ETH zürich
ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Many Shades of TinyML Acceleration

a RISC-V open platform approach

Luca Benini (lbenini@ethz.ch, luca.Benini@unibo.it)

Horizon Europe
THE NEXT EU RESEARCH & INNOVATION PROGRAMME (2021 – 2027)
erc
European Research Council
EuroHPC
Joint Undertaking
FNSNF
FONDS NATIONAL SUISSE
SCHWEIZERISCHER NATIONALFONDS
FONDO NAZIONALE SVIZZERO
SWISS NATIONAL SCIENCE FOUNDATION

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

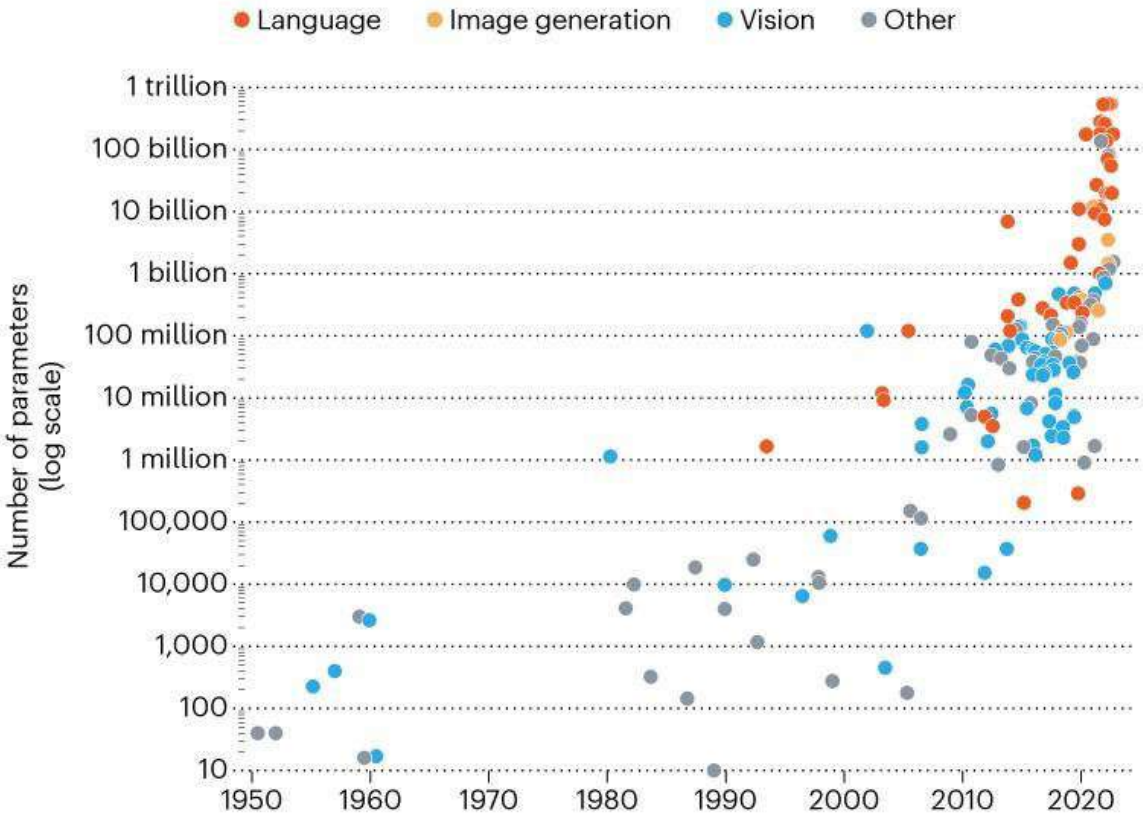
@pulp_platform
pulp-platform.org
youtube.com/pulp_platform

AI, ML Disruption: Technology cannot Keep the Pace

THE DRIVE TO BIGGER AI MODELS

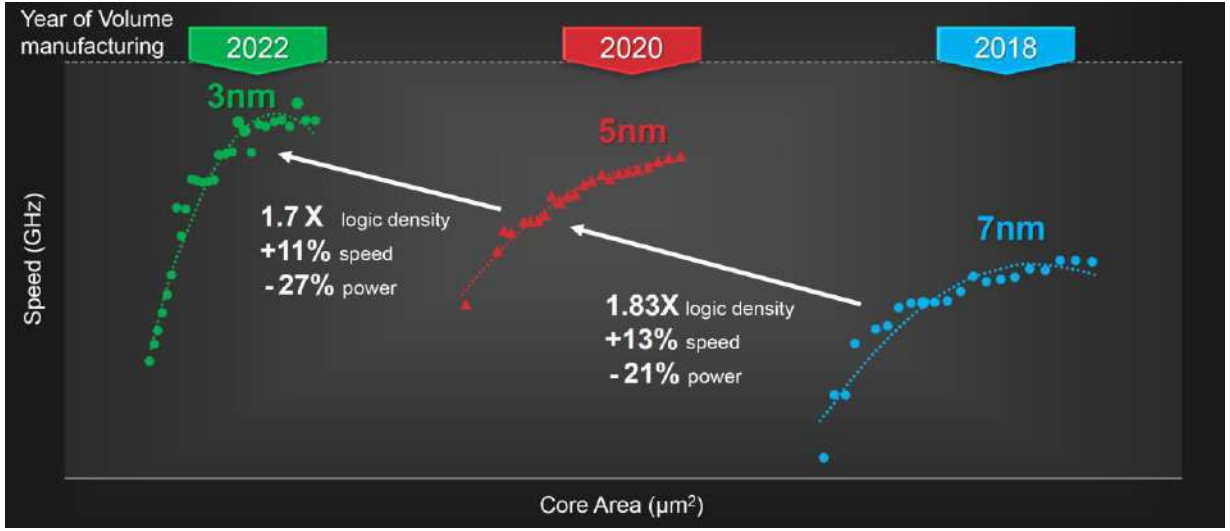
10x every 2 years

The scale of artificial-intelligence neural networks is growing exponentially, as measured by the models' parameters (roughly, the number of connections between their neurons)*.



*'Sparse' models, which have more than one trillion parameters but use only a fraction of them in each computation, are not shown.

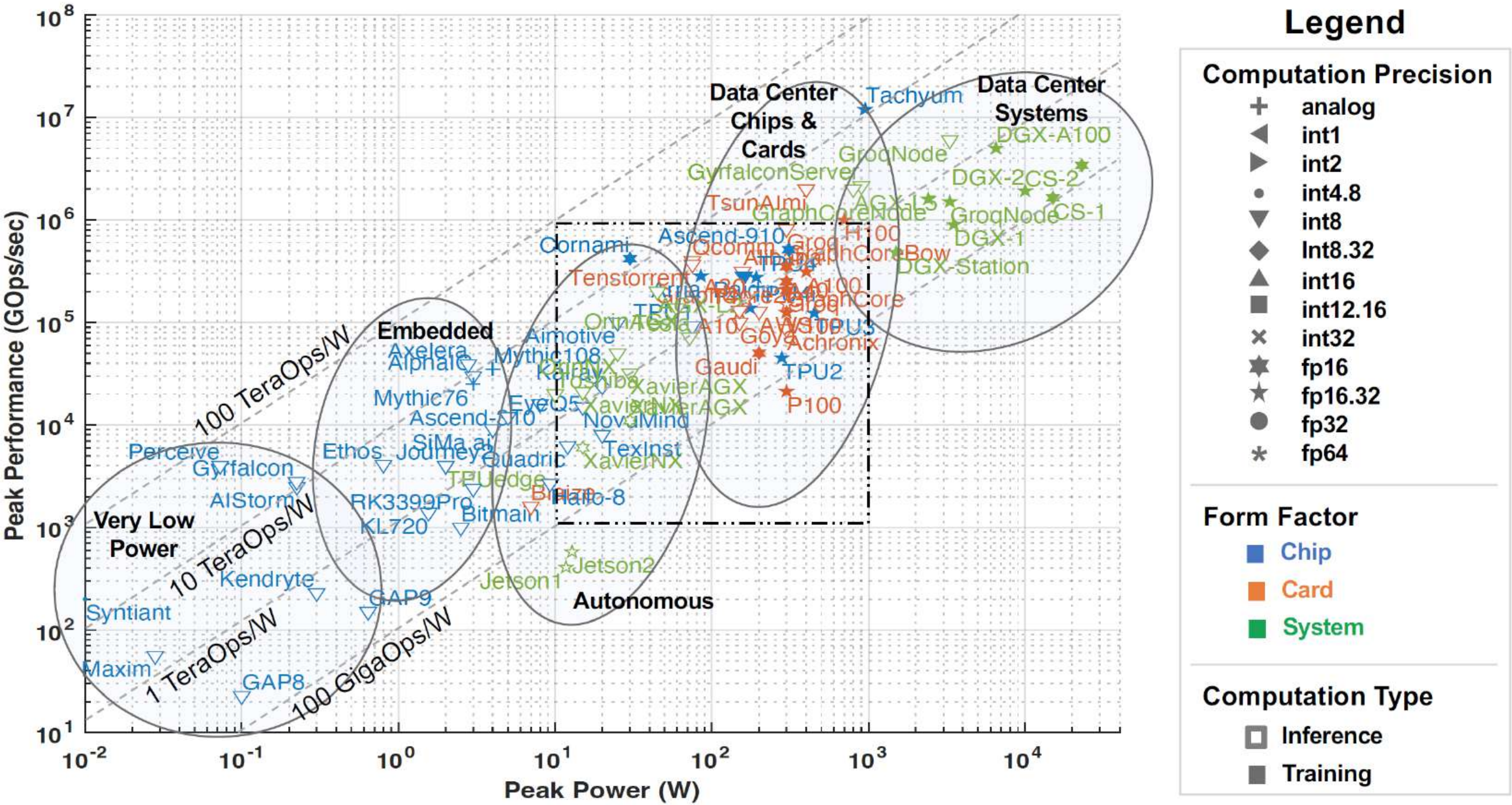
©nature



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

10x every 12 years…

Necessity is the Mother of Invention

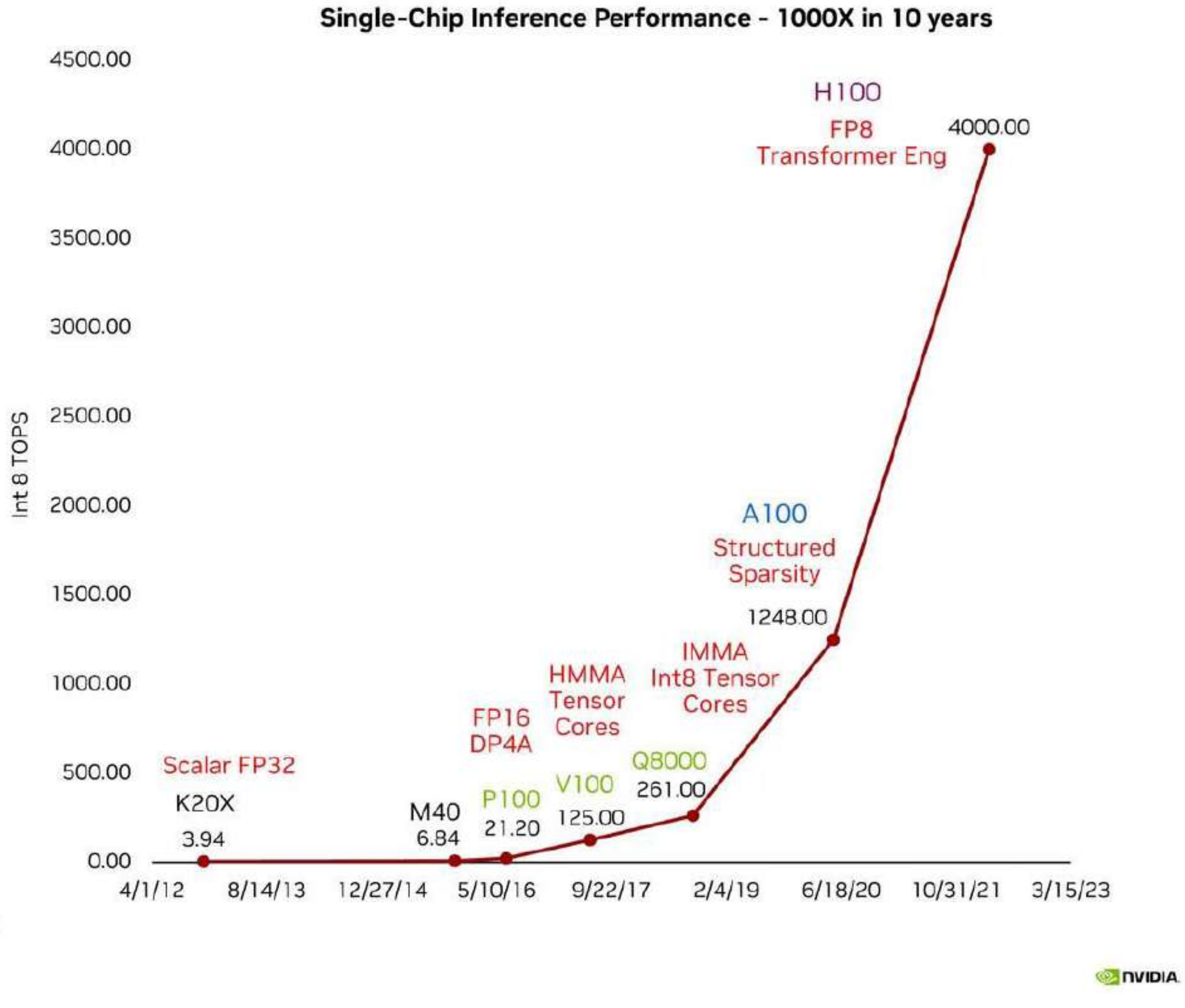


Reuther 22: arXiv:2210.04055

The Renaissance of Design

Gains from

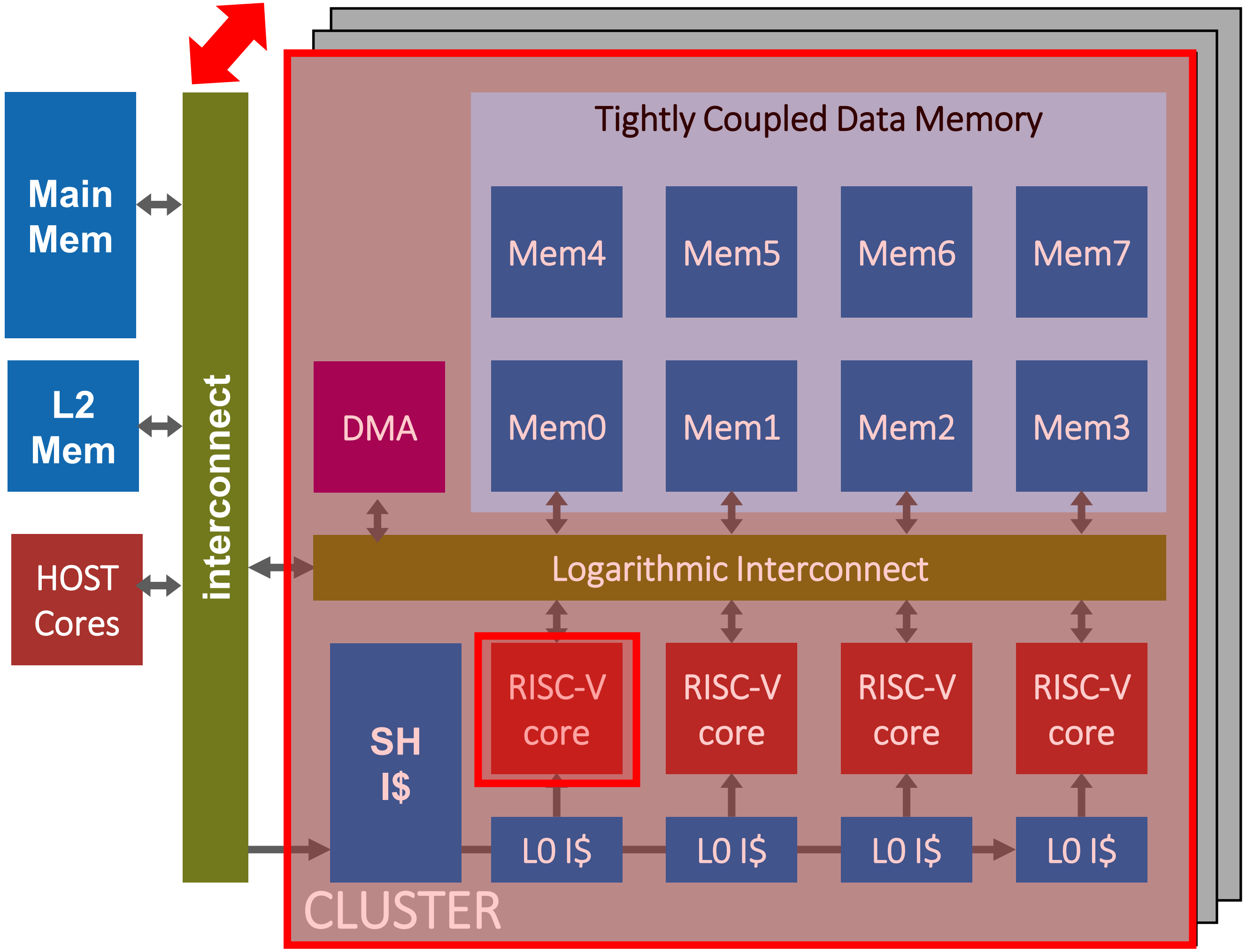
- Number Representation
  - FP32, FP16, Int8
  - (TF32, BF16)
  - ~16x
- Complex Instructions
  - DP4, HMMA, IMMA
  - ~12.5x
- Process
  - 28nm, 16nm, 7nm, 5nm
  - ~2.5x
- Sparsity
  - ~2x
- Model efficiency has also improved – overall gain > 1000x



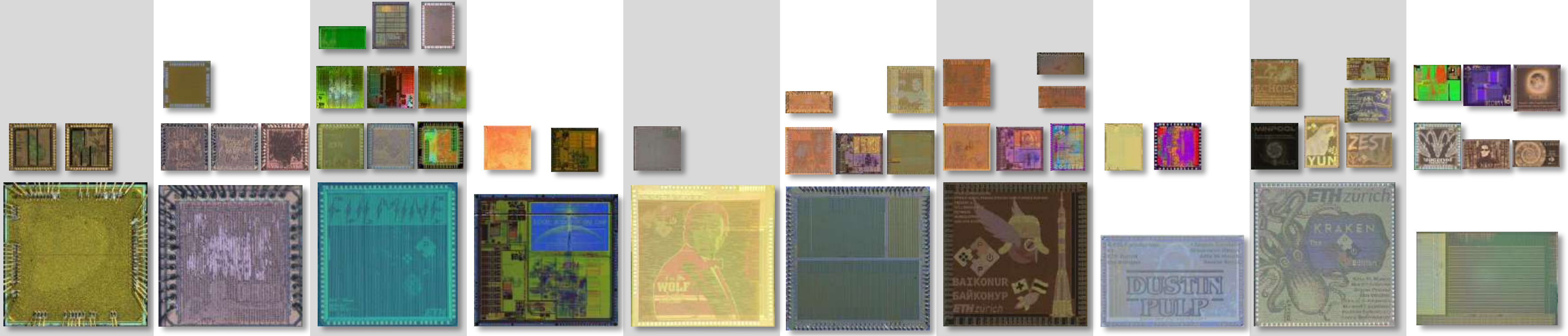
Dally HotChips 2023

Energy-Efficient Computing: Core to Platform

- Core
  - Improving core efficiency with ISA and uAch extensions
- Cluster
  - Efficient shared-mem cluster
  - From a few to thousand processing elements
- Full platform
  - Heterogeneity: host, processor, accelerator(s)
  - IOs, main memory
  - Chips→chiplets→stacks (2D to 3D)

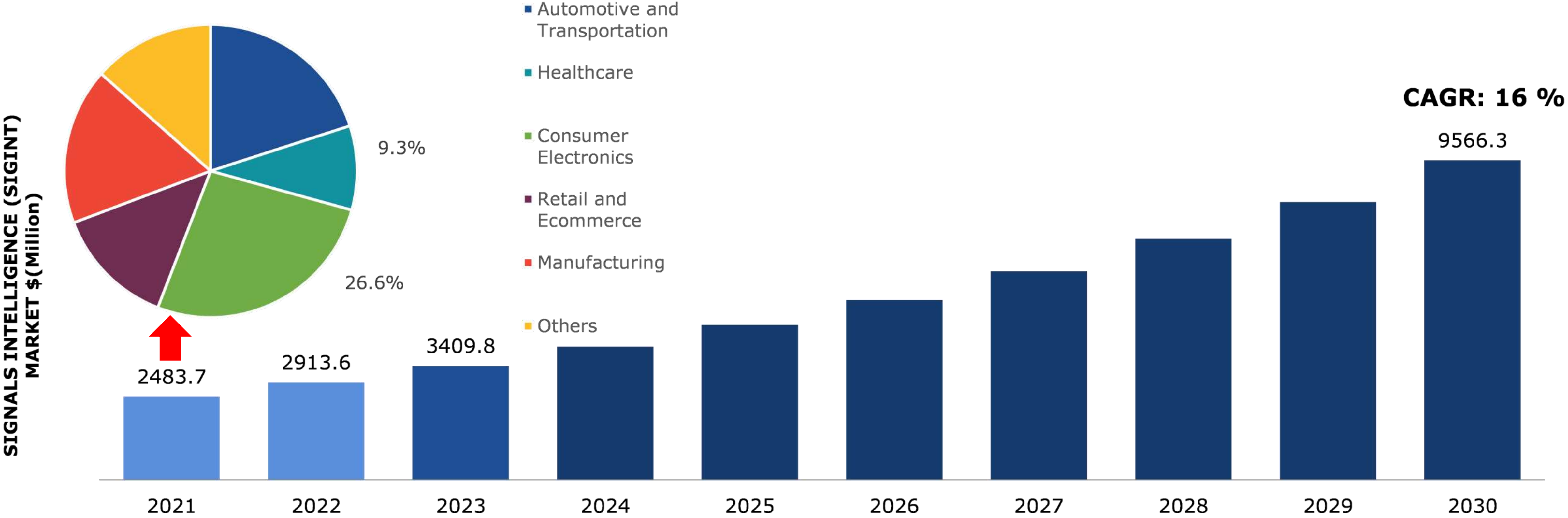


Proving ideas on Silicon



2013	(3)	**PULPv1** *STM 28FDSOI* Multi-core processor
2014	(5)	**Diana** *UMC 65* 4-core system with approximate FPUs
2015	(10)	**Fulmine** *UMC 65* 4-core system with ML and Crypto accelerators
2016	(3)	**VivoSoC 2.001** *SMIC 130* Mixed signal system for biosignal acquisiton
2017	(2)	**Mr. Wolf** *TSMC 40* 8+1 core IoT processor
2018	(6)	**Poseidon** *GF 22FDX* 64bit RISC-V core, 32bit Microcontroller system, ML accelerator
2019	(7)	**Baikonur** *GF 22FDX* Dual 64bit RISC-V core, 3x 8core snitch clusters, Body biasing test vehicle
2020	(3)	**Dustin** *TSMC 65* IoT processor with 16 cores and QNN enhancements
2021	(7)	**Kraken** *GF 22FDX* IoT processor with Spiking Neural and Ternary Inference Engines
2022	(7)	**Occamy** *GF 12LPP* ML accelerator with 216 + 1 cores and HBM interface

Edge ML Market



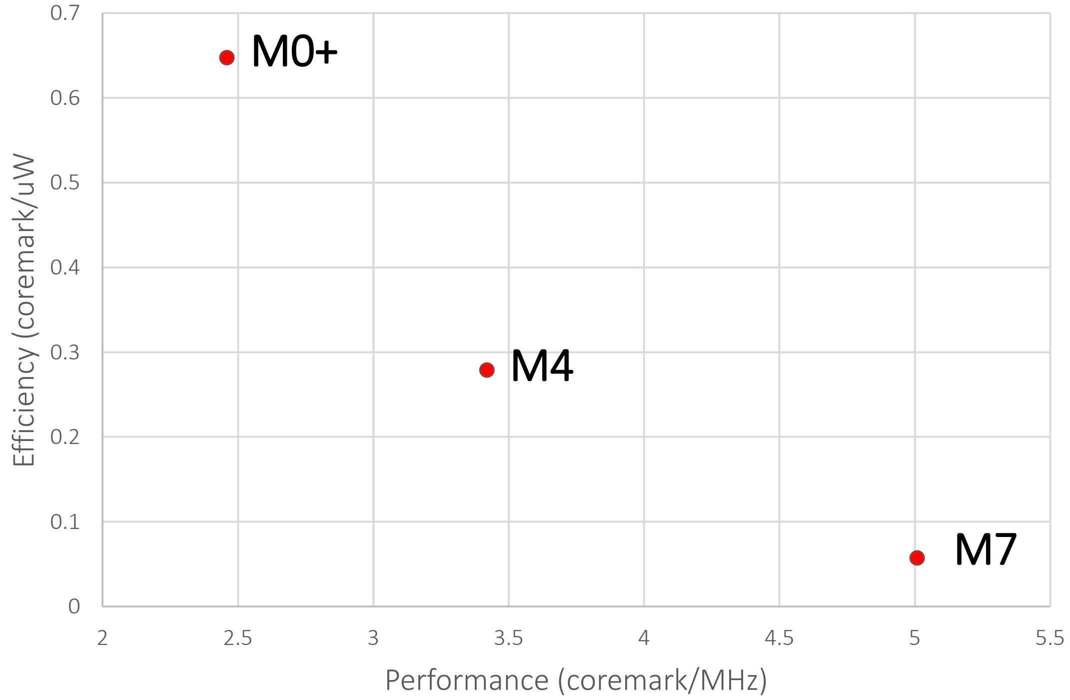
[Researchdive22]

TinyML challenge

AI capabilities in the power envelope of an MCU: 10-mW peak (1mW avg)

