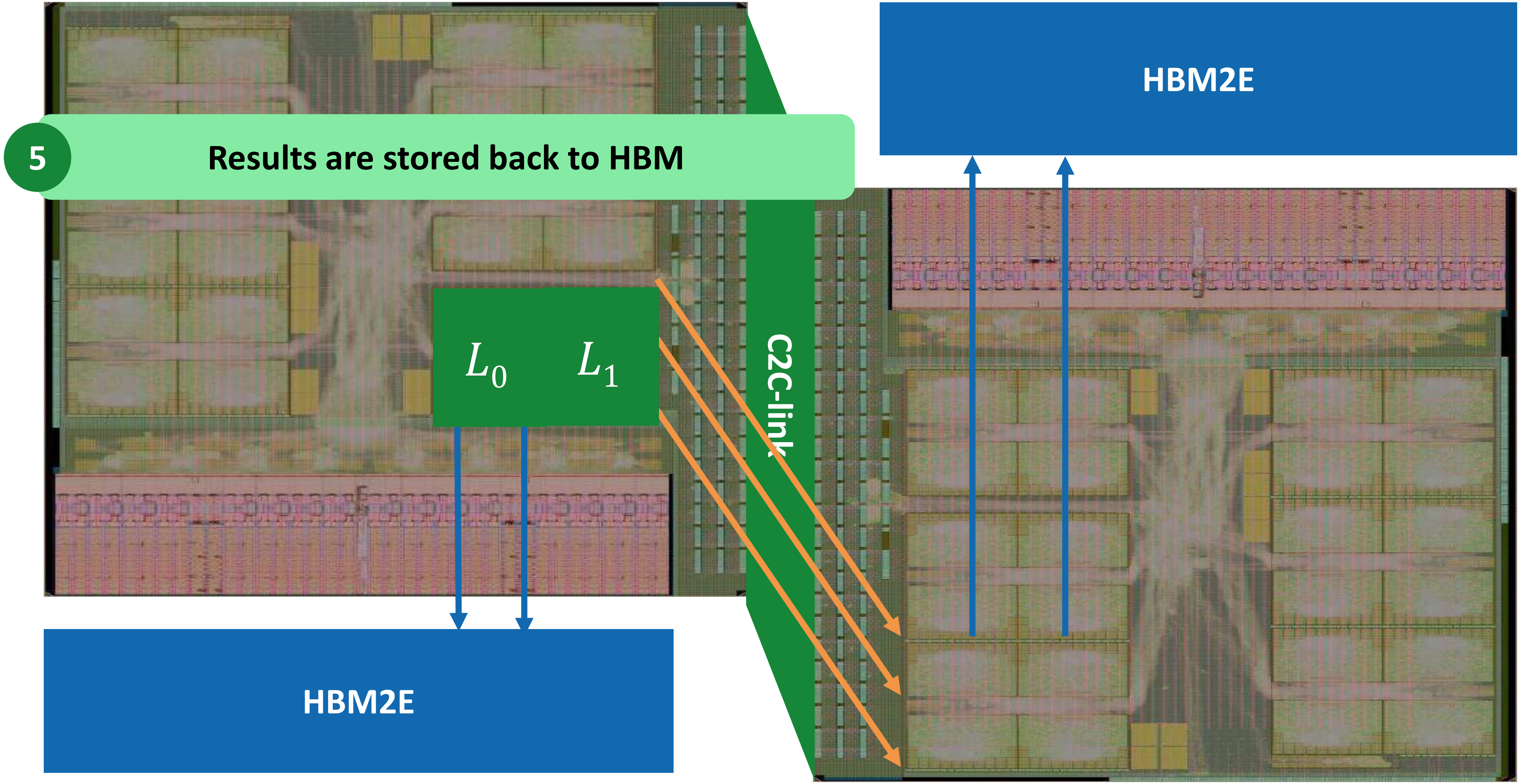
Linear Projection & Head Concatenation Fusion