The Challenge: Energy efficiency@GOPS

ARM Cortex-M MCUs: M0+, M4, M7 (40LP, typ, 1.1V)*



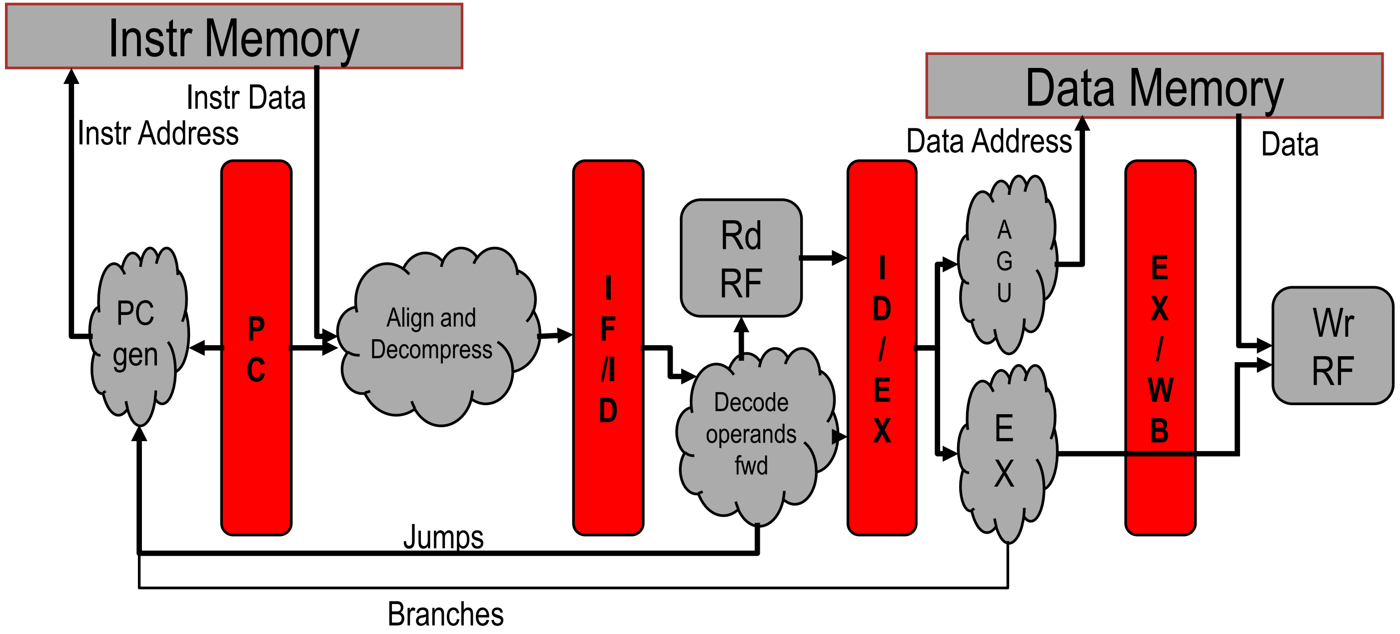
Energy-efficient MCUs

High performance MCUs

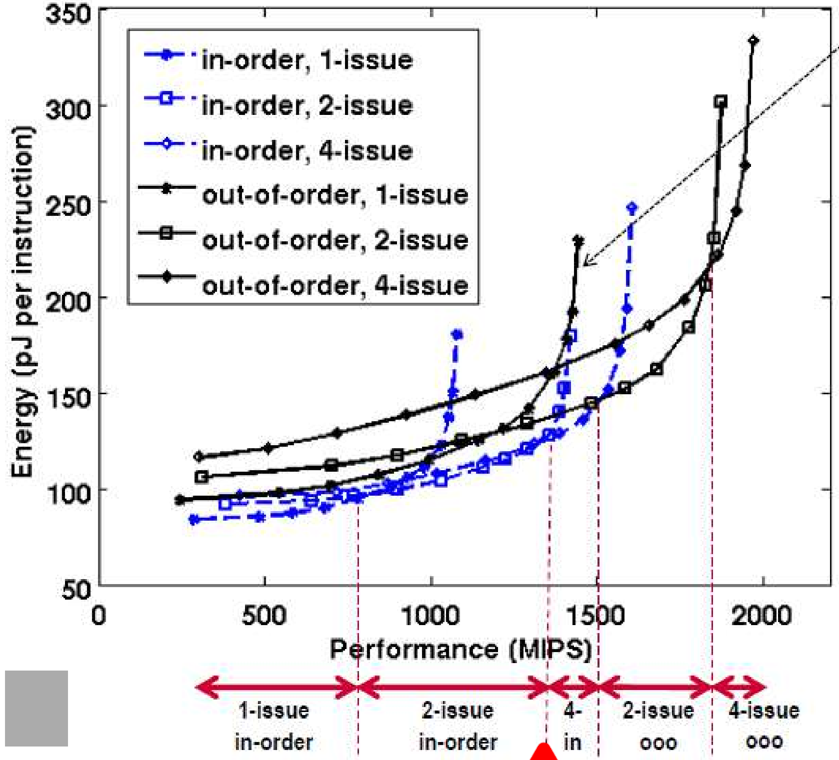
Why?

*data from ARM’s web

High-Performance vs. Energy-Efficient



[Azizi et al. ISCA10]

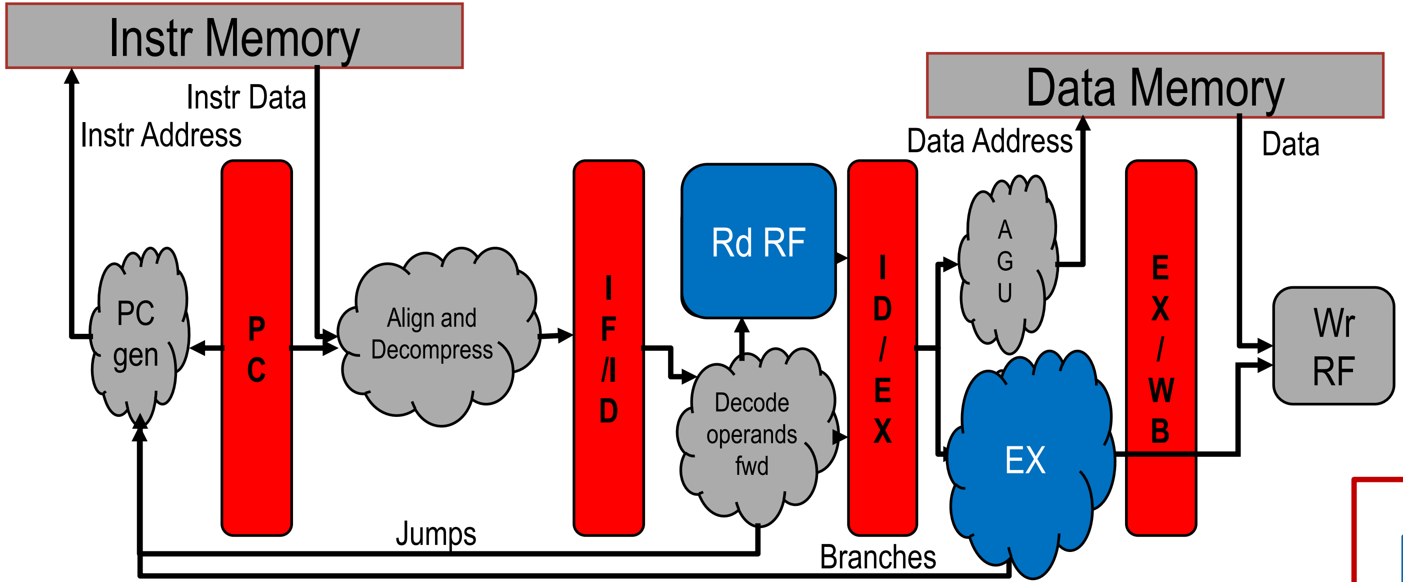


"Classical" core performance scaling trajectory

- Faster CLK → deeper pipeline → IPC drops
- Recover IPC → superscalar → ILP bottleneck (dependencies)
- Mitigate ILP bottlenecks → OOO → huge power, area cost!

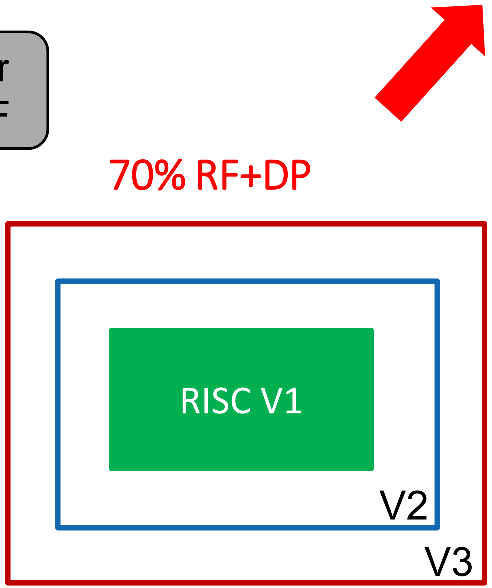
A way Out: Processor Specialization

3-cycle ALU-OP, 4-cyle MEM-OP→ only IPC loss: LD-use, Branch

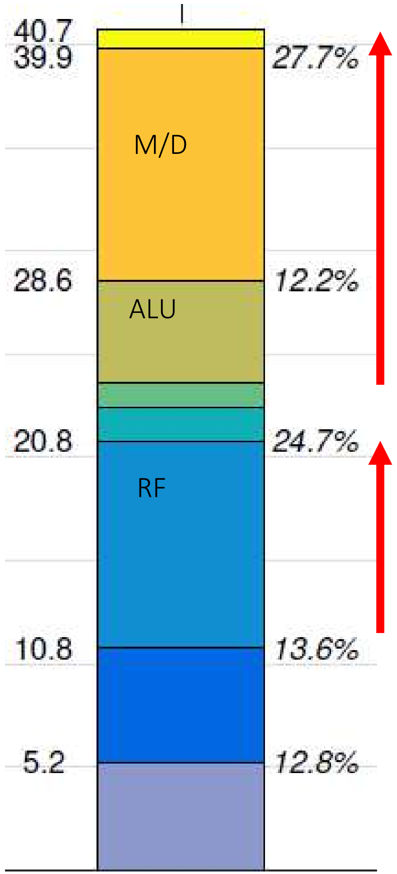


RISC-V ISA is extensible *by construction* (great!)

V1	Baseline RV (not good for ML)
V2	Extensions for Data Processing Data motion (e.g. auto-increment) Data processing (e.g. MAC)
V3	Domain specific data processing Narrow bitwidth HW support for special arithmetic



[Gautschi et al. TVLSI 2017]



ISA extension cost 25 kGE → 40 kGE (1.6x), energy efficient if 0.6Texec

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
```

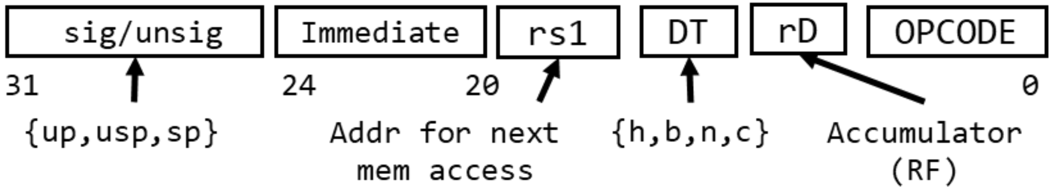
can we remove?

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`



sig/unsig | Immediate | rs1 | DT | rD | OPCODE

31 24 20 0

{up,usp,sp}

Addr for next mem access

{h,b,n,c}

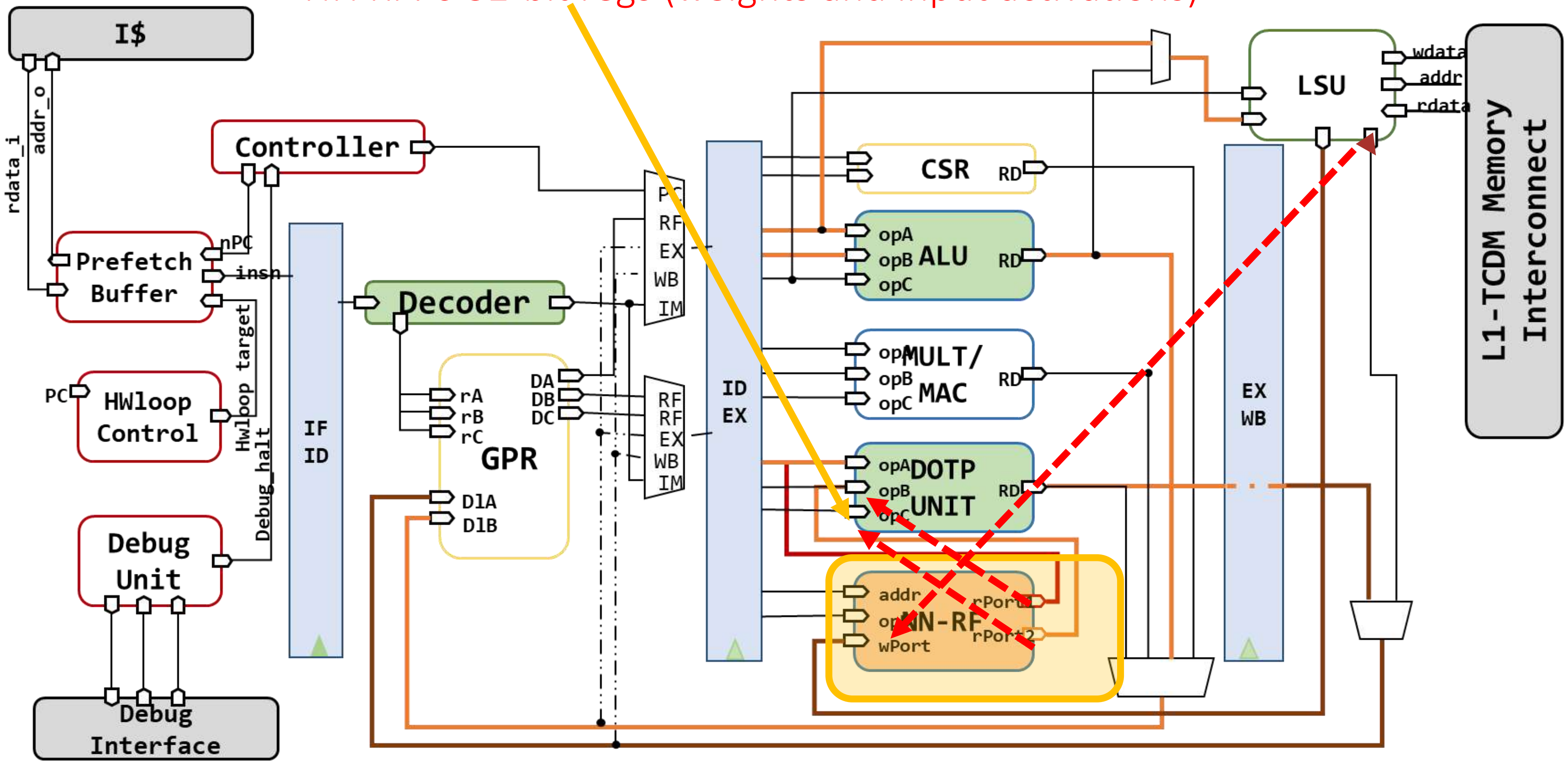
Accumulator (RF)

9x less instructions than RV32IMC

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

Hardware for `dotp+ld`

NN RF: 6 32-bit regs (weights and input activations)



🟫 Out of Core IPs 🟩 Extended RI5CY IPs 🟧 New core IPs

Special-purpose registers

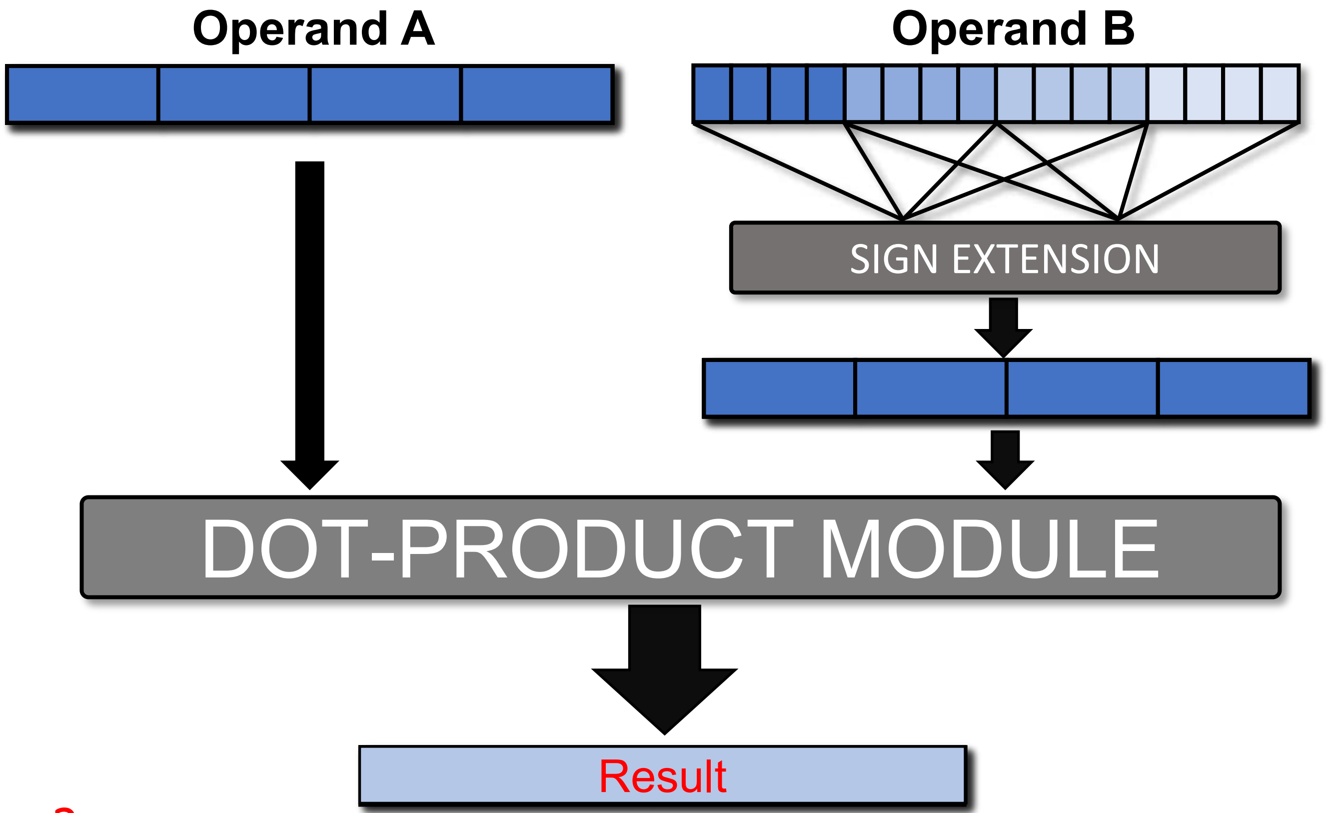
[A. Garofalo et al., TEC21]

Mixed Precision SIMD Processor

- Can support all variants:
  - 16x16, 16x8, 16x4, 16x2
  - 8x8, 8x4, 8x2
  - 4x4, 4x2
  - 2x2

- Avoids Pack/unpack Overheads
- Maximized performance (SIMD)
- Maximizes RF use (Data Locality)

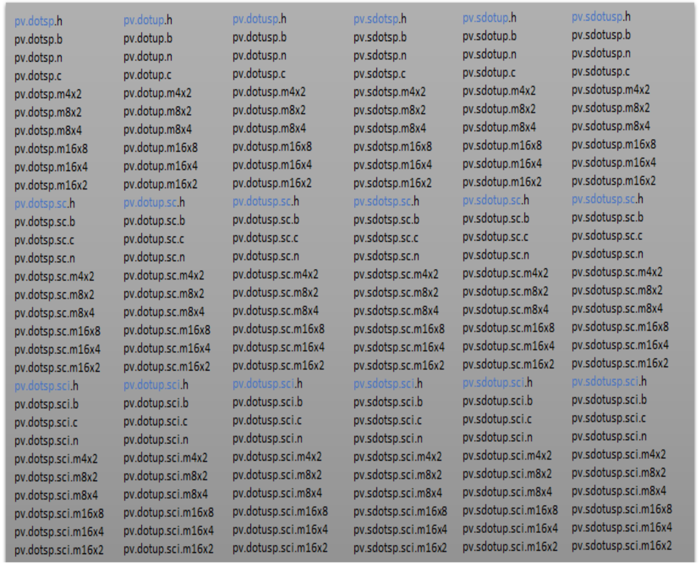
[Ottavi et al. ISVLSI20]



How to encode all these instructions?

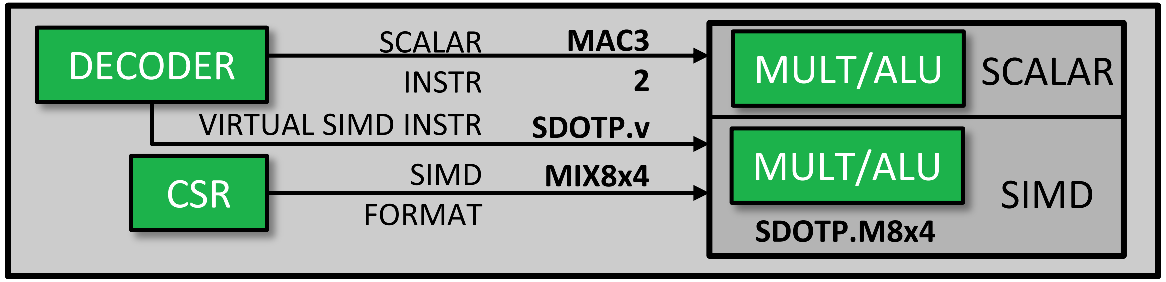
Virtual SIMD Instructions

- Encode operation as a virtual SIMD in the ISA (e.g. sdotsp.v)
- Format specified at runtime by a Control Register (e.g. 4x4)
- 180→18 Instructions needed for SIMD DOTP
- Potential to avoid code replication for different formats
- Tiny Overhead on QNN for Switching format
  - Format switch not frequent in DNN, e.g. every layer.



pv.dotsp.h	pv.dotup.h	pv.dotusp.h	pv.sdotsp.h	pv.sdotup.h	pv.sdotusp.h
pv.dotsp.b	pv.dotup.b	pv.dotusp.b	pv.sdotsp.b	pv.sdotup.b	pv.sdotusp.b
pv.dotsp.n	pv.dotup.n	pv.dotusp.n	pv.sdotsp.n	pv.sdotup.n	pv.sdotusp.n
pv.dotsp.c	pv.dotup.c	pv.dotusp.c	pv.sdotsp.c	pv.sdotup.c	pv.sdotusp.c
pv.dotsp.m4x2	pv.dotup.m4x2	pv.dotusp.m4x2	pv.sdotsp.m4x2	pv.sdotup.m4x2	pv.sdotusp.m4x2
pv.dotsp.m8x2	pv.dotup.m8x2	pv.dotusp.m8x2	pv.sdotsp.m8x2	pv.sdotup.m8x2	pv.sdotusp.m8x2
pv.dotsp.m8x4	pv.dotup.m8x4	pv.dotusp.m8x4	pv.sdotsp.m8x4	pv.sdotup.m8x4	pv.sdotusp.m8x4
pv.dotsp.m16x8	pv.dotup.m16x8	pv.dotusp.m16x8	pv.sdotsp.m16x8	pv.sdotup.m16x8	pv.sdotusp.m16x8
pv.dotsp.m16x4	pv.dotup.m16x4	pv.dotusp.m16x4	pv.sdotsp.m16x4	pv.sdotup.m16x4	pv.sdotusp.m16x4
pv.dotsp.m16x2	pv.dotup.m16x2	pv.dotusp.m16x2	pv.sdotsp.m16x2	pv.sdotup.m16x2	pv.sdotusp.m16x2
pv.dotsp.sc.h	pv.dotup.sc.h	pv.dotusp.sc.h	pv.sdotsp.sc.h	pv.sdotup.sc.h	pv.sdotusp.sc.h
pv.dotsp.sc.b	pv.dotup.sc.b	pv.dotusp.sc.b	pv.sdotsp.sc.b	pv.sdotup.sc.b	pv.sdotusp.sc.b
pv.dotsp.sc.c	pv.dotup.sc.c	pv.dotusp.sc.c	pv.sdotsp.sc.c	pv.sdotup.sc.c	pv.sdotusp.sc.c
pv.dotsp.sc.n	pv.dotup.sc.n	pv.dotusp.sc.n	pv.sdotsp.sc.n	pv.sdotup.sc.n	pv.sdotusp.sc.n
pv.dotsp.sc.m4x2	pv.dotup.sc.m4x2	pv.dotusp.sc.m4x2	pv.sdotsp.sc.m4x2	pv.sdotup.sc.m4x2	pv.sdotusp.sc.m4x2
pv.dotsp.sc.m8x2	pv.dotup.sc.m8x2	pv.dotusp.sc.m8x2	pv.sdotsp.sc.m8x2	pv.sdotup.sc.m8x2	pv.sdotusp.sc.m8x2
pv.dotsp.sc.m8x4	pv.dotup.sc.m8x4	pv.dotusp.sc.m8x4	pv.sdotsp.sc.m8x4	pv.sdotup.sc.m8x4	pv.sdotusp.sc.m8x4
pv.dotsp.sc.m16x8	pv.dotup.sc.m16x8	pv.dotusp.sc.m16x8	pv.sdotsp.sc.m16x8	pv.sdotup.sc.m16x8	pv.sdotusp.sc.m16x8
pv.dotsp.sc.m16x4	pv.dotup.sc.m16x4	pv.dotusp.sc.m16x4	pv.sdotsp.sc.m16x4	pv.sdotup.sc.m16x4	pv.sdotusp.sc.m16x4
pv.dotsp.sc.m16x2	pv.dotup.sc.m16x2	pv.dotusp.sc.m16x2	pv.sdotsp.sc.m16x2	pv.sdotup.sc.m16x2	pv.sdotusp.sc.m16x2
pv.dotsp.sci.h	pv.dotup.sci.h	pv.dotusp.sci.h	pv.sdotsp.sci.h	pv.sdotup.sci.h	pv.sdotusp.sci.h
pv.dotsp.sci.b	pv.dotup.sci.b	pv.dotusp.sci.b	pv.sdotsp.sci.b	pv.sdotup.sci.b	pv.sdotusp.sci.b
pv.dotsp.sci.c	pv.dotup.sci.c	pv.dotusp.sci.c	pv.sdotsp.sci.c	pv.sdotup.sci.c	pv.sdotusp.sci.c