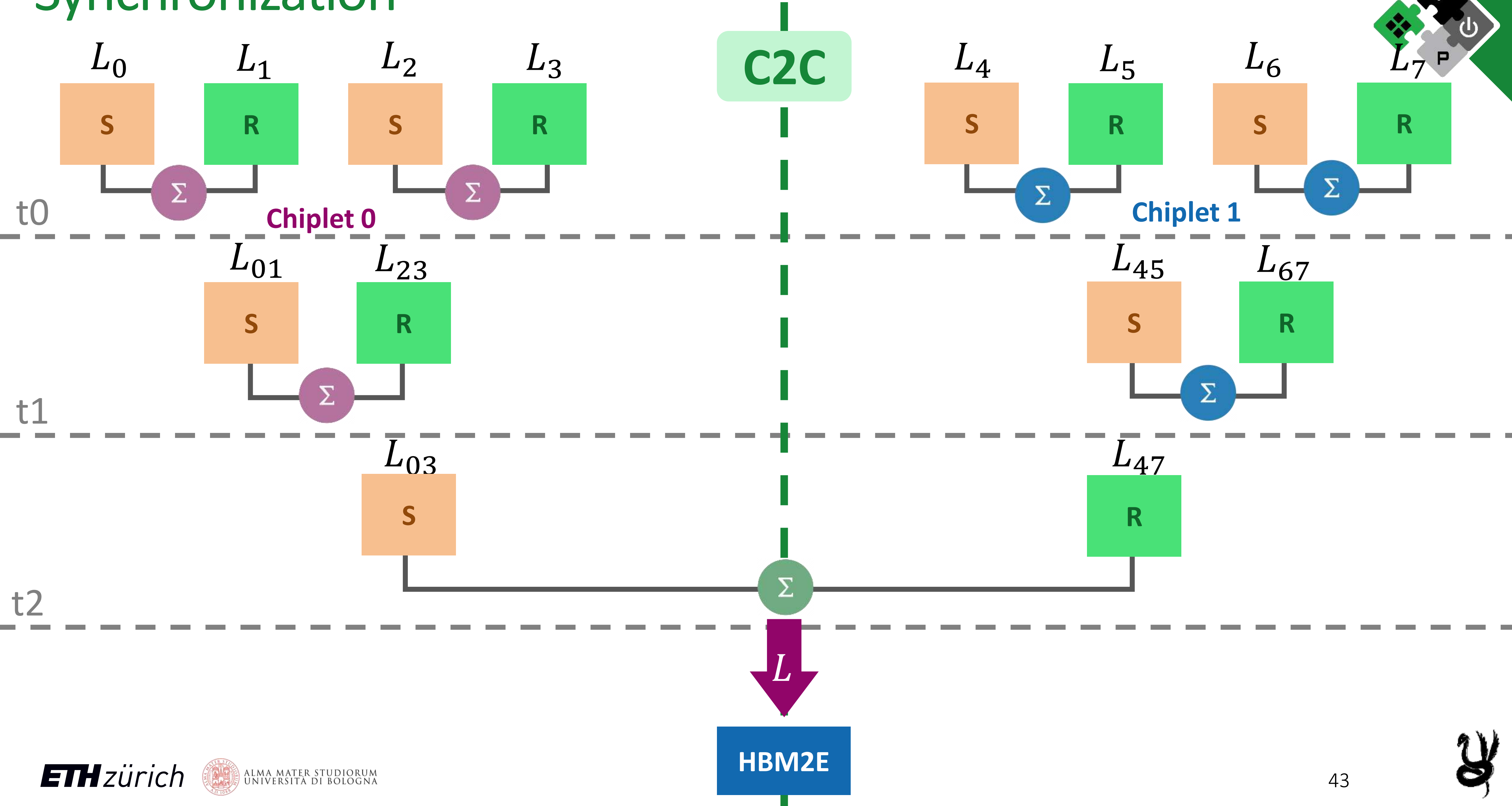
4 Reduction across chiplet

3 Logarithmic-tree result reduction on-chiplet