pv.dotsp.sci.n	pv.dotup.sci.n	pv.dotusp.sci.n	pv.sdotsp.sci.n	pv.sdotup.sci.n	pv.sdotusp.sci.n
pv.dotsp.sci.m4x2	pv.dotup.sci.m4x2	pv.dotusp.sci.m4x2	pv.sdotsp.sci.m4x2	pv.sdotup.sci.m4x2	pv.sdotusp.sci.m4x2
pv.dotsp.sci.m8x2	pv.dotup.sci.m8x2	pv.dotusp.sci.m8x2	pv.sdotsp.sci.m8x2	pv.sdotup.sci.m8x2	pv.sdotusp.sci.m8x2
pv.dotsp.sci.m8x4	pv.dotup.sci.m8x4	pv.dotusp.sci.m8x4	pv.sdotsp.sci.m8x4	pv.sdotup.sci.m8x4	pv.sdotusp.sci.m8x4
pv.dotsp.sci.m16x8	pv.dotup.sci.m16x8	pv.dotusp.sci.m16x8	pv.sdotsp.sci.m16x8	pv.sdotup.sci.m16x8	pv.sdotusp.sci.m16x8
pv.dotsp.sci.m16x4	pv.dotup.sci.m16x4	pv.dotusp.sci.m16x4	pv.sdotsp.sci.m16x4	pv.sdotup.sci.m16x4	pv.sdotusp.sci.m16x4
pv.dotsp.sci.m16x2	pv.dotup.sci.m16x2	pv.dotusp.sci.m16x2	pv.sdotsp.sci.m16x2	pv.sdotup.sci.m16x2	pv.sdotusp.sci.m16x2

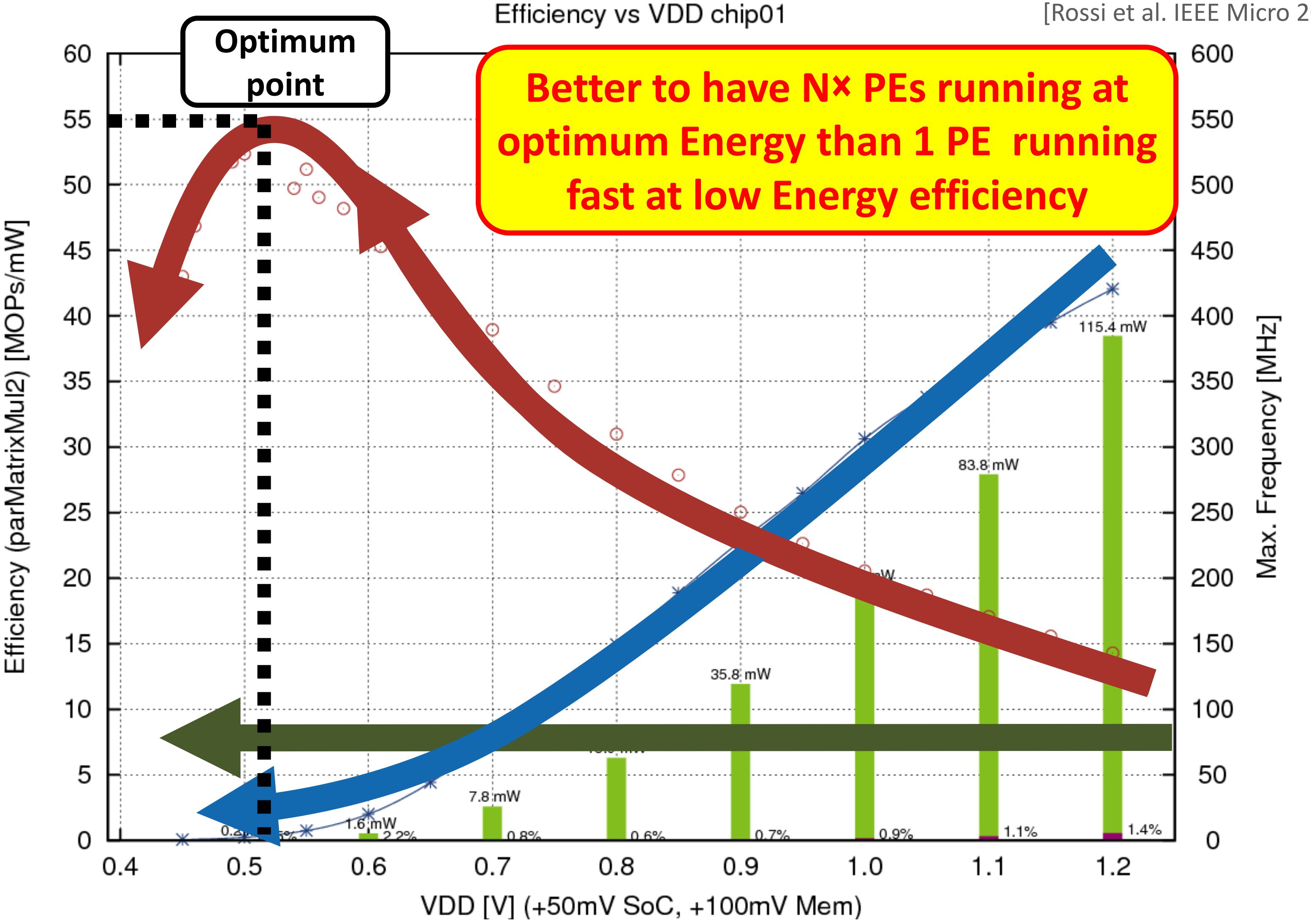
pv.dotsp.v	pv.sdotsp .v
pv.dotsp.sc .v	pv.sdotsp.sc .v
pv.dotsp.sci .v	pv.sdotsp.sci .v
pv.dotup .v	pv.sdotup .v
pv.dotup.sc .v	pv.sdotup.sc .v
pv.dotup.sci .v	pv.sdotup.sci .v
pv.dotusp .v	pv.sdotusp .v
pv.dotusp.sc .v	pv.sdotusp.sc .v
pv.dotusp.sci .v	pv.sdotusp.sci .v



Scaling performance: Parallel, Ultra-Low Power (PULP)

- As VDD decreases, operating speed decreases
- However efficiency increases→ more work done per Joule
- Until leakage effects start to dominate
- Put more units in parallel to get performance up and keep them busy with a parallel workload

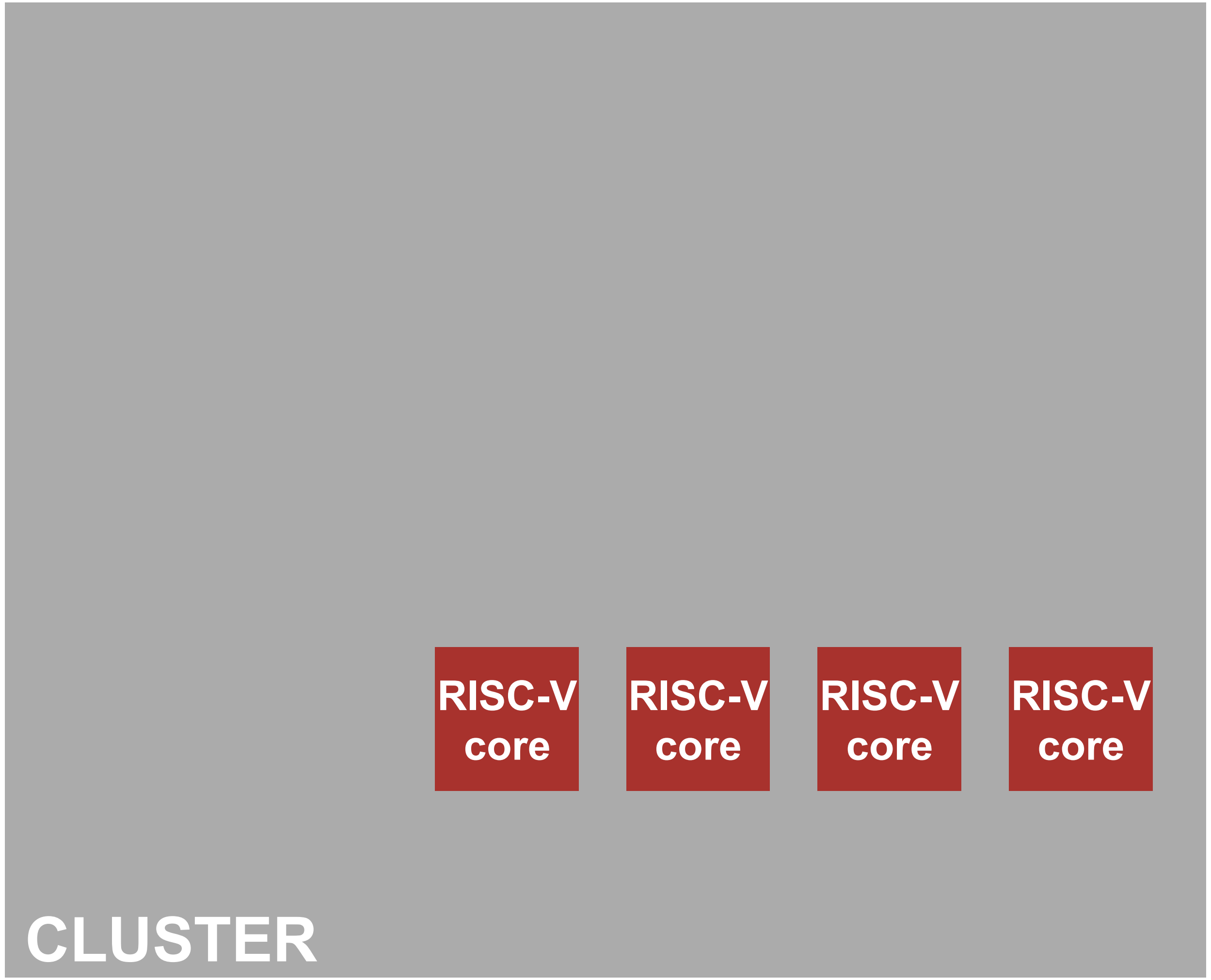
ML is massively parallel and scales well
(P/S ↑ with NN size)



[Rossi et al. IEEE Micro 2017]

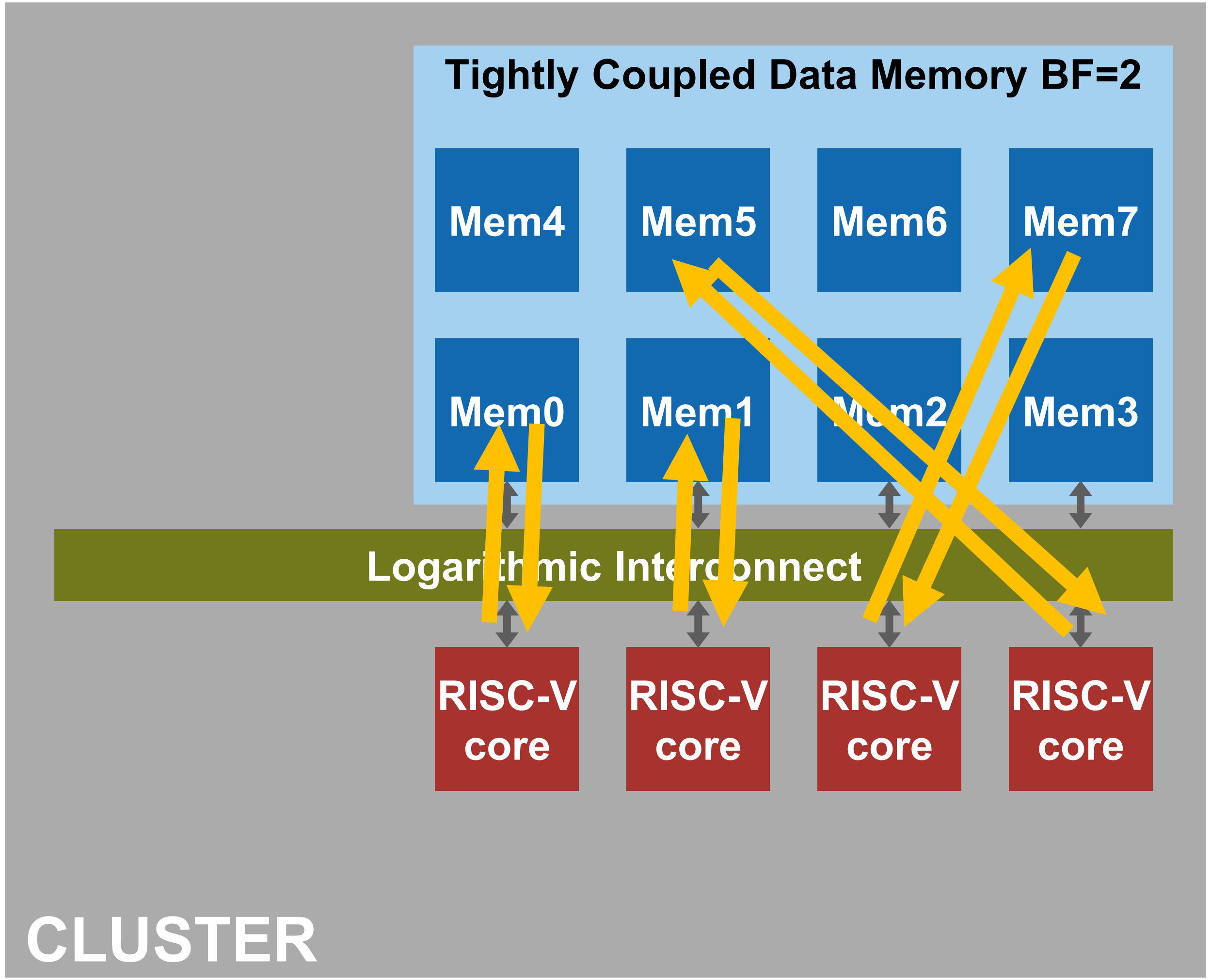
Better to have N× PEs running at optimum Energy than 1 PE running fast at low Energy efficiency

Multiple RI5CY Cores (1-16)

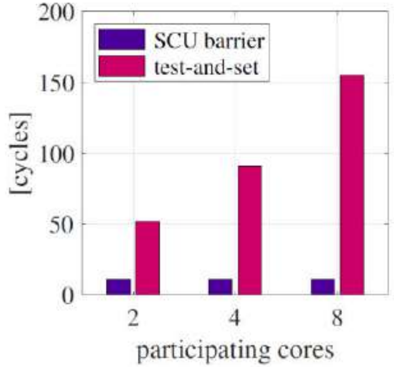


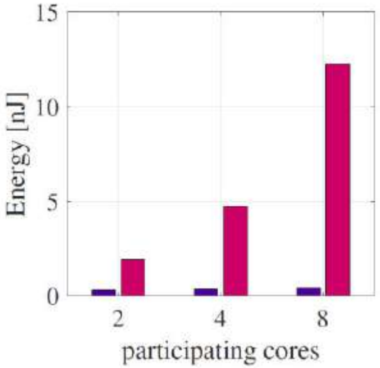
Low-Latency Shared TCDM

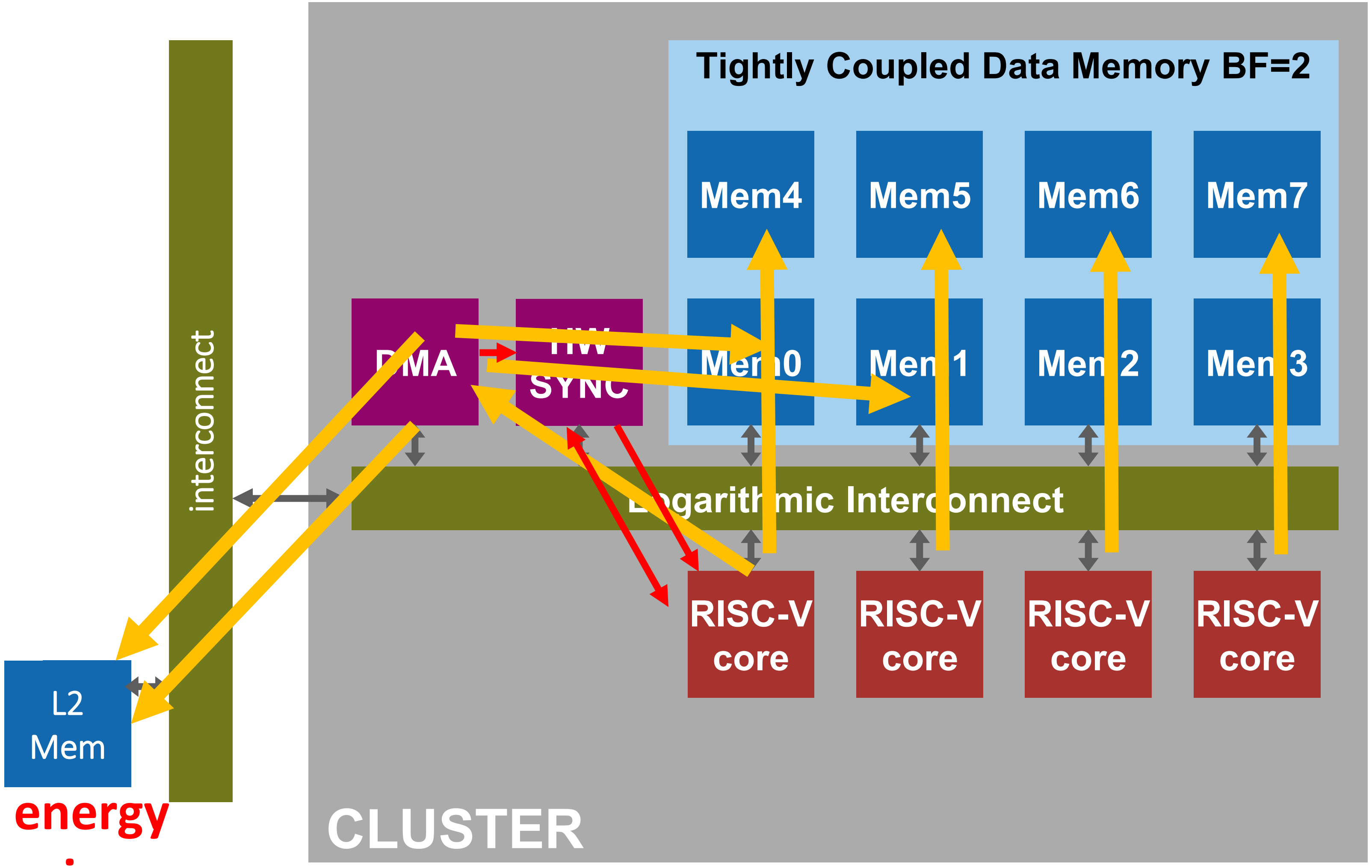
- Parallel memory access with low contention
  - Multi-banked, address-interleaved L1
- Fast interconnect with physical design awareness
  - Logarithmic depth of combinational switchboxes



Fast synchronization, non-blocking DMA L1-L2 copies





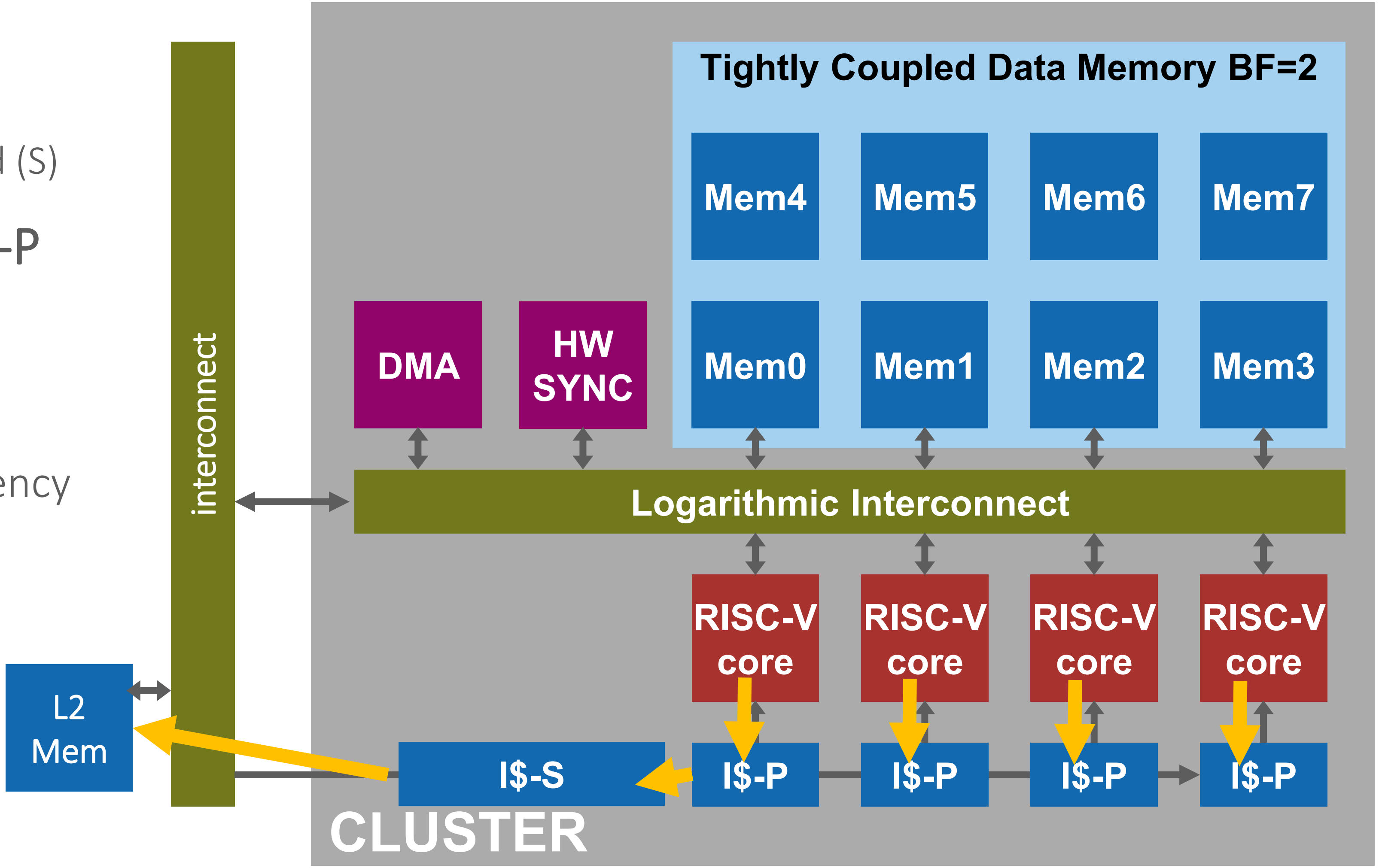


~15x latency and energy reduction for a barrier

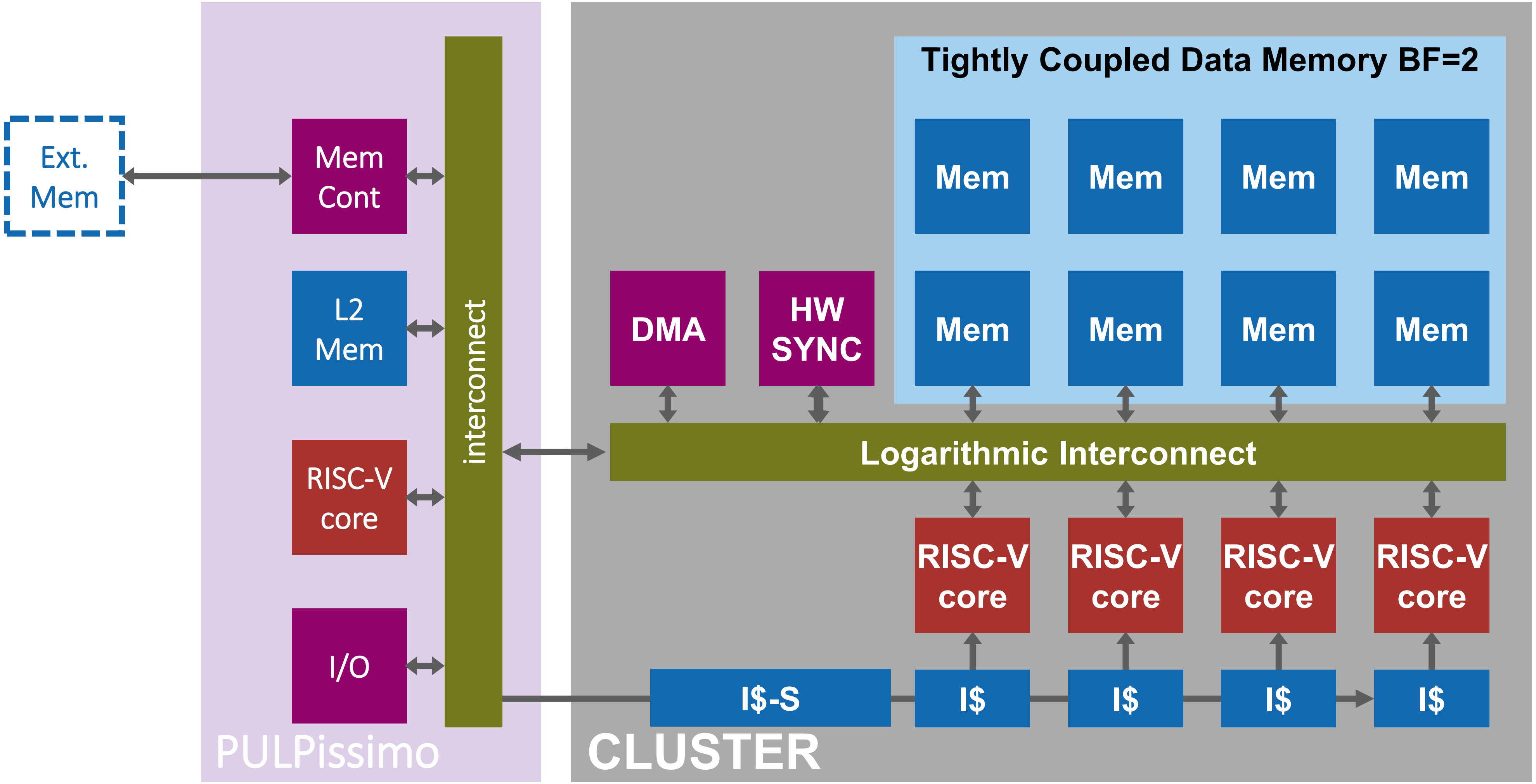
[Glaser TPDS20]

Shared instruction cache with private “loop buffer”

- Two-level I$
    - Private (P) + Shared (S)
- Most IFs from I$-P
    - Low IF energy
- I$-S for capacity
    - Reduces miss latency



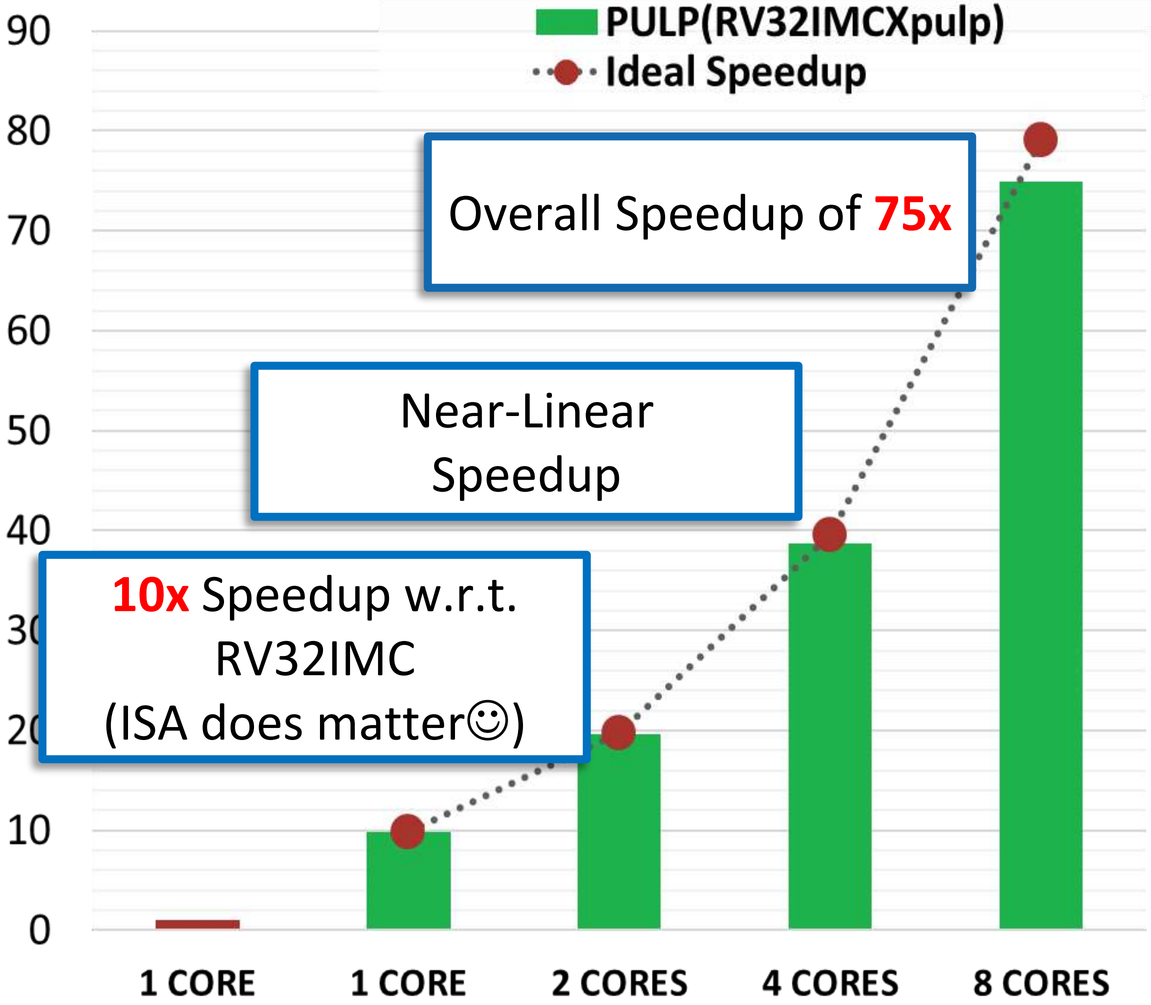
Host for sequential, I/O + Data-Parallel Cluster



Open sourced since 2017: github.com/pulp-platform/pulp

Combining ISA extension + Efficient parallel execution

- 8-bit convolution
    - Open source DNN library
- 10x through xPULP
    - Extensions bring real speedup
- Near-linear speedup
    - Scales well for regular workloads
- 75x overall gain
- 7-8 GMACs
    - 250MHz
    - 4 MAC/Cycle (8bit)
    - 8 Cores



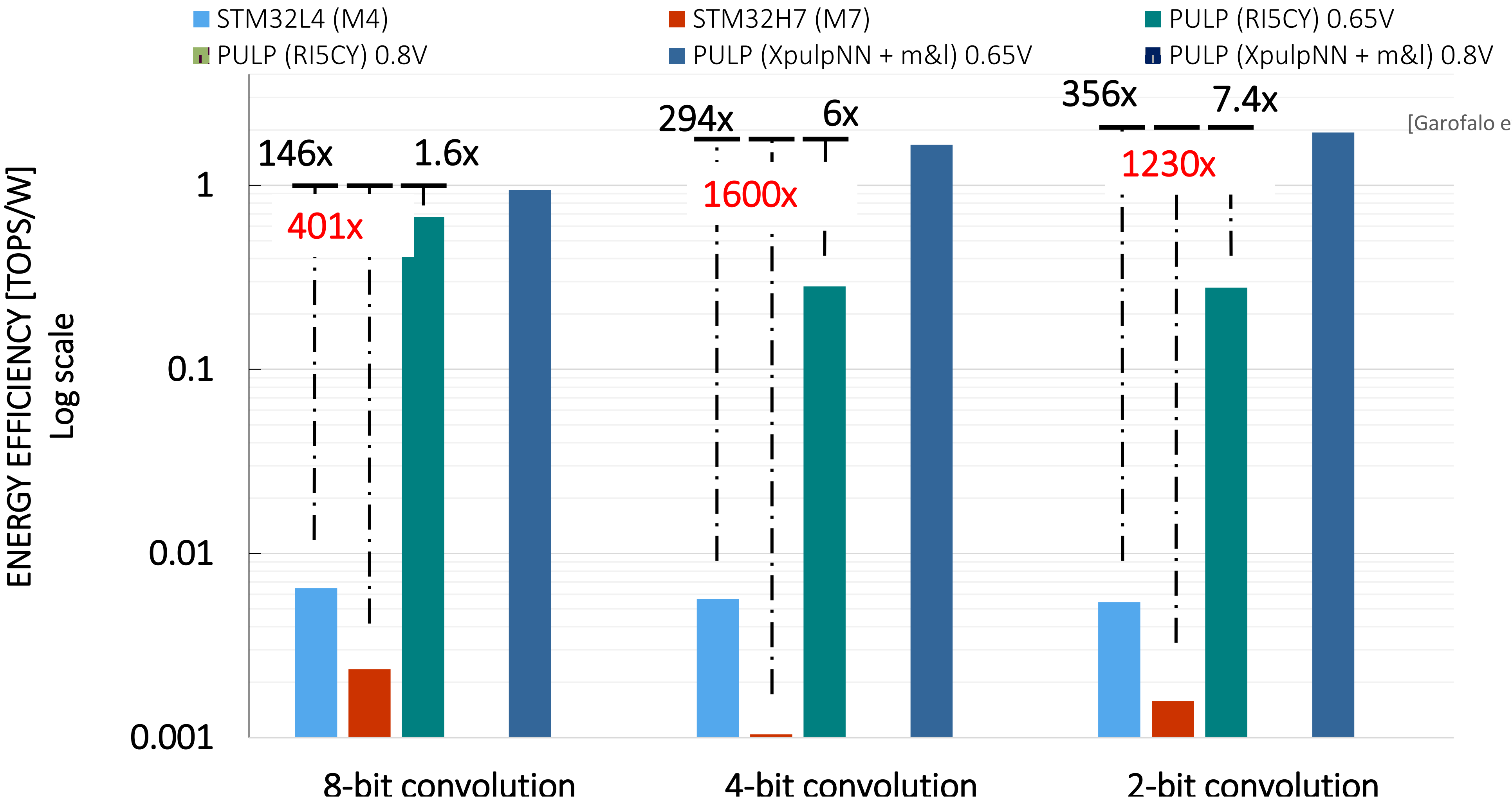
Overall Speedup of **75x**

Near-Linear Speedup

10x Speedup w.r.t. RV32IMC (ISA does matter☺)

[Garofalo et al. Philos. Trans. R. Soc 20]

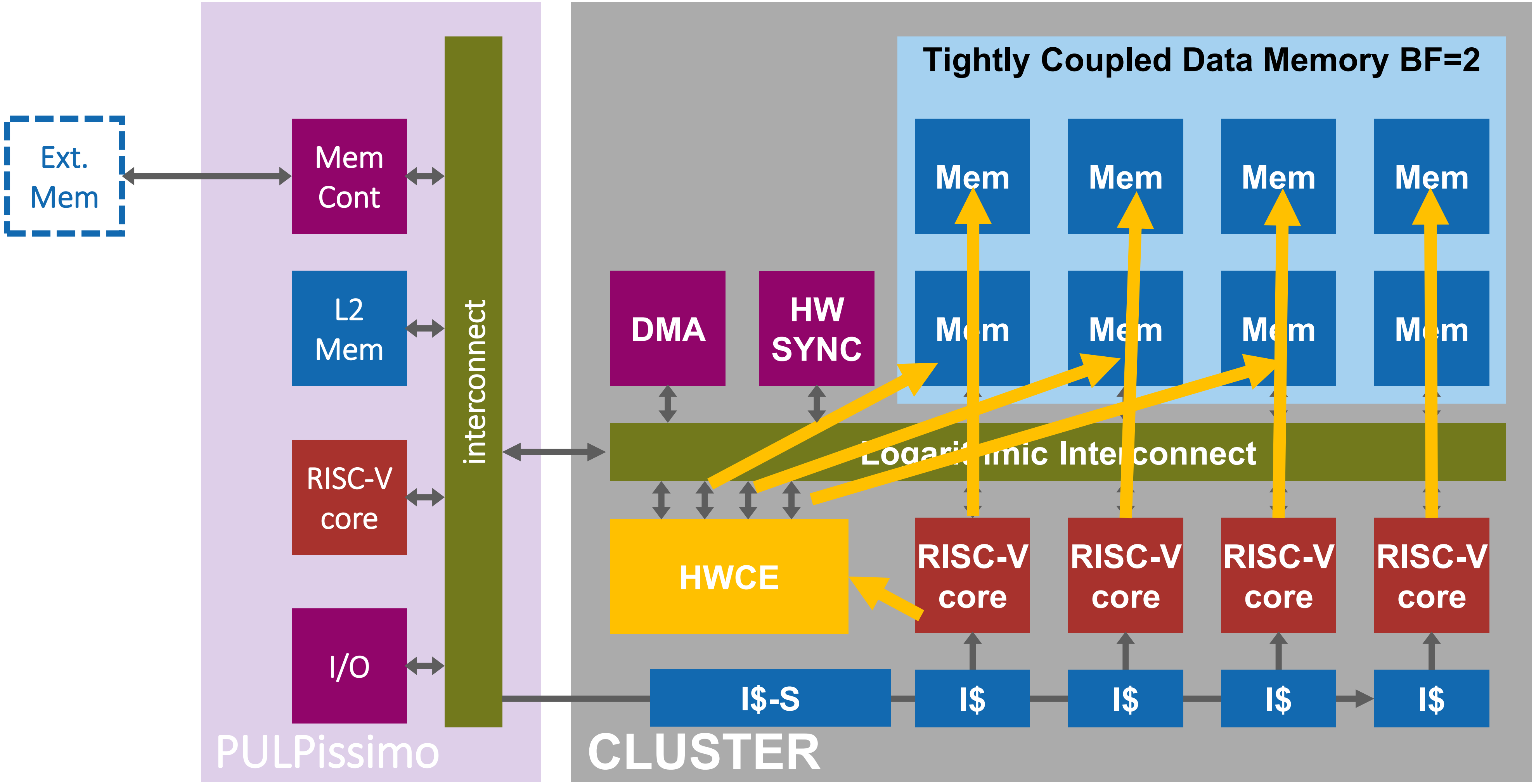
8-Cores Cluster + ISA (FDX22nm)



[Garofalo et al. OJSSCS22]

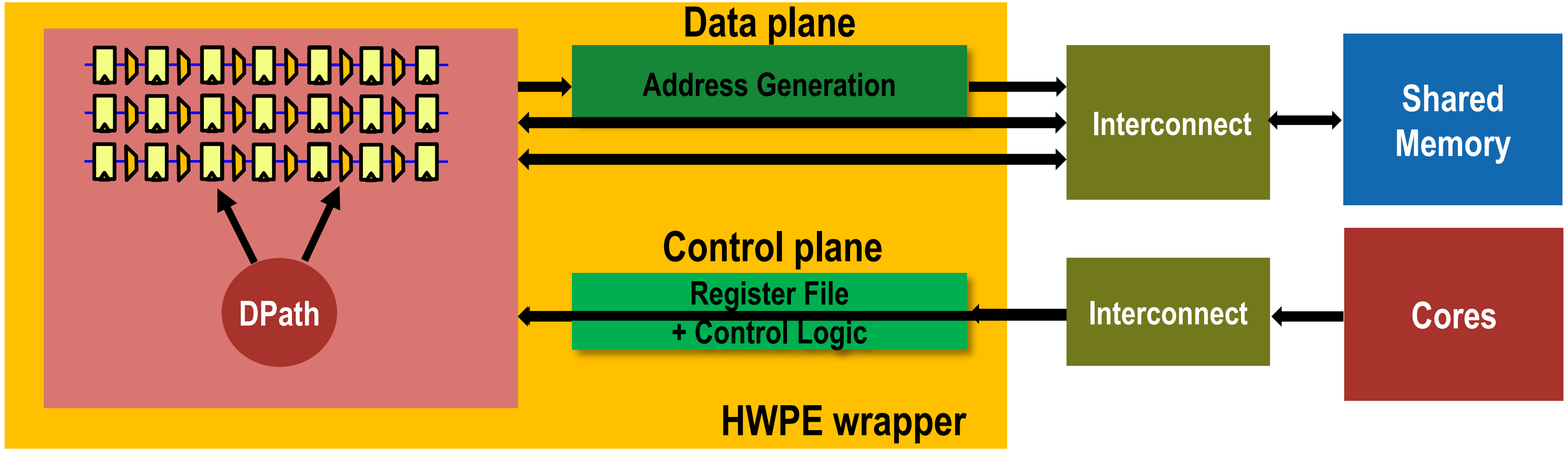
More GOPS, Less Power?

What's next? Tightly-coupled HW Compute Engine



Acceleration with flexibility: zero-copy HW-SW cooperation

Hardware Processing Engines (HWPEs)

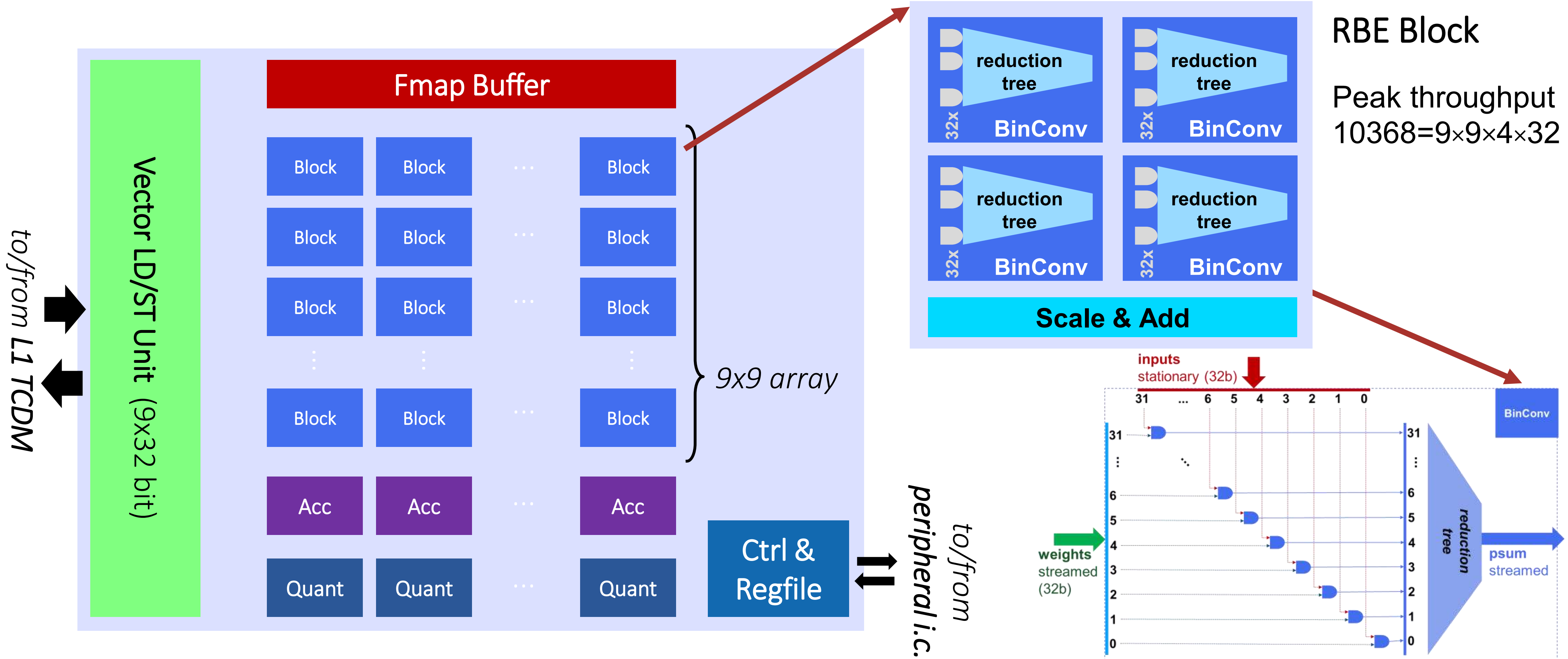


HWPE efficiency ($\frac{MAC}{A(mm^2),E(J),W(bps)}$) vs. optimized RISC-V core

1. Dedicated control (no I-fetch) with shadow registers (overlapped config-exec)
2. Specialized high-BW interco into L1 (on data-plane)