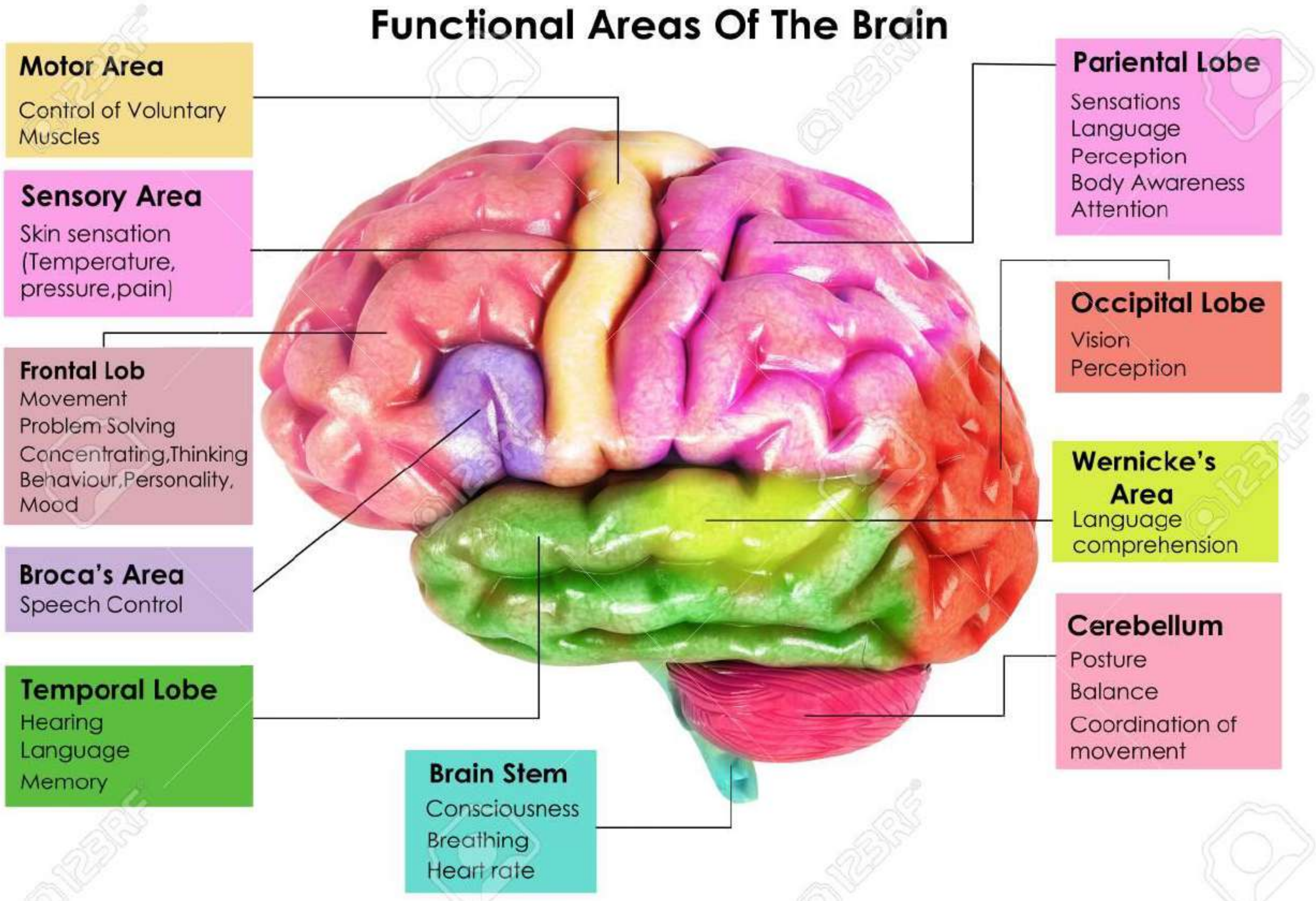
Linear Projection & Head Concatenation Fusion