3. Specialized datapath: supporting configurable & aggressive quantization ⬅

Reconfigurable Binary Engine

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



RBE Block

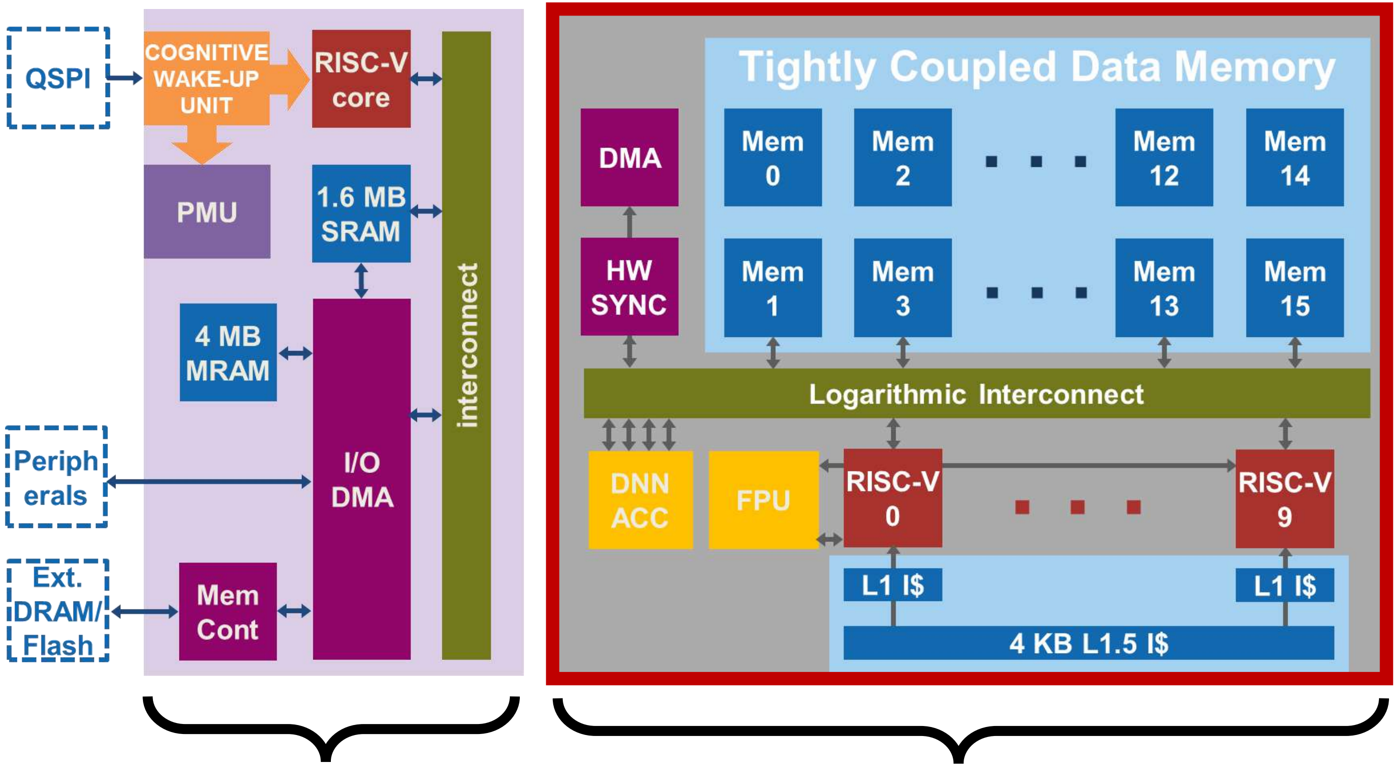
Peak throughput
10368=9×9×4×32

Energy efficiency 10-20x (0.1pJ/OP) wrt to SW on cluster @same accuracy

All together in VEGA: Extreme Edge IoT Processor

[Rossi et al. ISSCC21]

- RISC-V cluster (8cores +1) 614GOPS/W @ 7.6GOPS (8bit DNNs), 79GFLOPS/W @ 1GFLOP (32bit FP appl)
- Multi-precision HWCE(4b/8b/16b) 3×3×3 MACs with normalization / activation: 32.2GOPS and 1.3TOPS/W (8bit)
- 1.7 µW cognitive unit for autonomous wake-up from retentive sleep mode



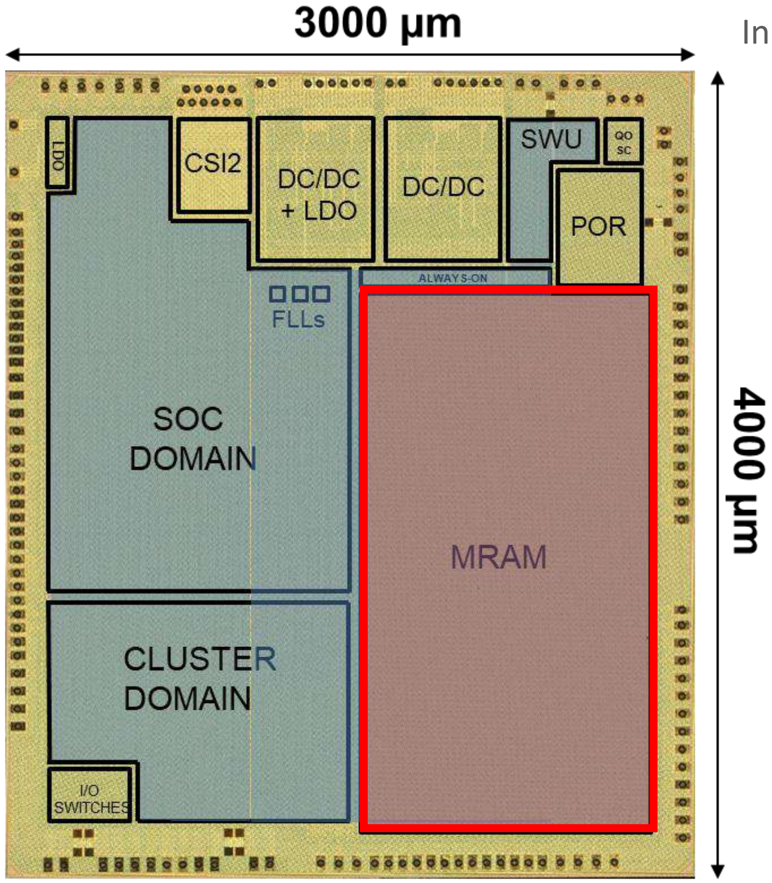
SoC: SoA MCU

Parallel SW prog. accelerator

In cooperation with GREENWAVES TECHNOLOGIES

All together in VEGA: Extreme Edge IoT Processor

- RISC-V cluster (8cores +1) 614GOPS/W @ 7.6GOPS (8bit DNNs), 79GFLOPS/W @ 1GFLOP (32bit FP appl)
- Multi-precision HWCE(4b/8b/16b) 3×3×3 MACs with normalization / activation: 32.2GOPS and 1.3TOPS/W (8bit)
- 1.7 µW cognitive unit for autonomous wake-up from retentive sleep mode
- Fully-on chip DNN inference with 4MB MRAM (high-density NVM with good scaling)

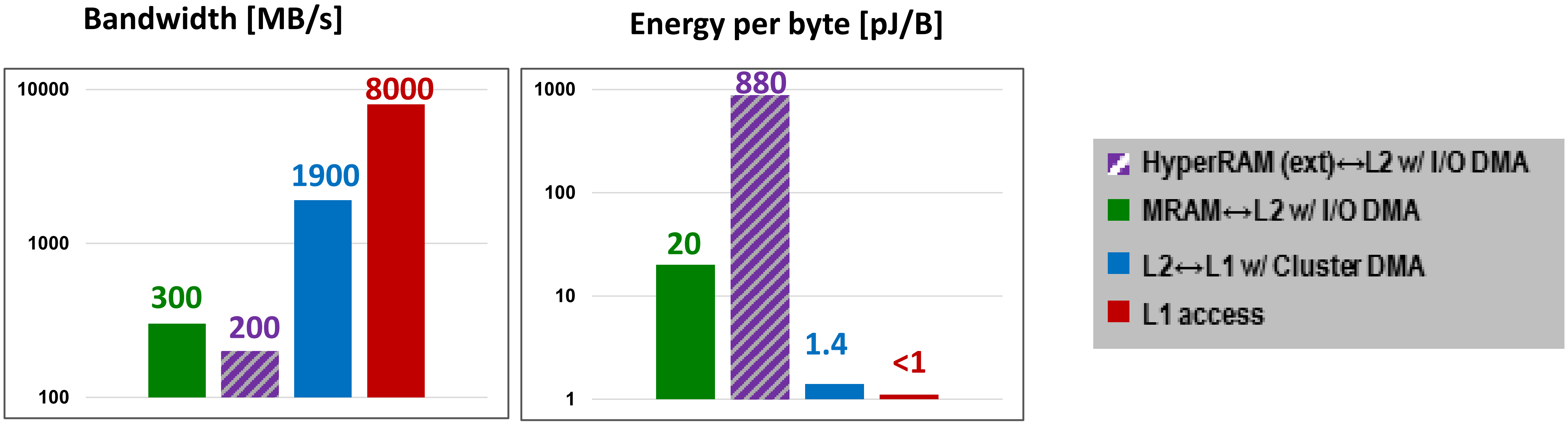


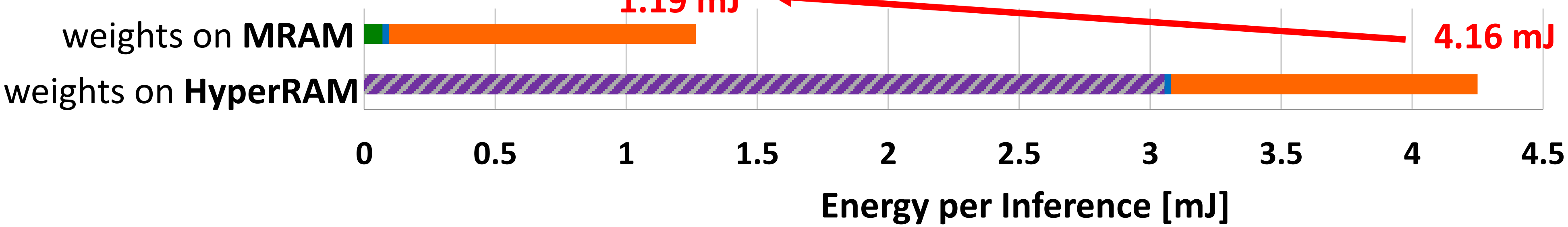
[D. Rossi, ISSCC21]

In cooperation with GREENWAVES TECHNOLOGIES

Technology	22nm FDSOI
Chip Area	12mm^2
SRAM	1.7 MB
MRAM	4 MB
VDD range	0.5V - 0.8V
VBB range	0V - 1.1V
Fr. Range	32 kHz - 450 MHz
Pow. Range	1.7 µW - 49.4 mW

Full DNN Energy (MobileNetV2) on Vega





end-to-end on-chip computation

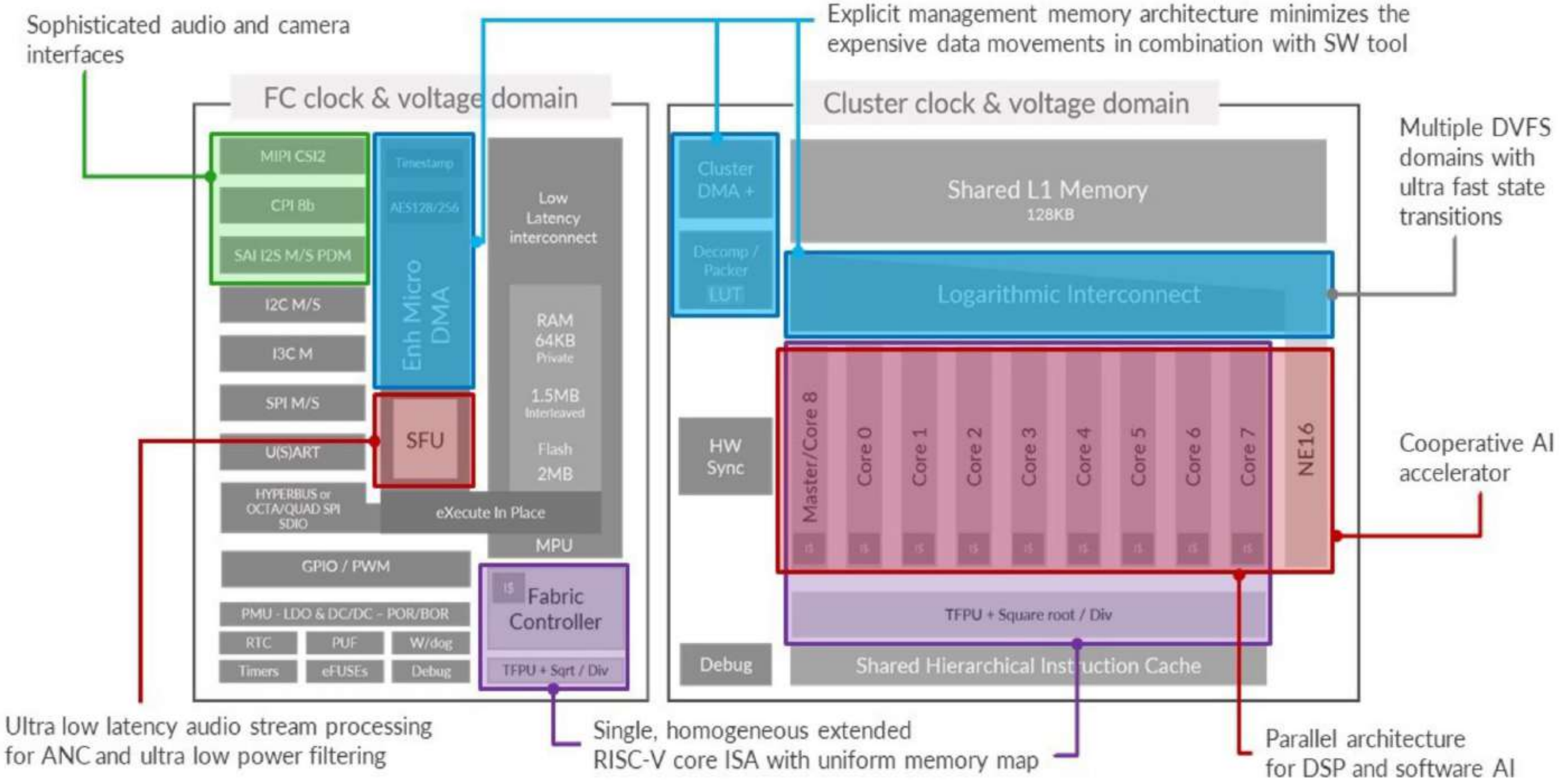
1.19 mJ

3.5x less energy

4.16 mJ

PULP→GAP8, VEGA→GAP9

GREENWAVES TECHNOLOGIES



Sophisticated audio and camera interfaces

Explicit management memory architecture minimizes the expensive data movements in combination with SW tool

Multiple DVFS domains with ultra fast state transitions

Cooperative AI accelerator

Ultra low latency audio stream processing for ANC and ultra low power filtering

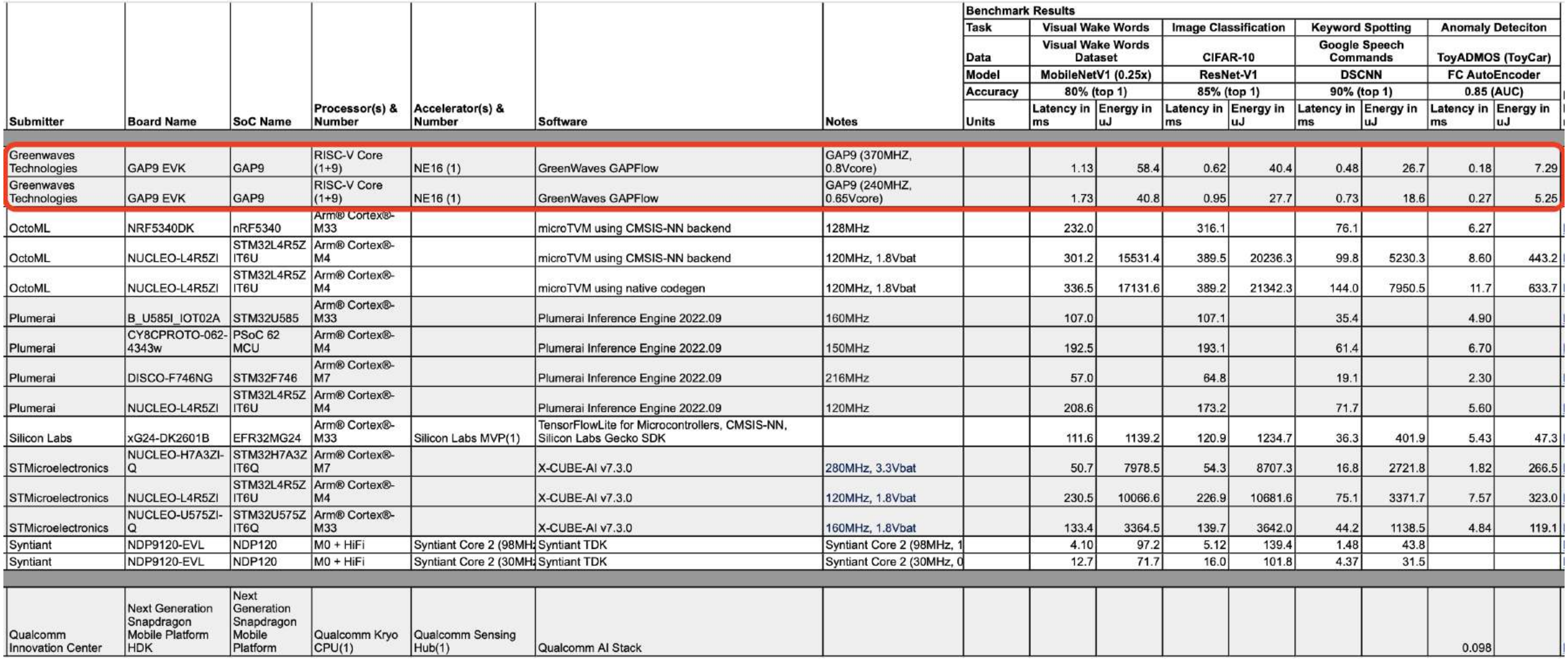
Single, homogeneous extended RISC-V core ISA with uniform memory map

Parallel architecture for DSP and software AI

Respectively 85% and 65% of GAP8 and GAP9 are based on open-source IPs

PULP→GAP8, VEGA→GAP9

November 09, 2022 – Inference: Tiny v1.0 mlcommons.org



							Benchmark Results								
							Task	Visual Wake Words		Image Classification		Keyword Spotting		Anomaly Deteciton	
							Data	Visual Wake Words Dataset		CIFAR-10		Google Speech Commands		ToyADMOS (ToyCar)	
							Model	MobileNetV1 (0.25x)		ResNet-V1		DSCNN		FC AutoEncoder	
							Accuracy	80% (top 1)		85% (top 1)		90% (top 1)		0.85 (AUC)	
Submitter	Board Name	SoC Name	Processor(s) & Number	Accelerator(s) & Number	Software	Notes	Units	Latency in ms	Energy in uJ	Latency in ms	Energy in uJ	Latency in ms	Energy in uJ	Latency in ms	Energy in uJ
Greenwaves Technologies	GAP9 EVK	GAP9	RISC-V Core (1+9)	NE16 (1)	GreenWaves GAPFlow	GAP9 (370MHZ, 0.8Vcore)		1.13	58.4	0.62	40.4	0.48	26.7	0.18	7.29
Greenwaves Technologies	GAP9 EVK	GAP9	RISC-V Core (1+9)	NE16 (1)	GreenWaves GAPFlow	GAP9 (240MHZ, 0.65Vcore)		1.73	40.8	0.95	27.7	0.73	18.6	0.27	5.25
OctoML	NRF5340DK	nRF5340	Arm® Cortex®-M33		microTVM using CMSIS-NN backend	128MHz		232.0		316.1		76.1		6.27	
OctoML	NUCLEO-L4R5ZI	STM32L4R5ZIT6U	Arm® Cortex®-M4		microTVM using CMSIS-NN backend	120MHz, 1.8Vbat		301.2	15531.4	389.5	20236.3	99.8	5230.3	8.60	443.2
OctoML	NUCLEO-L4R5ZI	STM32L4R5ZIT6U	Arm® Cortex®-M4		microTVM using native codegen	120MHz, 1.8Vbat		336.5	17131.6	389.2	21342.3	144.0	7950.5	11.7	633.7
Plumerai	B_U585I_IOT02A	STM32U585	Arm® Cortex®-M33		Plumerai Inference Engine 2022.09	160MHz		107.0		107.1		35.4		4.90	
Plumerai	CY8CPROTO-062-4343w	PSoC 62 MCU	Arm® Cortex®-M4		Plumerai Inference Engine 2022.09	150MHz		192.5		193.1		61.4		6.70	
Plumerai	DISCO-F746NG	STM32F746	Arm® Cortex®-M7		Plumerai Inference Engine 2022.09	216MHz		57.0		64.8		19.1		2.30	
Plumerai	NUCLEO-L4R5ZI	STM32L4R5ZIT6U	Arm® Cortex®-M4		Plumerai Inference Engine 2022.09	120MHz		208.6		173.2		71.7		5.60	
Silicon Labs	xG24-DK2601B	EFR32MG24	Arm® Cortex®-M33	Silicon Labs MVP(1)	TensorFlowLite for Microcontrollers, CMSIS-NN, Silicon Labs Gecko SDK			111.6	1139.2	120.9	1234.7	36.3	401.9	5.43	47.3
STMicroelectronics	NUCLEO-H7A3ZI-Q	STM32H7A3ZIT6Q	Arm® Cortex®-M7		X-CUBE-AI v7.3.0	280MHz, 3.3Vbat		50.7	7978.5	54.3	8707.3	16.8	2721.8	1.82	266.5
STMicroelectronics	NUCLEO-L4R5ZI	STM32L4R5ZIT6U	Arm® Cortex®-M4		X-CUBE-AI v7.3.0	120MHz, 1.8Vbat		230.5	10066.6	226.9	10681.6	75.1	3371.7	7.57	323.0
STMicroelectronics	NUCLEO-U575ZI-Q	STM32U575ZIT6Q	Arm® Cortex®-M33		X-CUBE-AI v7.3.0	160MHz, 1.8Vbat		133.4	3364.5	139.7	3642.0	44.2	1138.5	4.84	119.1
Syntiant	NDP9120-EVL	NDP120	M0 + HiFi	Syntiant Core 2 (98MH	Syntiant TDK	Syntiant Core 2 (98MHz, 1		4.10	97.2	5.12	139.4	1.48	43.8		
Syntiant	NDP9120-EVL	NDP120	M0 + HiFi	Syntiant Core 2 (30MH	Syntiant TDK	Syntiant Core 2 (30MHz, 0		12.7	71.7	16.0	101.8	4.37	31.5		
Qualcomm Innovation Center	Next Generation Snapdragon Mobile Platform HDK	Next Generation Snapdragon Mobile Platform	Qualcomm Kryo CPU(1)	Qualcomm Sensing Hub(1)	Qualcomm AI Stack									0.098	

RISC-V (PULP) Currently dominates the TinyML benchmarks

Extreme Edge Use Case: Nano-Drones

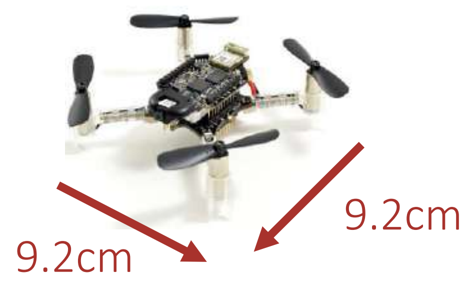
Advanced autonomous drone

[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

Nano-drone

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



- 3D Mapping & Motion Planning
- Object recognition & Avoidance
- 0.06m2 & **800g of weight**
- Battery Capacity **5410mAh**

- Smaller form factor of 0.008m2
- 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

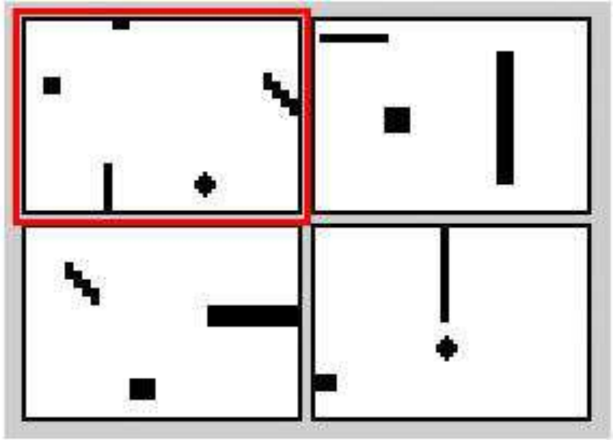
Object detection

Obstacle avoidance & Navigation

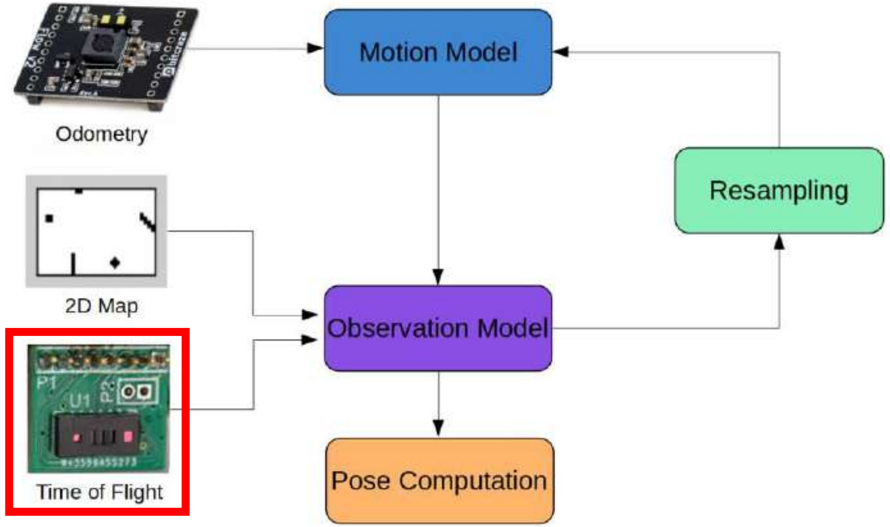
Environment exploration

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

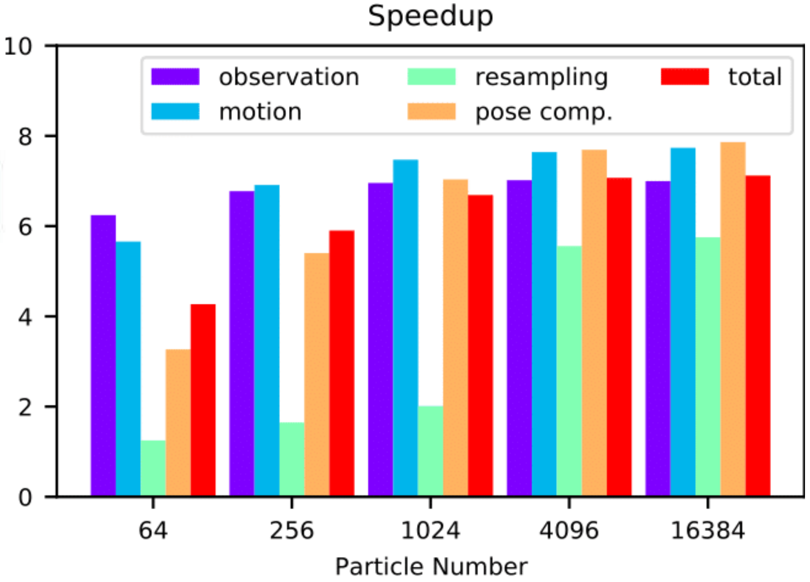
Monte Carlo Localization (Known Map)



Particle filter-based



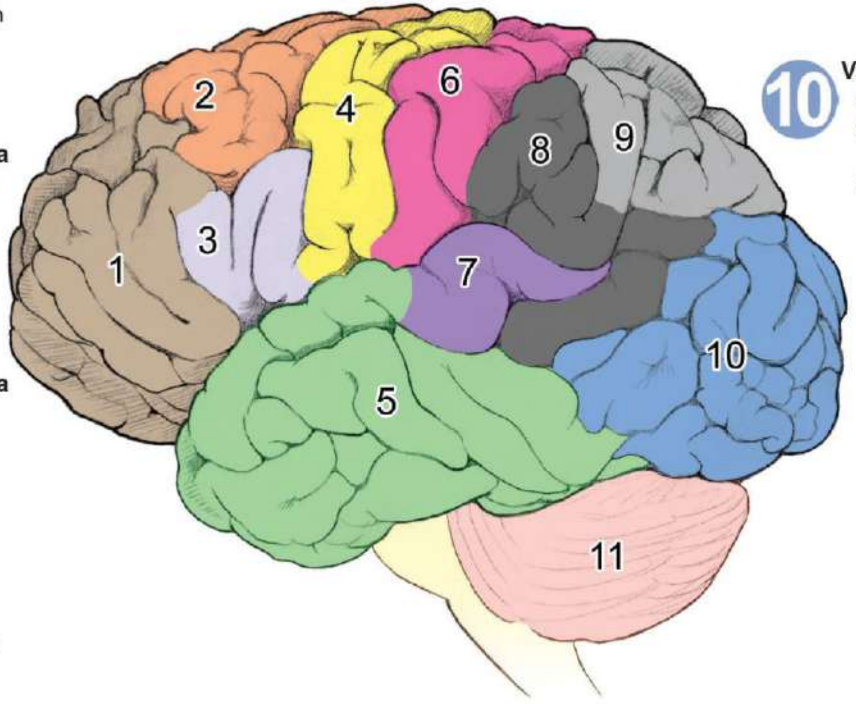
Convergence + Low ATE for Npart > 1024, 2ToF, FP16 acceptable



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

What’s next? Multiple Heterogeneous Accelerators

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

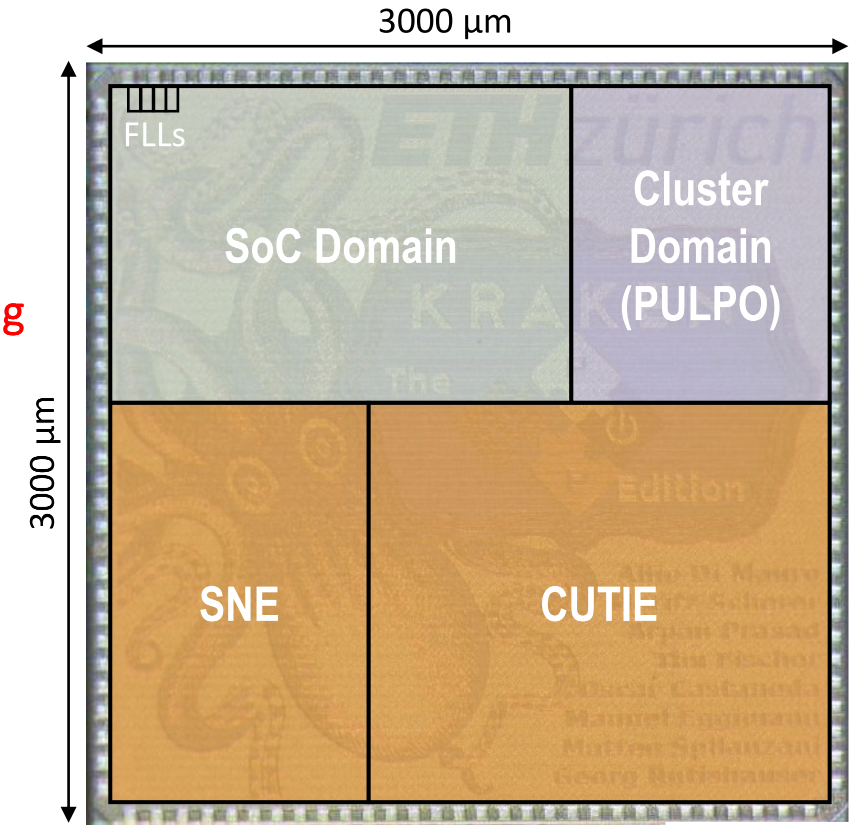
11 Motor Functions
- Coordination of movement
- Balance
- Posture

What’s next? 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
- PULPO – Floating point online optimizer (advanced control, state estimation…)

$$
\begin{aligned}
\mathbf{z} &= \mathbf{A}\mathbf{x} + \mathbf{y} && (\mathcal{M}) \\
\mathbf{z} &= \mathbf{y} - \tau \mathbf{A}^{H}\mathbf{x} && (\mathcal{H}) \\
\mathbf{z} &= \mathrm{prox}(\rho \mathbf{x}) && (\mathcal{P})
\end{aligned}
$$

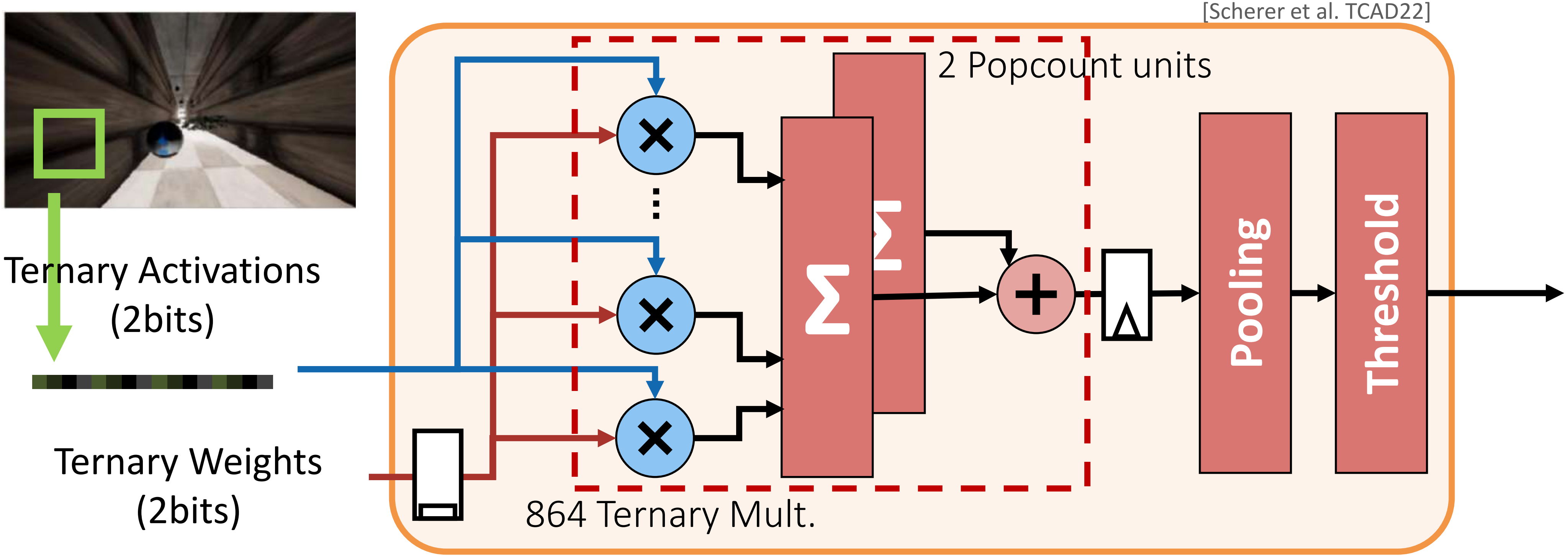


[Di Mauro HotChips22]

Technology	22 nm FDSOI
Chip Area	9 mm^2
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**

CUTIE: Minimize Switching Activity & Data Movement

[Scherer et al. TCAD22]



Ternary Activations (2bits)

Ternary Weights (2bits)

2 Popcount units

864 Ternary Mult.

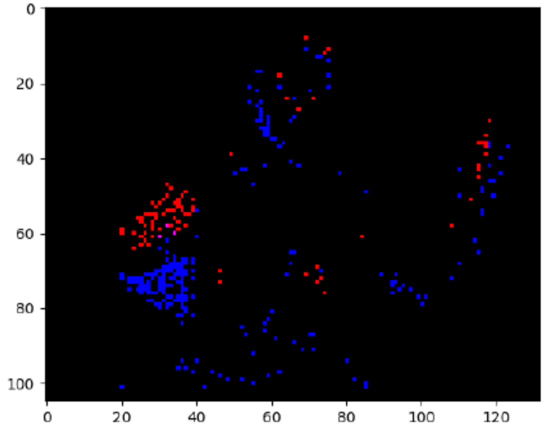
Output channel compute unit (OCU)

- 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 TMAC/cycle