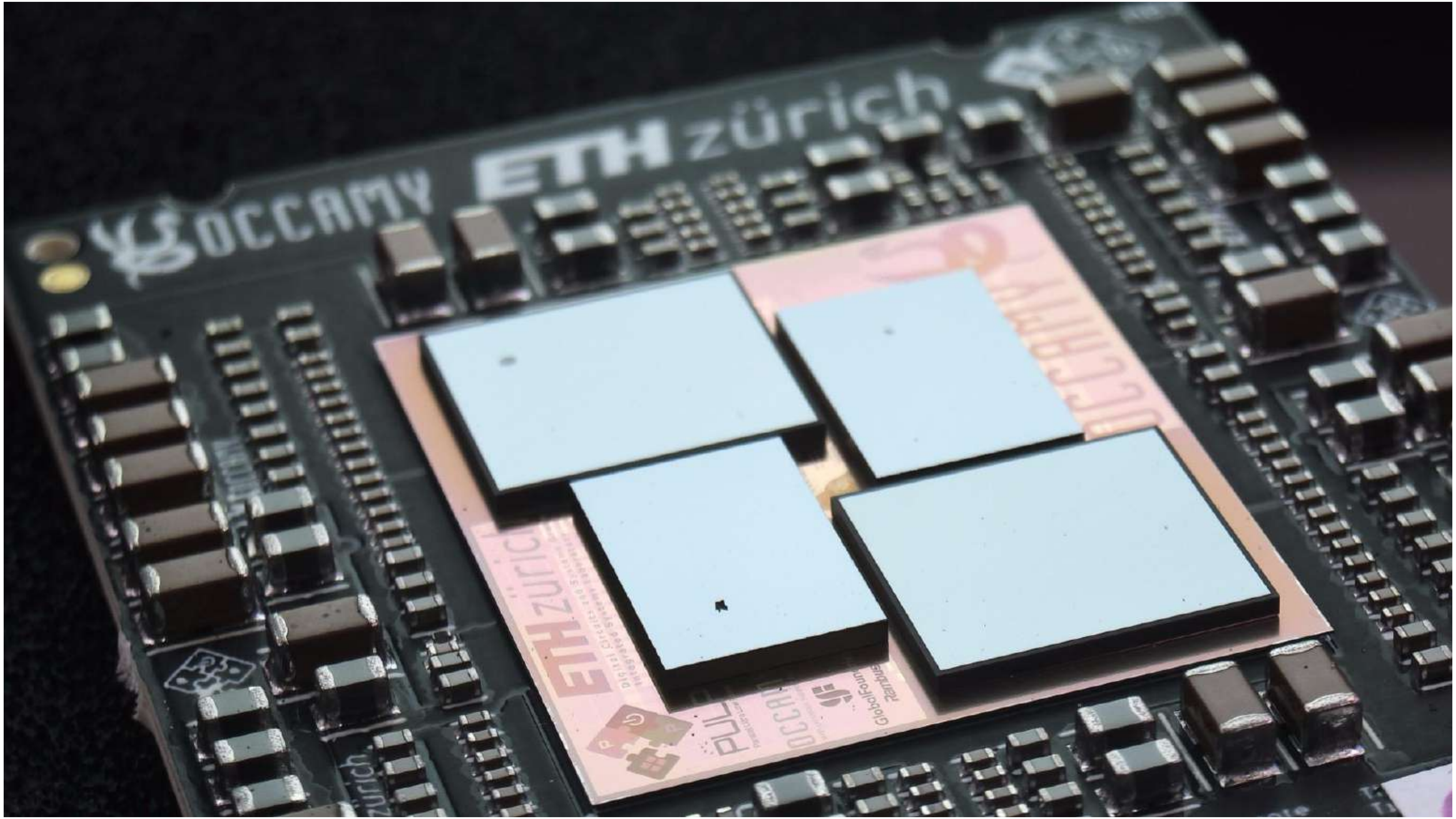
Synchronization



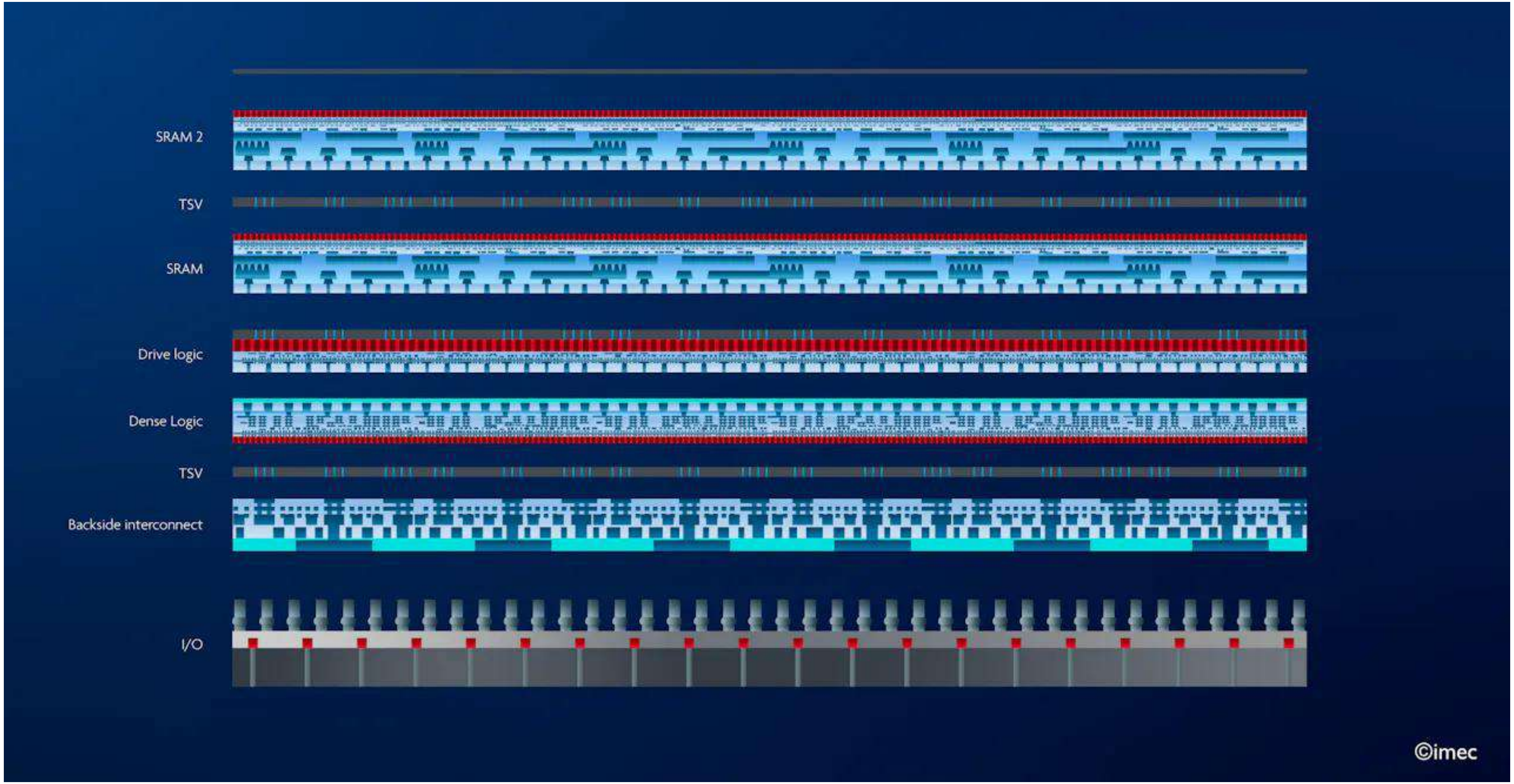
What’s next?


Functional Areas Of The Brain
Motor Area
Control of Voluntary Muscles
Sensory Area
Skin sensation (Temperature, pressure,pain)
Frontal Lob
Movement
Problem Solving
Concentrating,Thinking
Behaviour,Personality,
Mood
Broca’s Area
Speech Control
Temporal Lobe
Hearing
Language
Memory
Brain Stem
Consciousness
Breathing
Heart rate
Pariental Lobe
Sensations
Language
Perception
Body Awareness
Attention
Occipital Lobe
Vision
Perception
Wernicke’s Area
Language comprehension
Cerebellum
Posture
Balance
Coordination of movement

What’s next?


OCCAMY
ETH zürich
OCCAMY
ETH zürich
PULP
OCCAMY
GlobalFoundries

What’s next?


SRAM 2
TSV
SRAM
Drive logic
Dense Logic
TSV
Backside interconnect
I/O
©imec

Thank You!