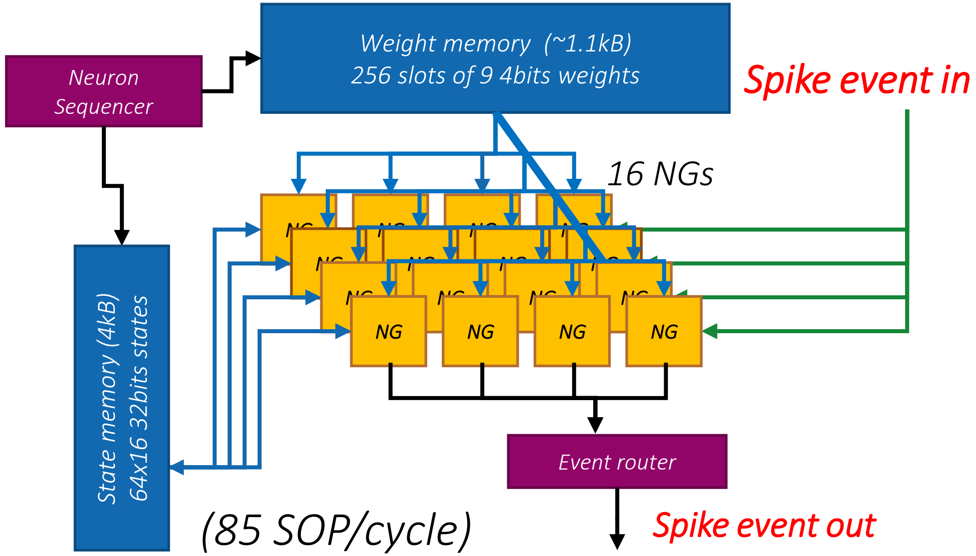
Aggressive quantization and full specialization

Different Sensor Type, different Acceleration Engine

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



[Di Mauro et al. DATE22]



Weight memory (~1.1kB)
256 slots of 9 4bits weights

Neuron Sequencer

Spike event in

16 NGs

NG

State memory (4kB)
64x16 32bits states

Event router

(85 SOP/cycle)

Spike event out

- SNE Engine: 16 Adaptive-LIF neuron data paths (NG). A NG executes one Synaptic Operation (SOP) per cycle
  - 1 SOP = 1 4b-ADD + 2 8b-MUL + 1 8b-ADD + 1 8b-CMP
- For fully connected layers one NG is time-shared for 64 virtual neurons
- Optimized buffering and neuron state update for 64x16 neurons in just 12 cycles for a 3x3 event receptive field
  - Equivalent number of 85 SOP/cycle per engine (682 SOP/cycle on 8 engines)

SNE works seamlessly with DVS (event-based) sensors

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)** ➡ **XPULPV2 &V3**

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

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

*sub 1fJ in 7nm

Advancing the SOA on all tasks

RISC-V Cluster

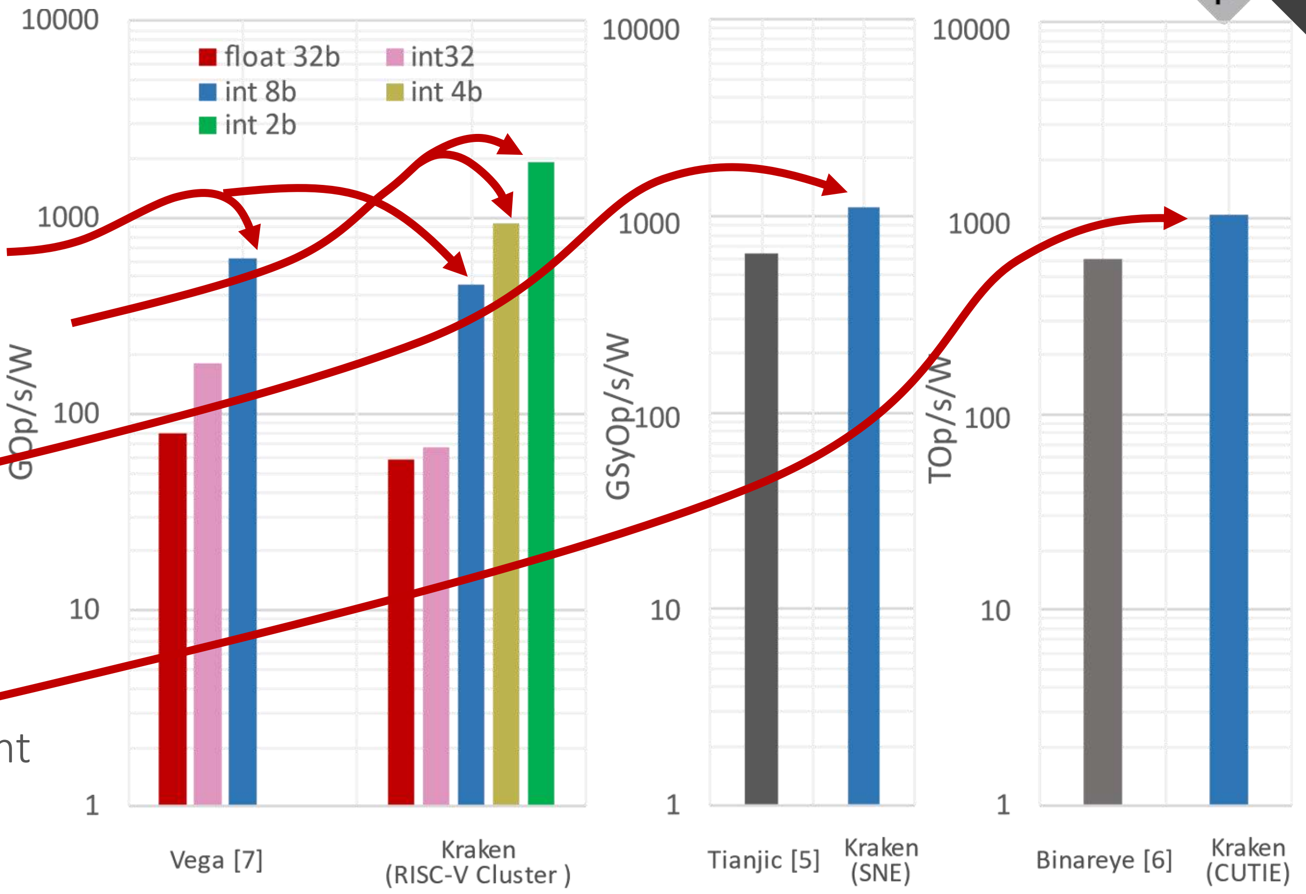
- Comparable 32bits-8bits SOA Energy efficiency to other PULPs [7]
- **The highest energy efficiency on sub-byte SIMD operations (4b-2b)**

SNE

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

CUTIE

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

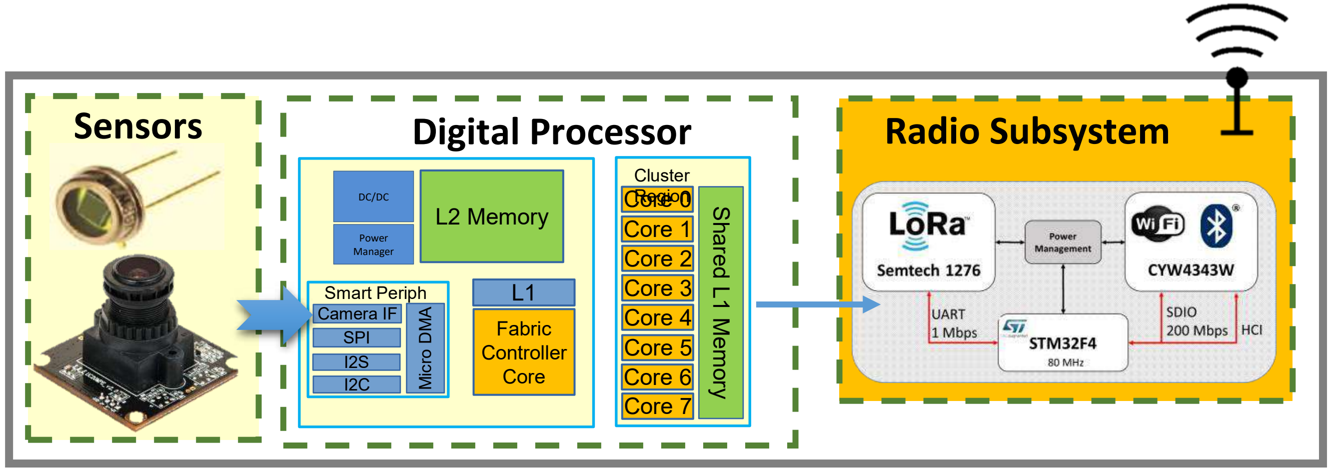


CUTIE, SNE can work concurrently for SNN + TNN "fused" inference (never done so far)

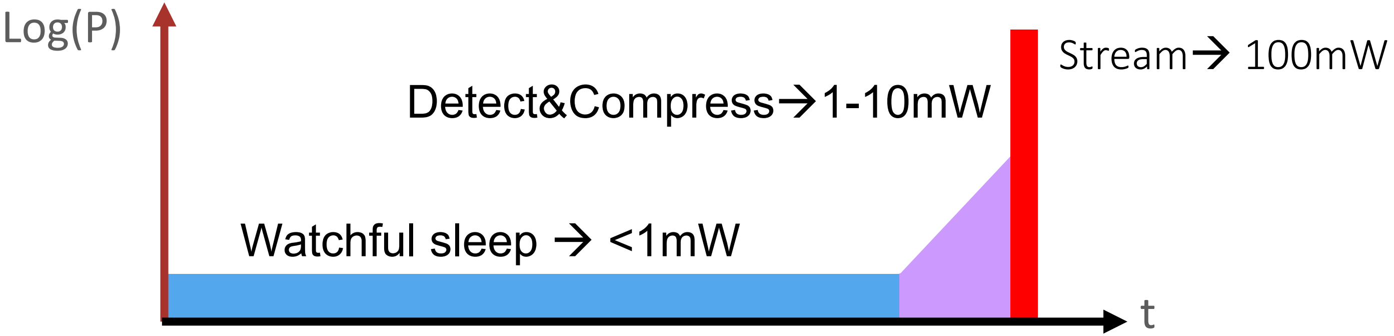
[5] L. Deng et al., "Tianjic," JSSC 2020
[6] B. Moons et al., "Binareye," CICC, 2018
[7] D. Rossi et al., "Vega," JSSC 2022.

Not only Efficiency: Achieving sub-mW Average Power?

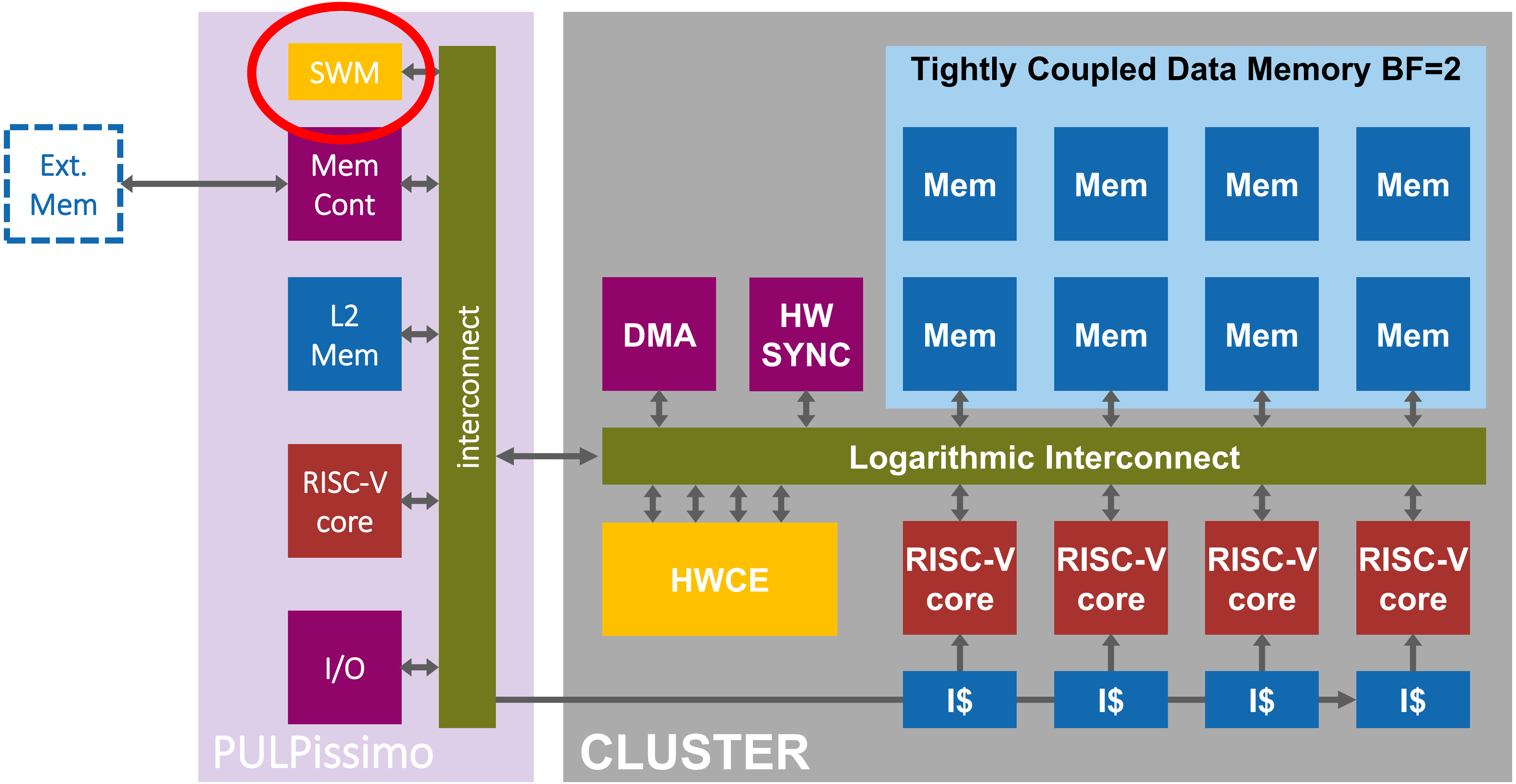
1mW average power with 10mW active power (10GOPS @ 1pJ/OP) → sub mW sleep



Duty cycling not acceptable when input events are asynchronous → **watchful Sleep**

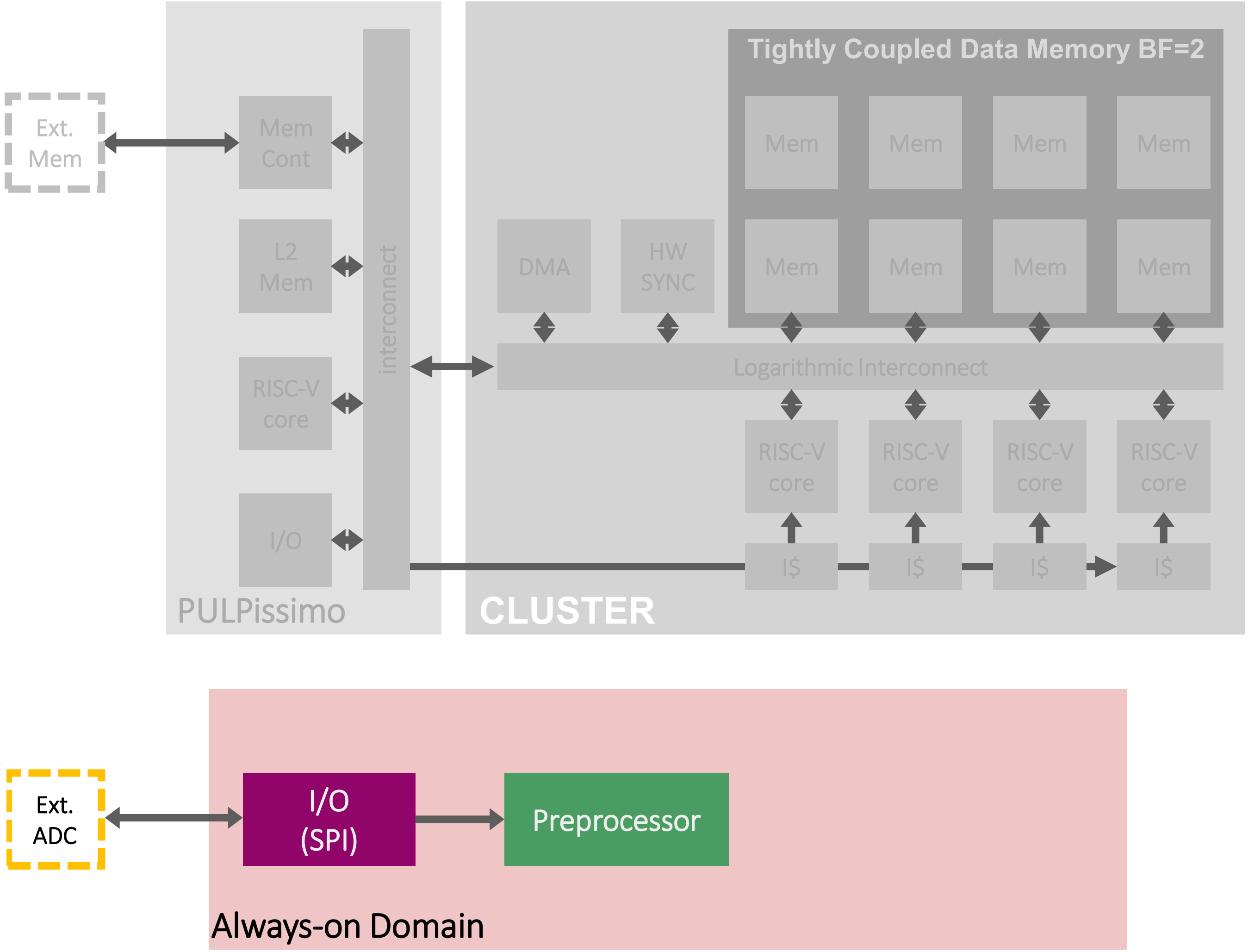


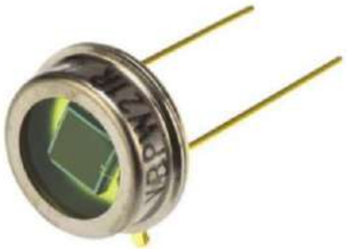
Need µW-range always-on Intelligence



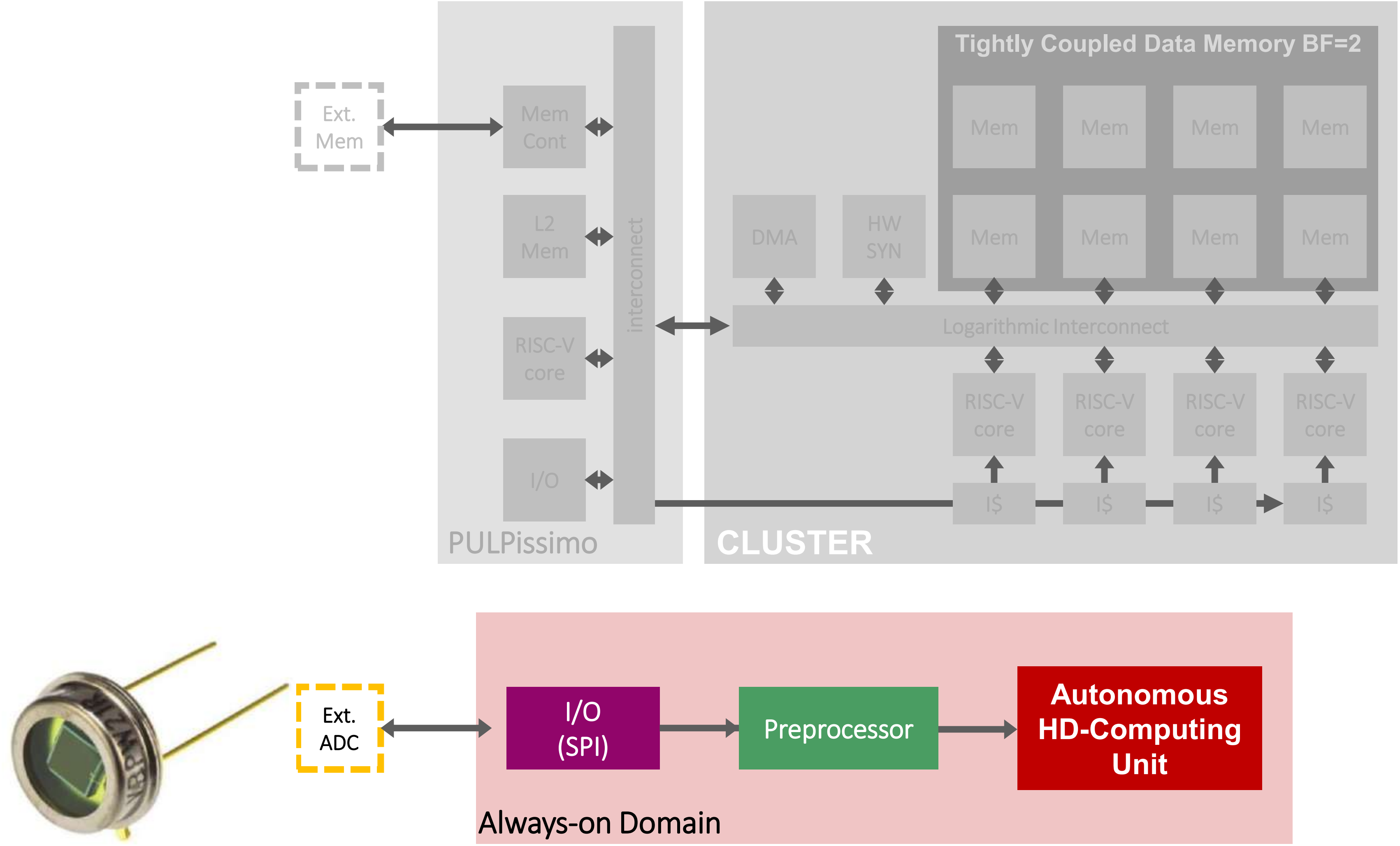
Smart Wakeup Module

HD-Based smart Wake-Up Module

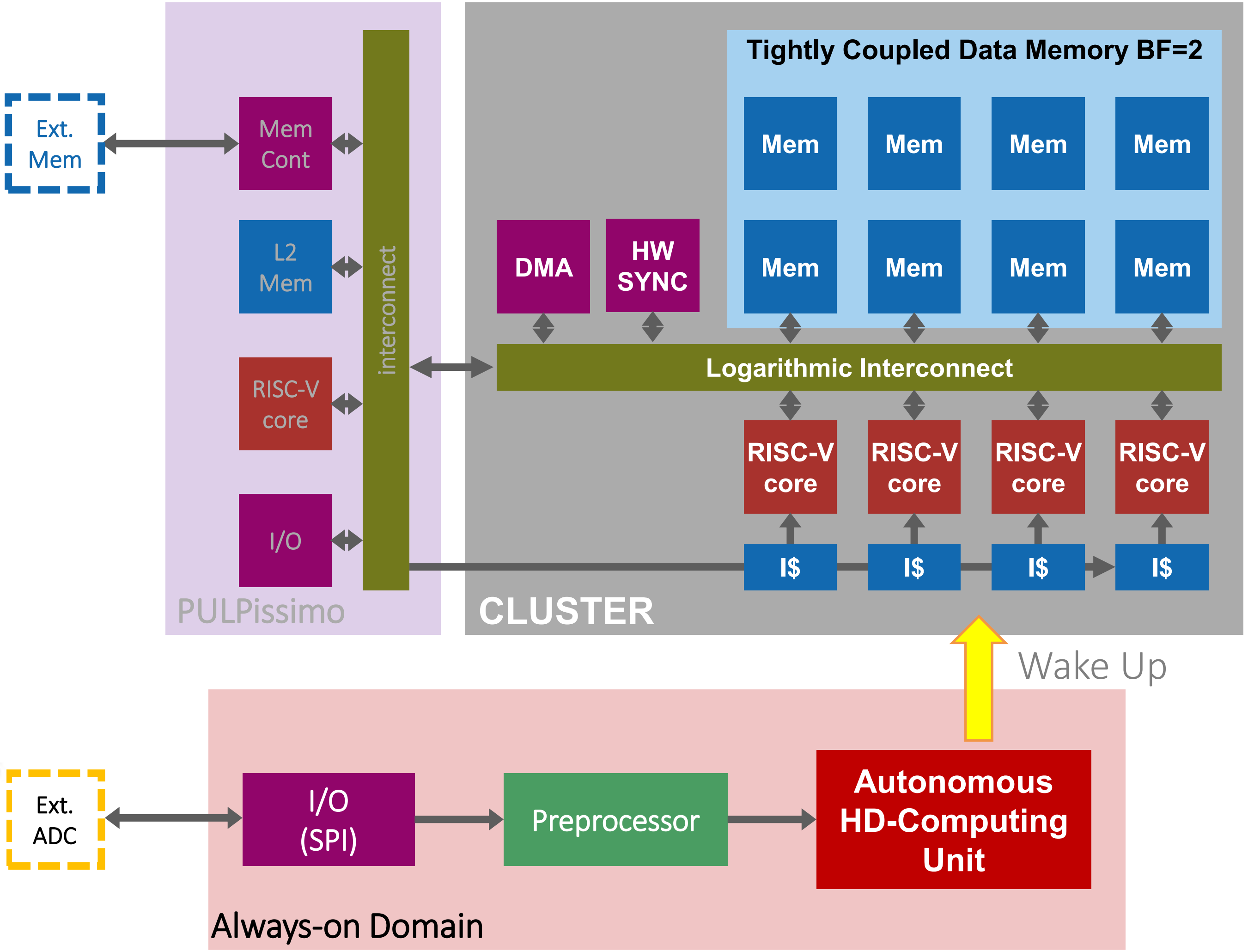


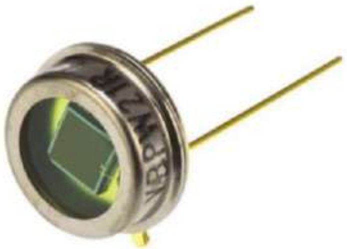


HD-Based smart Wake-Up Module

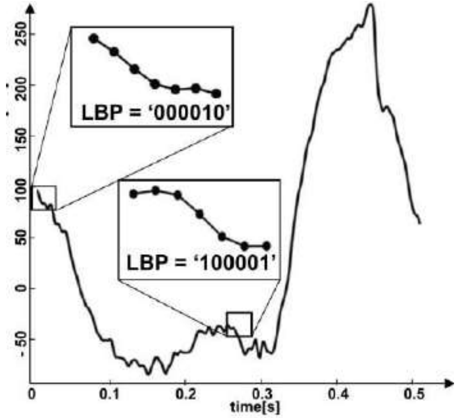


HD-Based smart Wake-Up Module





Not Only CNNs: Hyper-Dimensional Computing

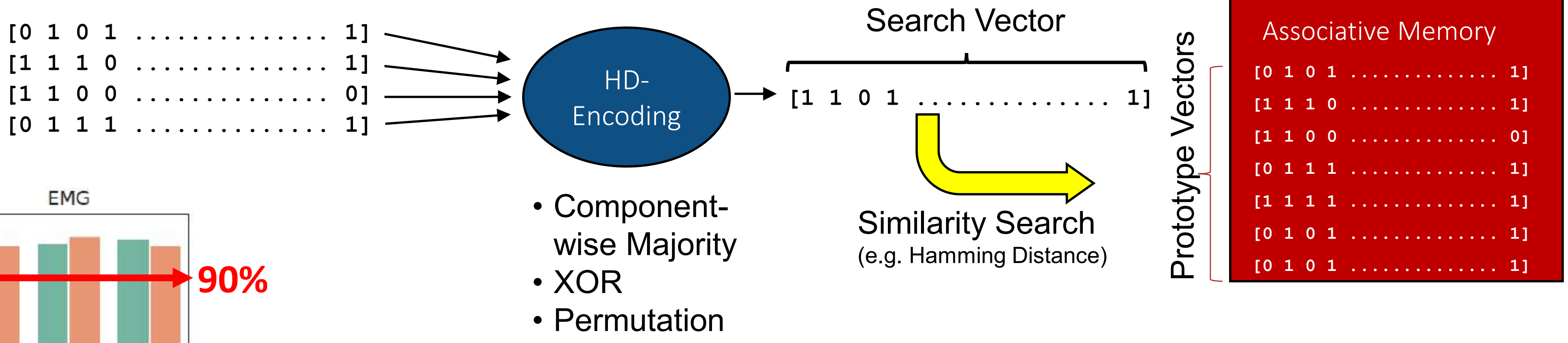


Mapping →

1st 2nd 3rd 4th 1000th

[0 1 0 1 .... 1]

Low Dimensional Input Data (e.g. 7-bit LBP)



[0 1 0 1 .... 1]
[1 1 1 0 .... 1]
[1 1 0 0 .... 0]
[0 1 1 1 .... 1]

HD-Encoding

- Component-wise Majority
- XOR
- Permutation

Search Vector

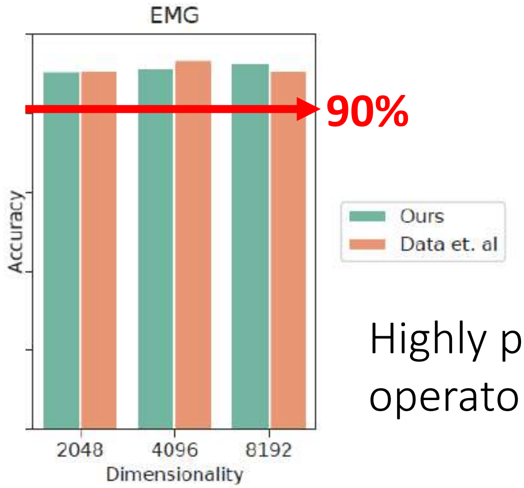
[1 1 0 1 .... 1]

Similarity Search
(e.g. Hamming Distance)

Prototype Vectors

Associative Memory

[0 1 0 1 .... 1]
[1 1 1 0 .... 1]
[1 1 0 0 .... 0]
[0 1 1 1 .... 1]
[1 1 1 1 .... 1]
[0 1 0 1 .... 1]
[0 1 0 1 .... 1]

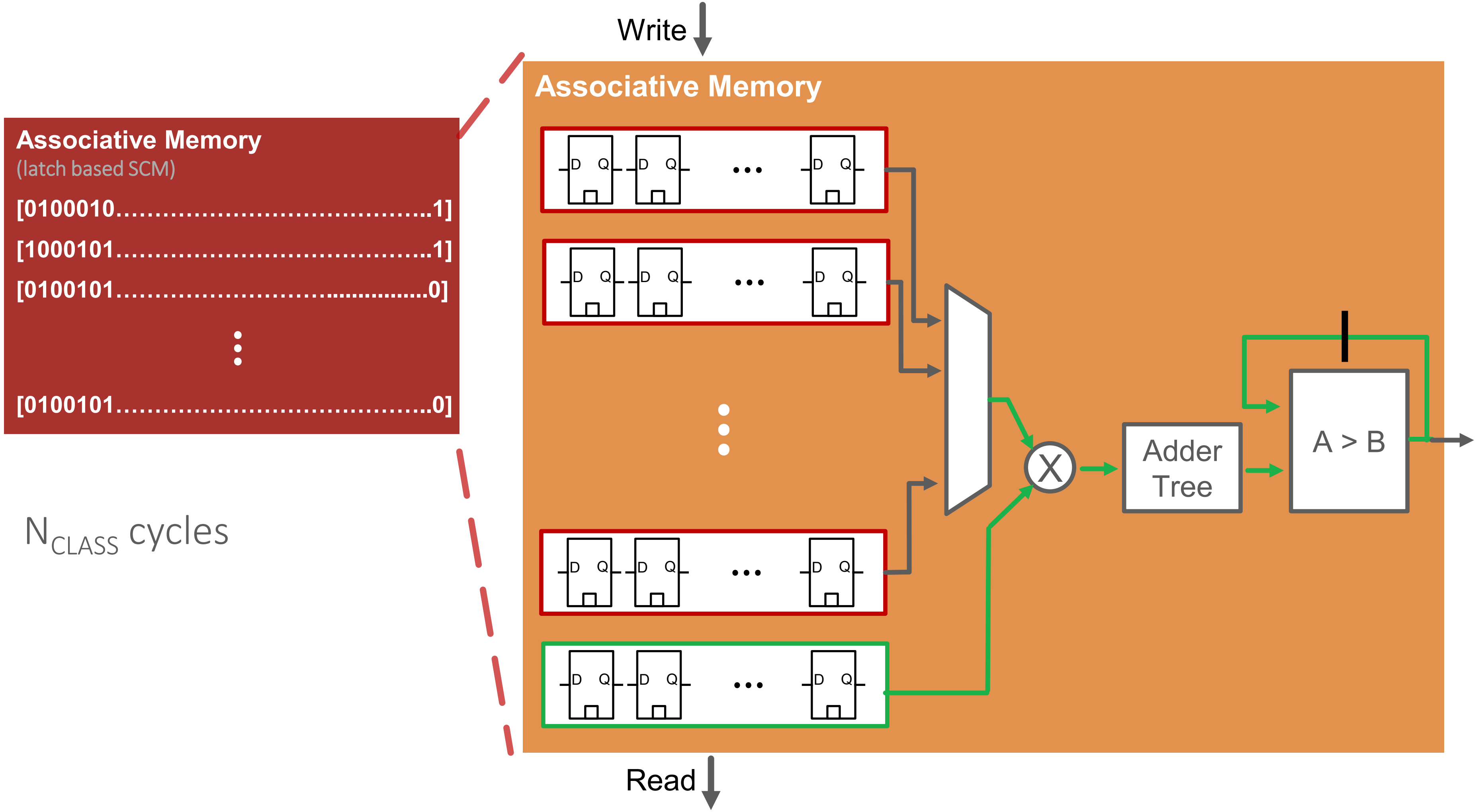


Highly parallel, fault-tolerant binary operators, assoc-min-distance search

➡

Merge storage & computation
i.e. In-memory computing

In-memory Hyperdimensional Computing



Associative Memory
(latch based SCM)

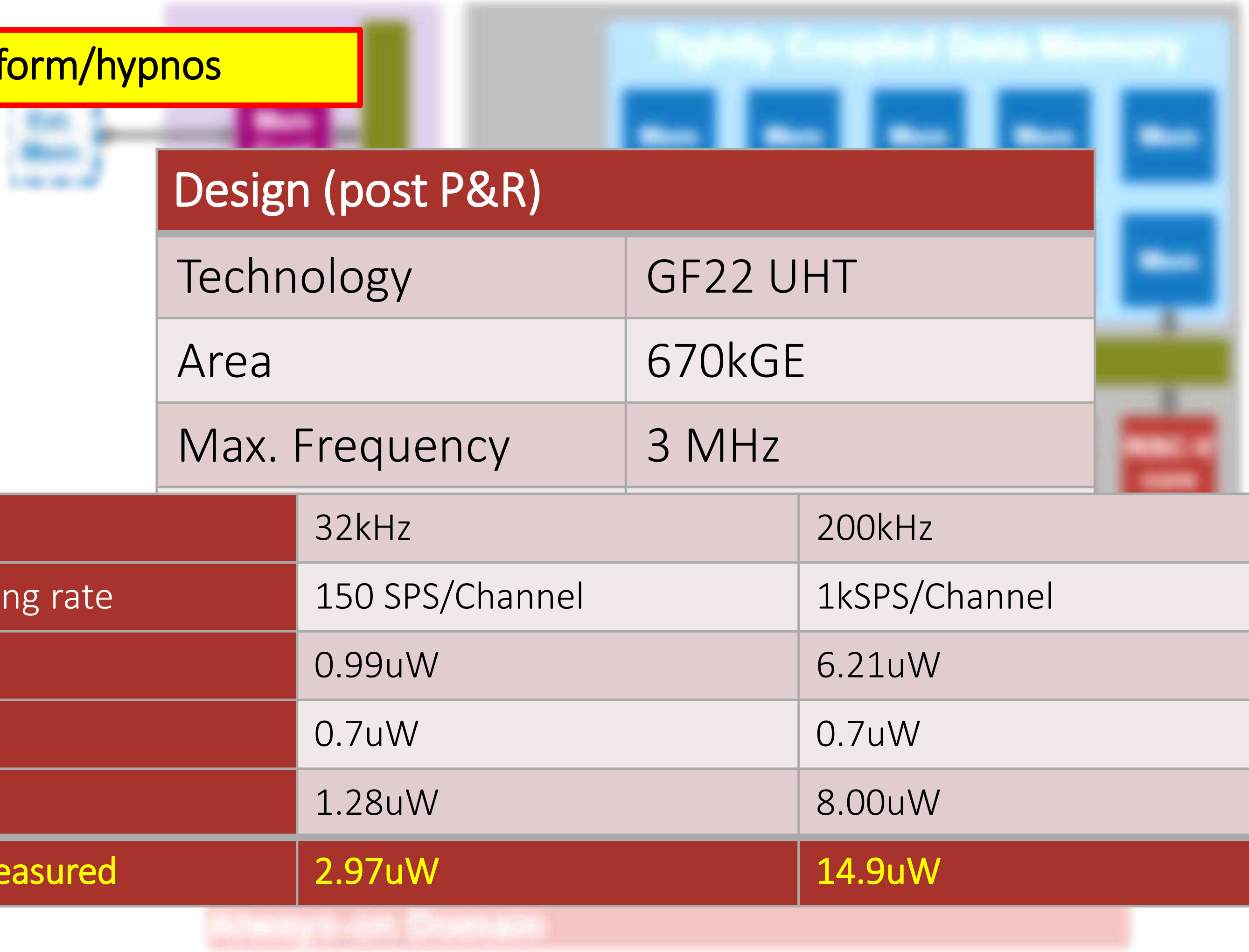
[0100010……………………………………..1]
[1000101……………………………………..1]
[0100101……………………………...............0]
⋮
[0100101……………………………………..0]

N_{CLASS} cycles

HD-Based smart Wake-Up Module - Hypnos

[Eggiman et al. TCAS22]

github.com/pulp-platform/hypnos



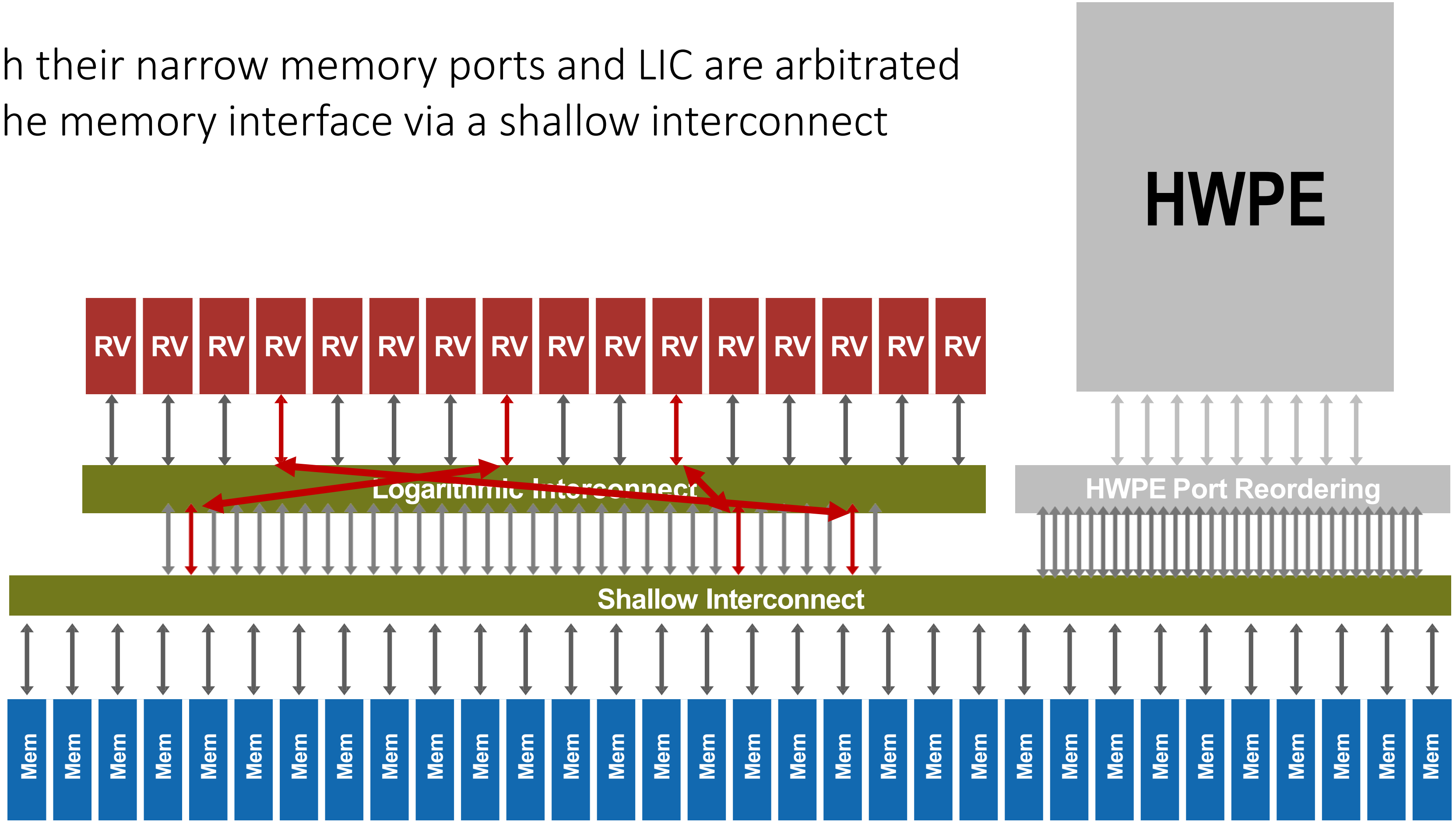
Design (post P&R)	
Technology	GF22 UHT
Area	670kGE
Max. Frequency	3 MHz

f_{clk}	32kHz	200kHz
max. sampling rate	150 SPS/Channel	1kSPS/Channel
$P_{SWU, dynamic}$	0.99uW	6.21uW
$P_{SWU, leakage}$	0.7uW	0.7uW
$P_{SPI, dynamic}$	1.28uW	8.00uW
$P_{SWU, total}$ Measured	2.97uW	14.9uW

Implemented with lowest leakage cell library (UHVT)

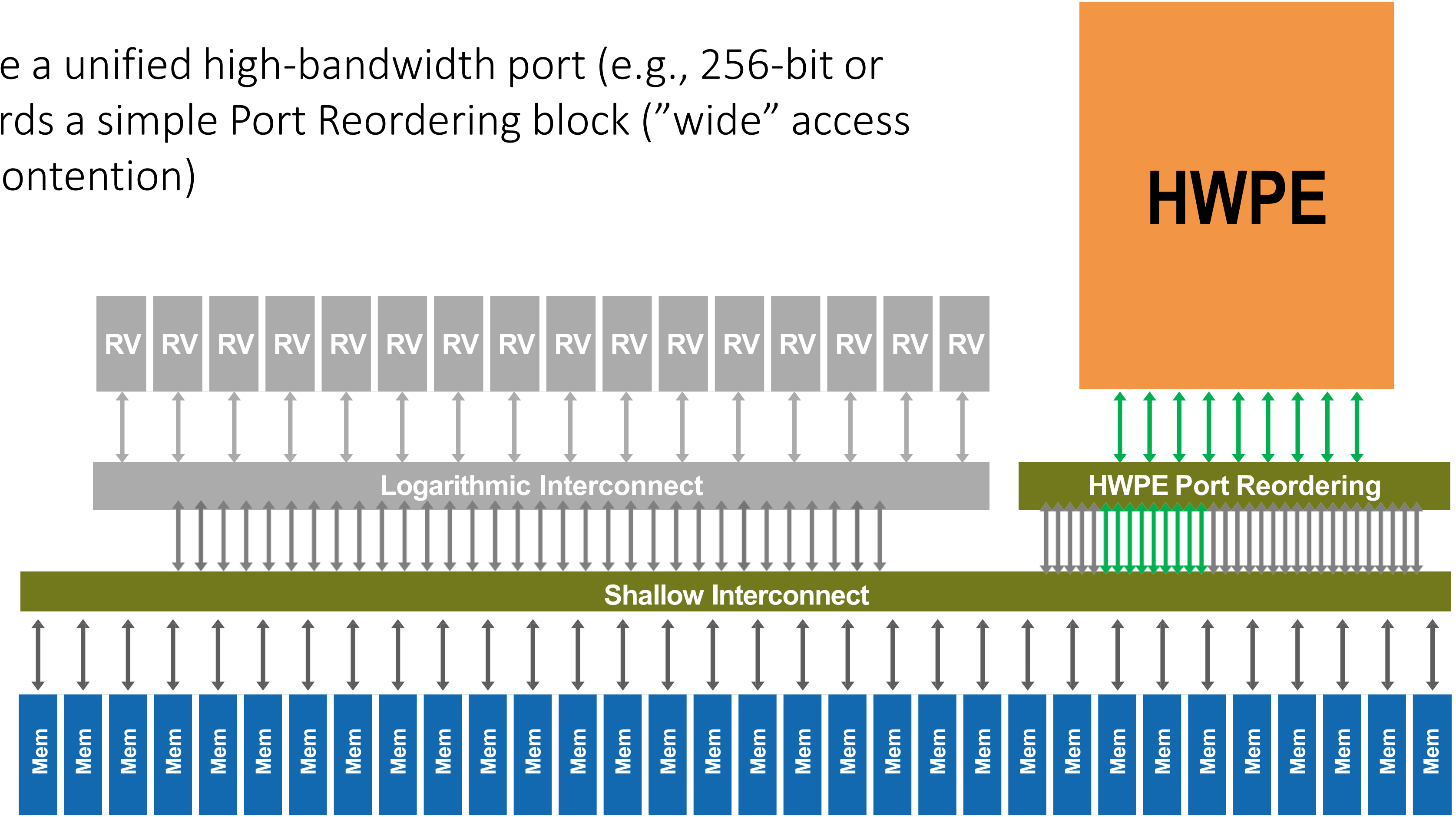
Many Accelerators → Dataflow Orchestration Challenge

Processors with their narrow memory ports and LIC are arbitrated vs. HWPEs at the memory interface via a shallow interconnect



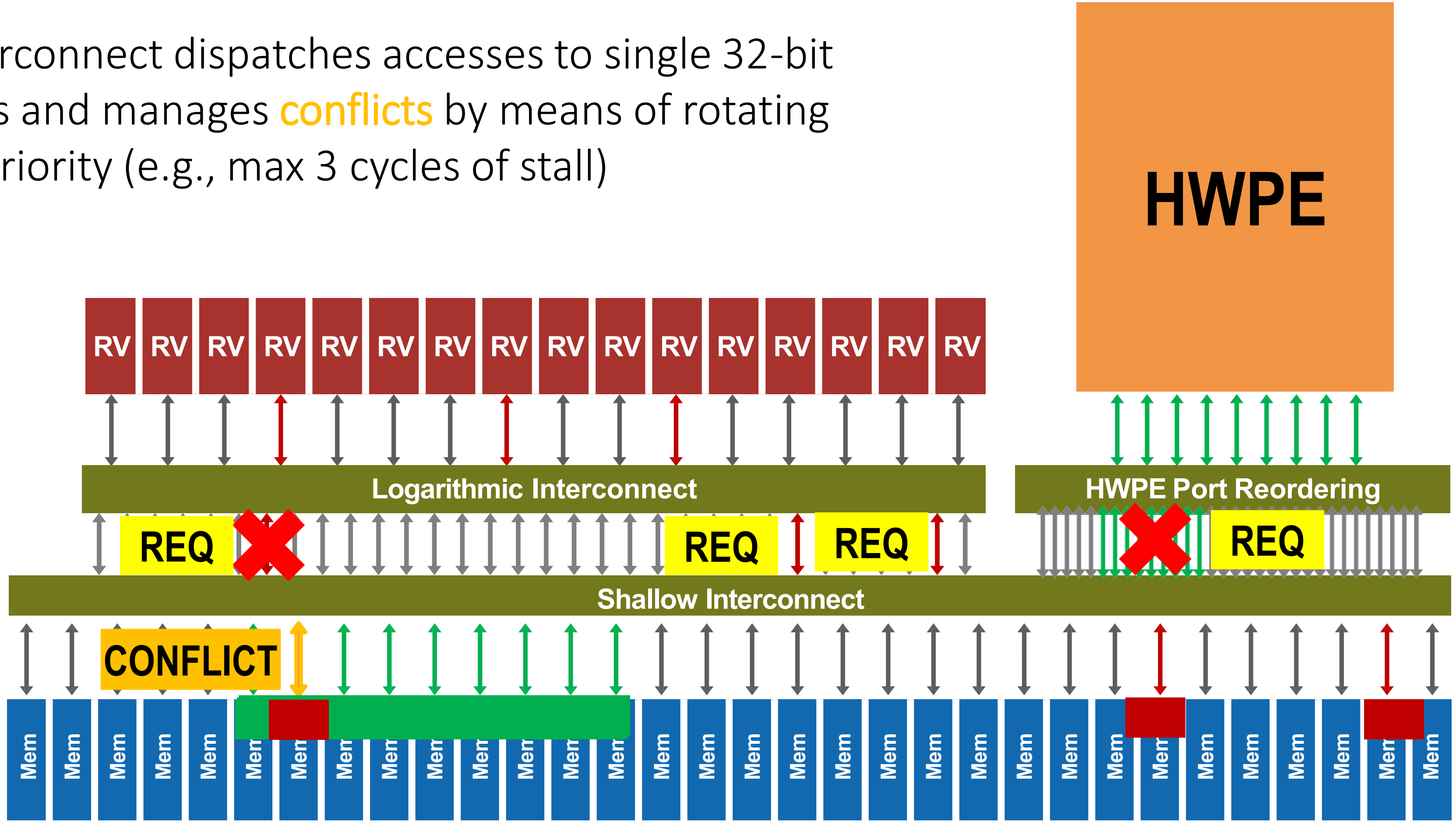
High-Bandwidth access: Heterogeneous Cluster Interconnect

HWPEs expose a unified high-bandwidth port (e.g., 256-bit or 512-bit) towards a simple Port Reordering block (”wide” access without self-contention)



High-Bandwidth access: Heterogeneous Cluster Interconnect

A Shallow Interconnect dispatches accesses to single 32-bit memory banks and manages conflicts by means of rotating configurable priority (e.g., max 3 cycles of stall)

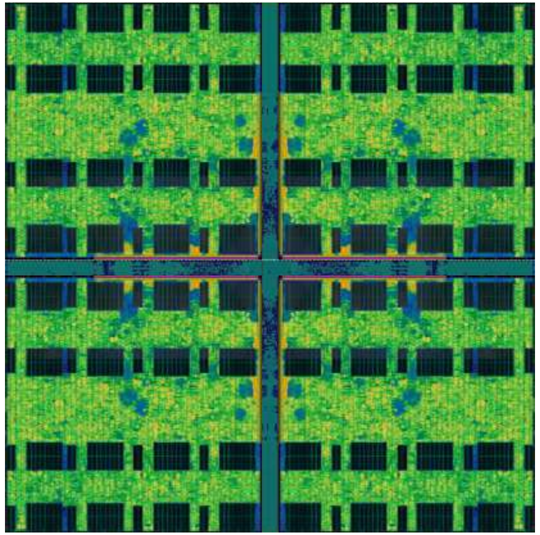


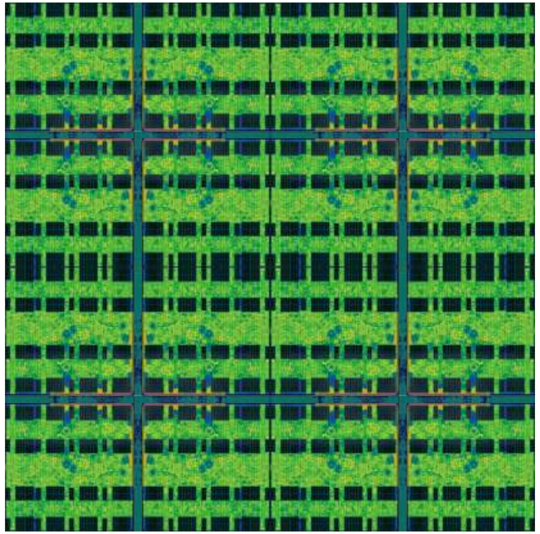
How much can we scale it? … a lot with a bit of latency!

ETH zürich
ALMA MATER STUDIORUM UNIVERSITÀ DI BOLOGNA

Scaling up: The MemPool Family

Hierarchical low-latency interconnect + Latency-Tolerant Core (snitch)





MinPool: first tape-out

- 16 cores, 64 KiB, 3cycles
- TSMC 65

MemPool: main driver

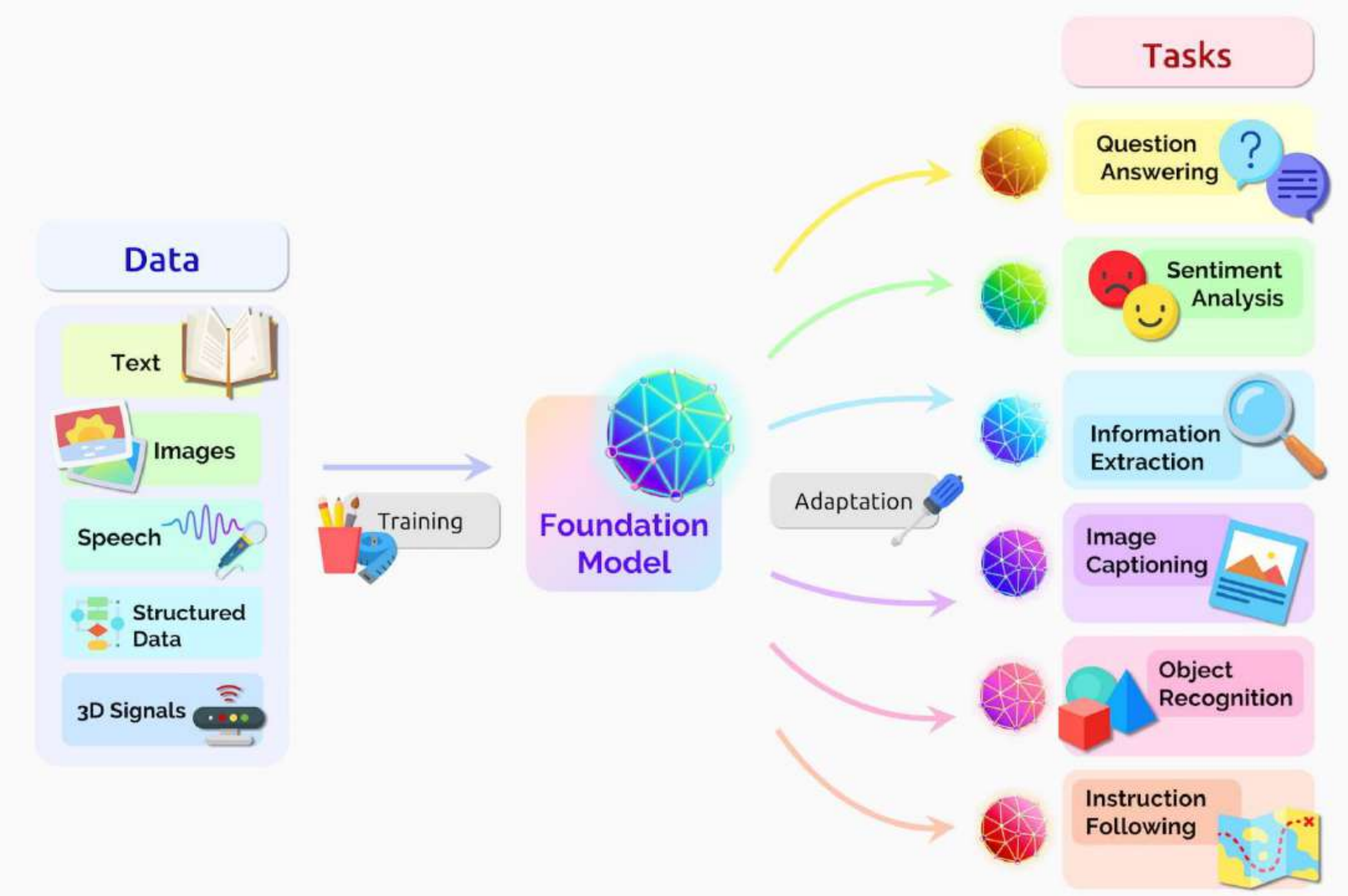
- 256 cores, 1 MiB, 5cycles
- GF 22FDX
  - 500 MHz (wc)
- MemPool-3D

TeraPool: going even bigger

- 1024 cores, 4 MiB, 7cycles
- GF 12 (500MHz+)
- Break the **TOPS** barrier
- TeraPool-3D?

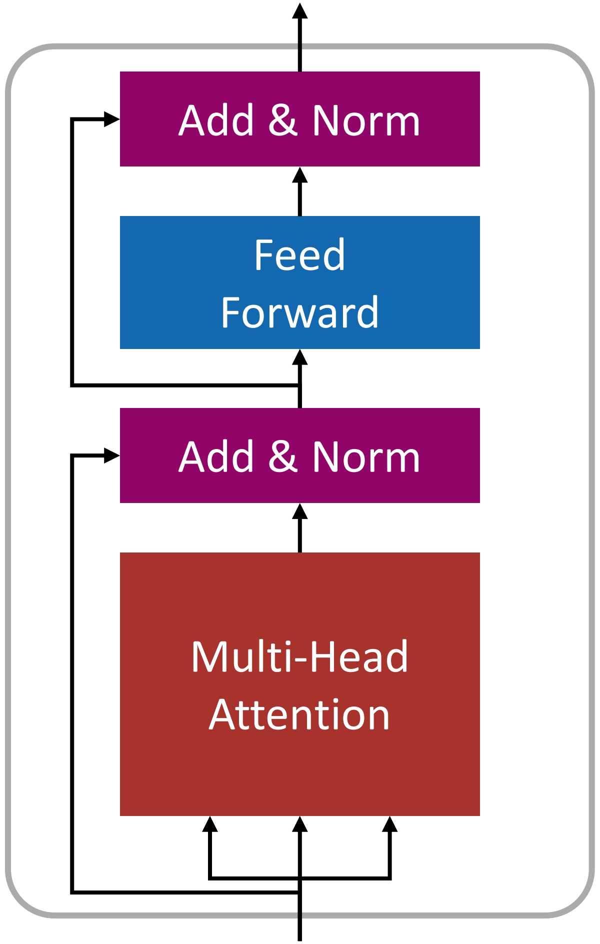
Why Scaleup? The Era of Foundation Models

- Versatility
  - Natural language processing, computer vision, robotics, biology, …
- Homogenization of models
  - Transformers as *foundation models!*
- Transfer learning
  - Train on a large-scale dataset and fine-tune on specific tasks with smaller datasets.



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!



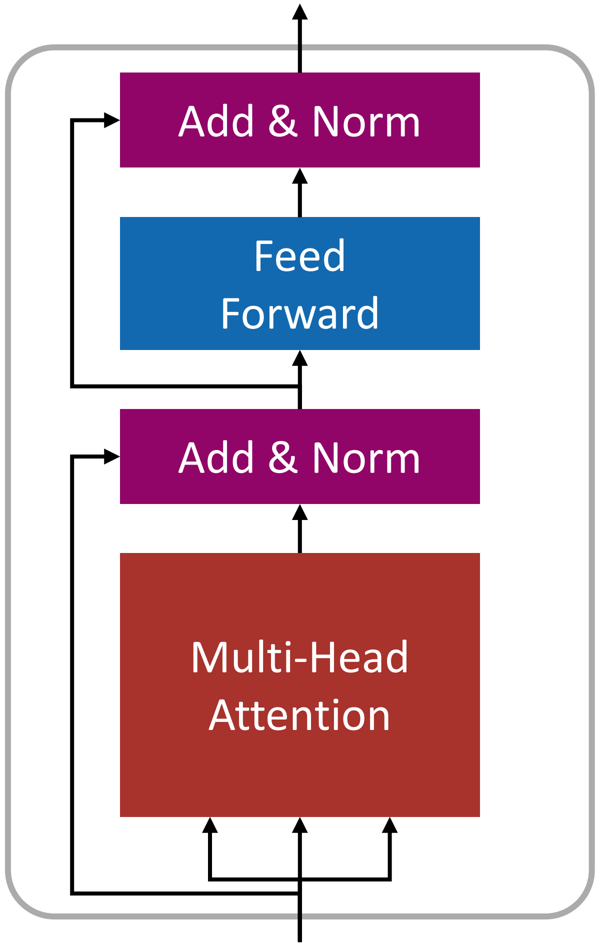
Add & Norm

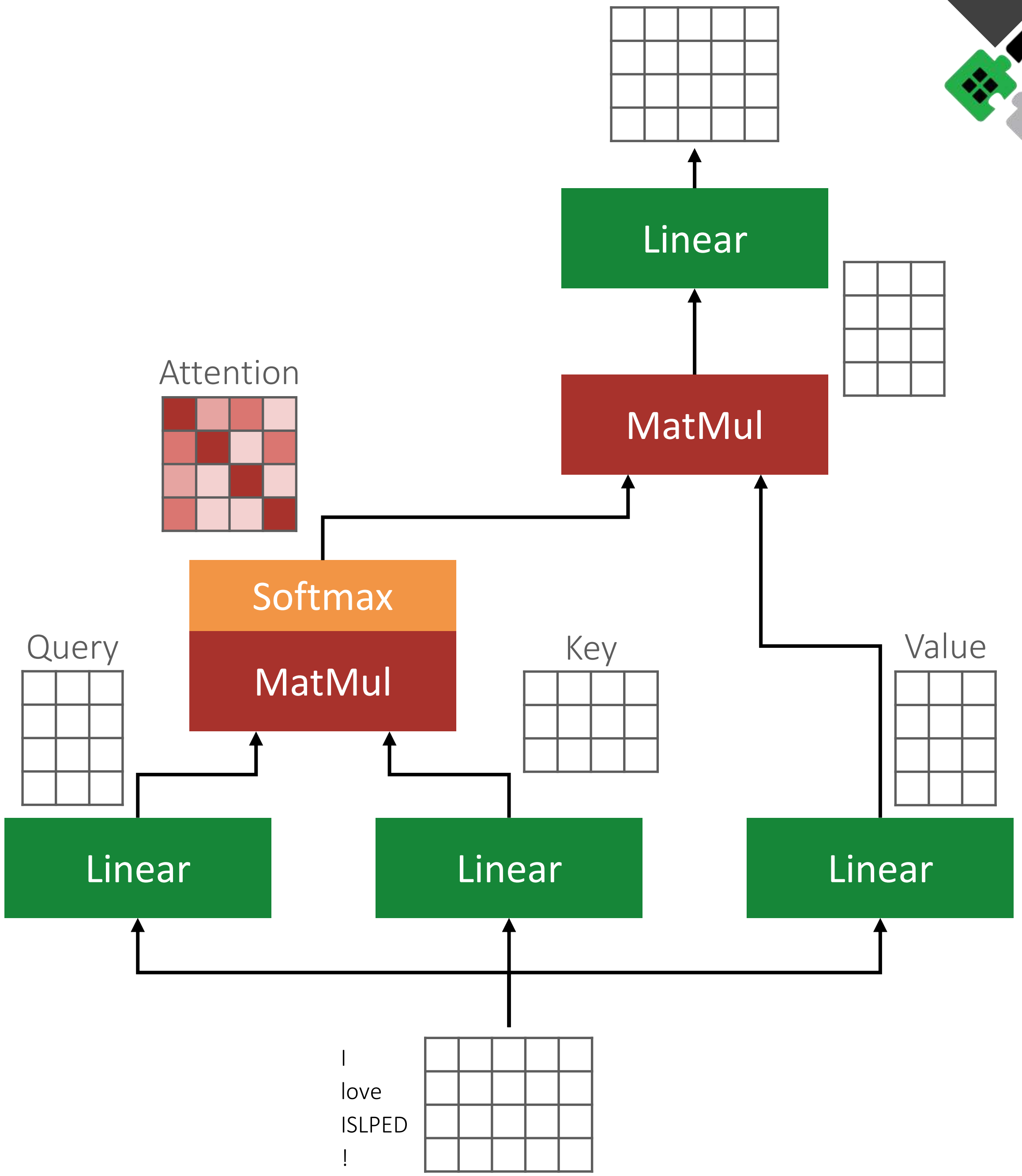
Feed Forward

Add & Norm

Multi-Head Attention

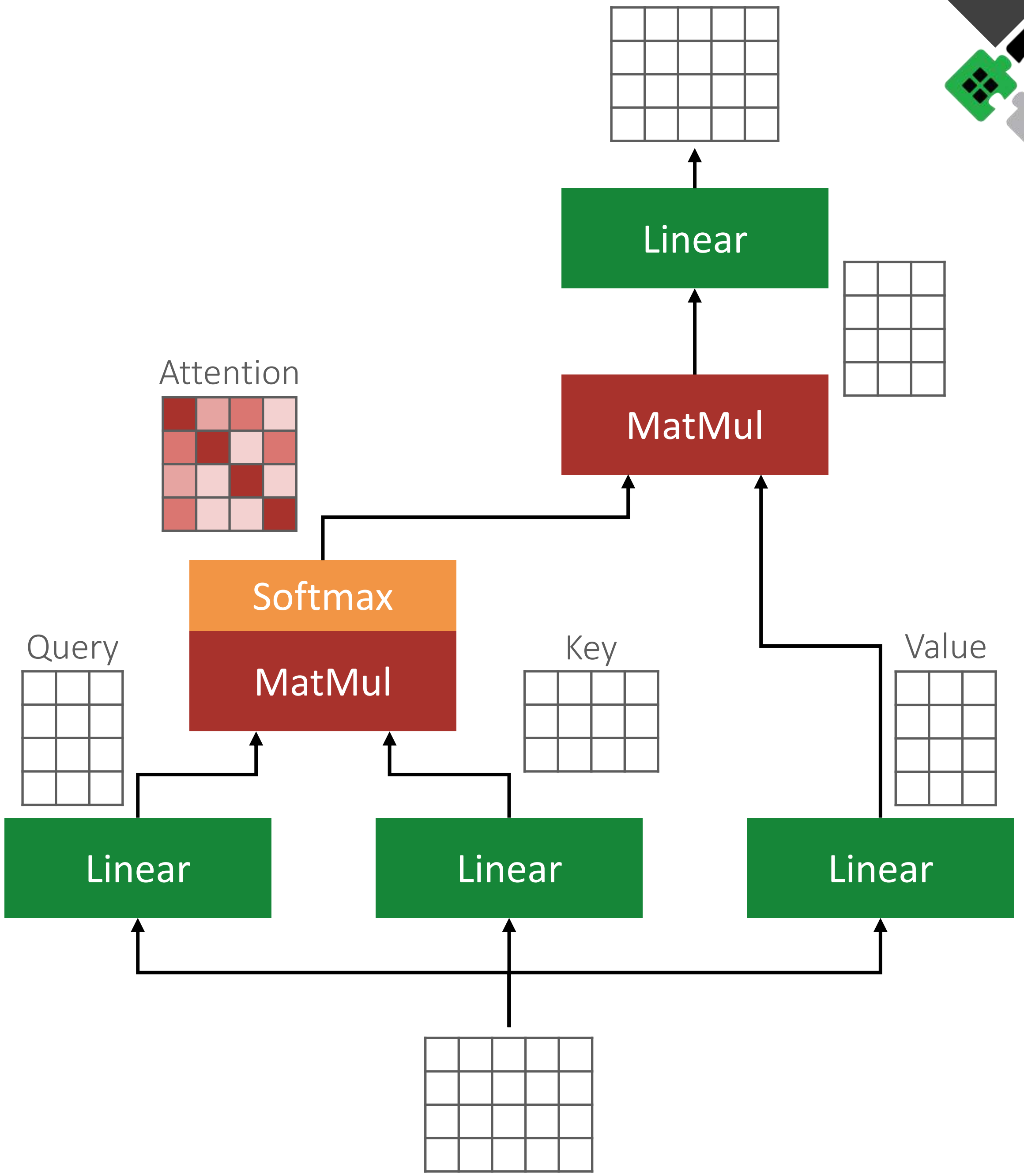
Attention but how?





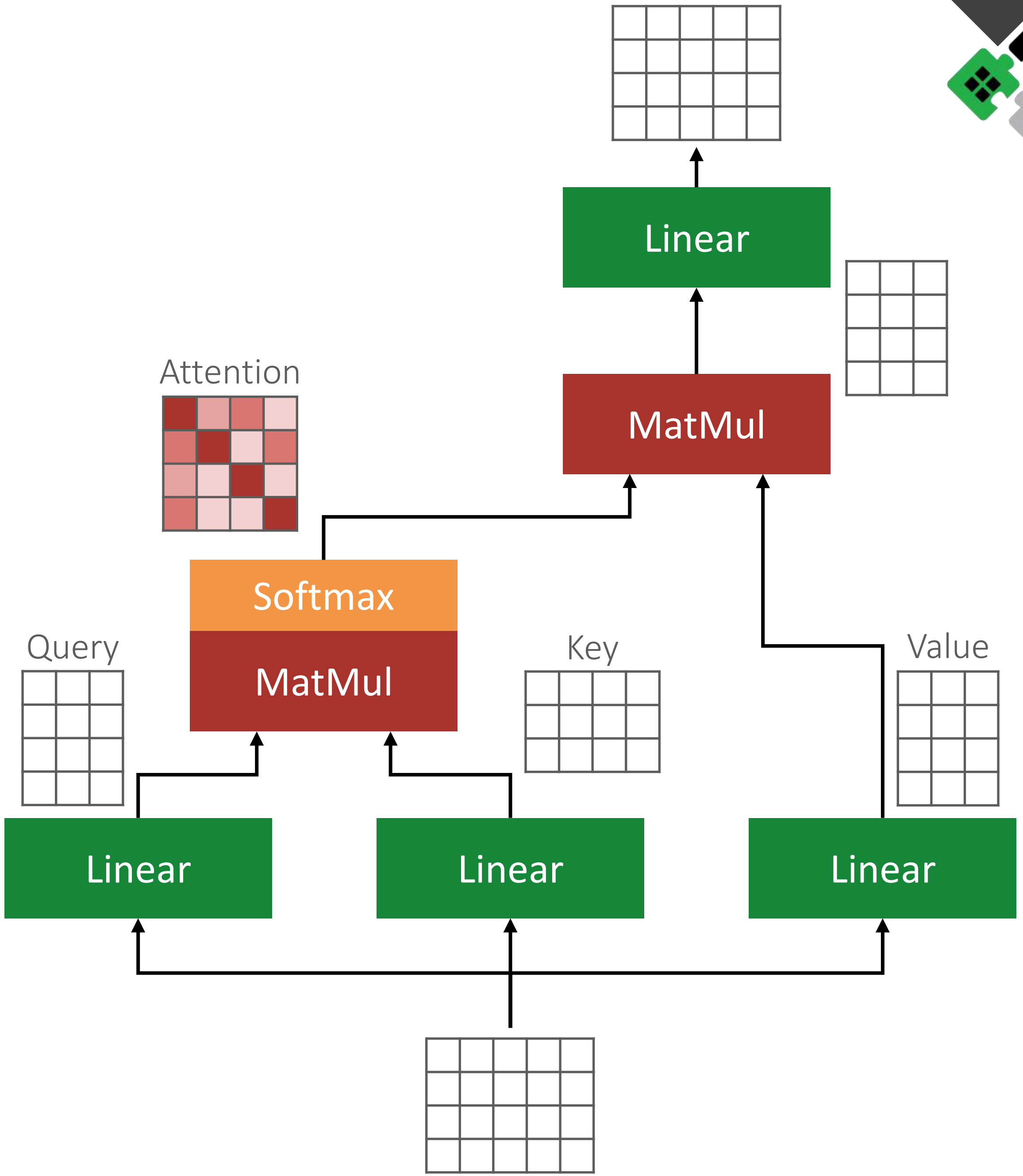
Challenges in ***Attention***

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



Challenges in ***Attention***

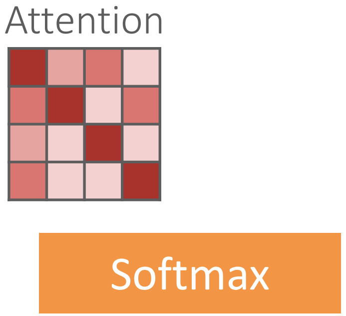
- Attention matrix is a square matrix of order input length.
  - Computational complexity
  - Memory requirements
- Every attention layer applies *Softmax* to attention matrix!



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.



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

ITA: Integer Transformer Accelerator

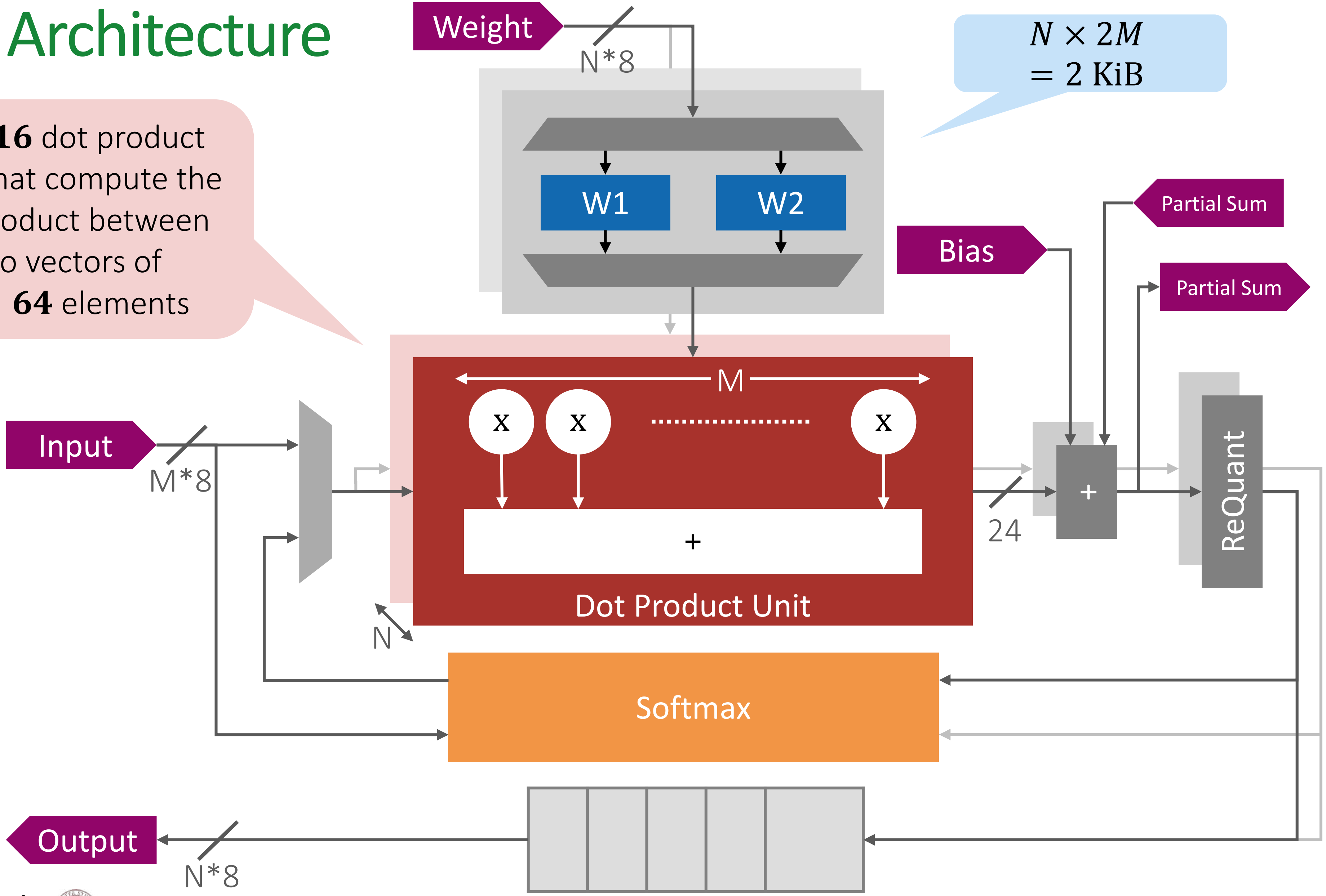
[Islamoglu et al. ISLPED23]

- ***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!

ITA – Architecture

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

$N \times 2M = 2\ \text{KiB}$



Hardware-friendly *Softmax*

$$\mathrm{Softmax}(\mathbf{x})_{\mathrm{i}} = \frac{\mathrm{e}^{\mathrm{x_i} - \max(\mathbf{x})}}{\sum_{\mathrm{j}}^{\mathrm{n}} \mathrm{e}^{\mathrm{x_j} - \max(\mathbf{x})}}$$

Softmax

Hardware-friendly *Softmax*

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

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

Softmax

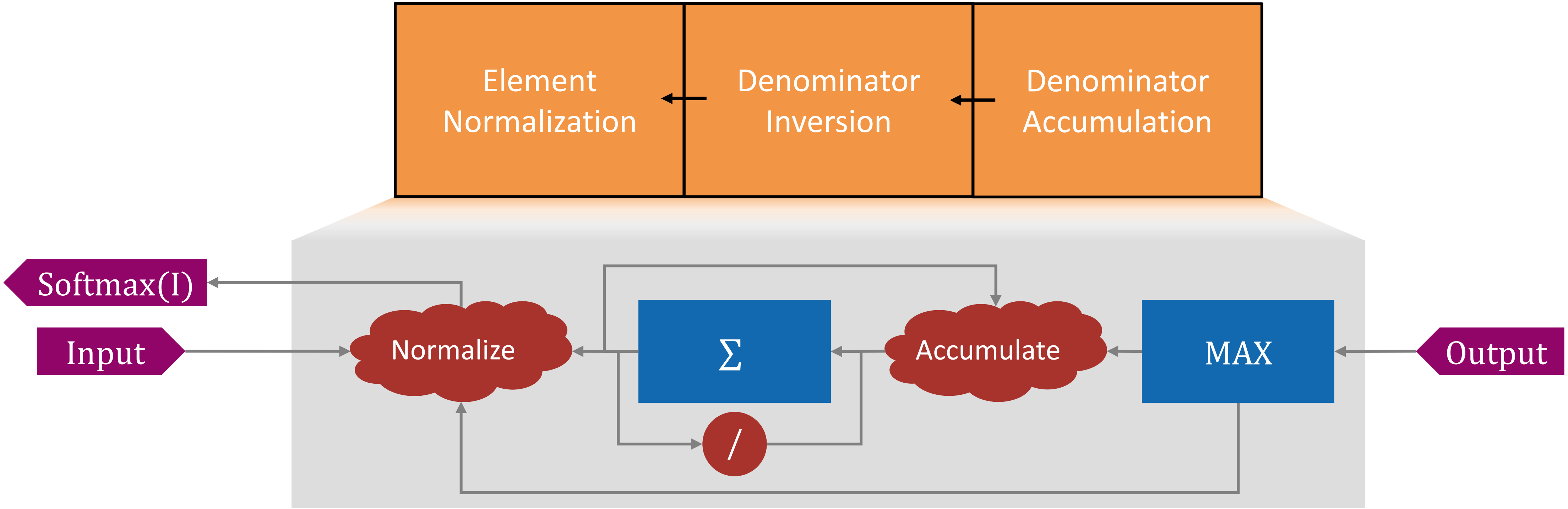
Directly operates on quantized values.

No exponentiation modules and multipliers.

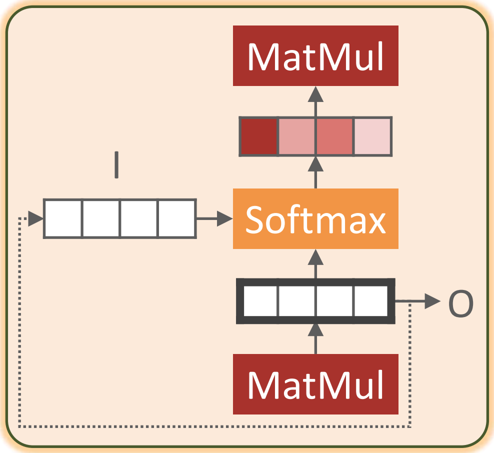
Computes softmax on streaming data.

Hardware-friendly *Softmax*

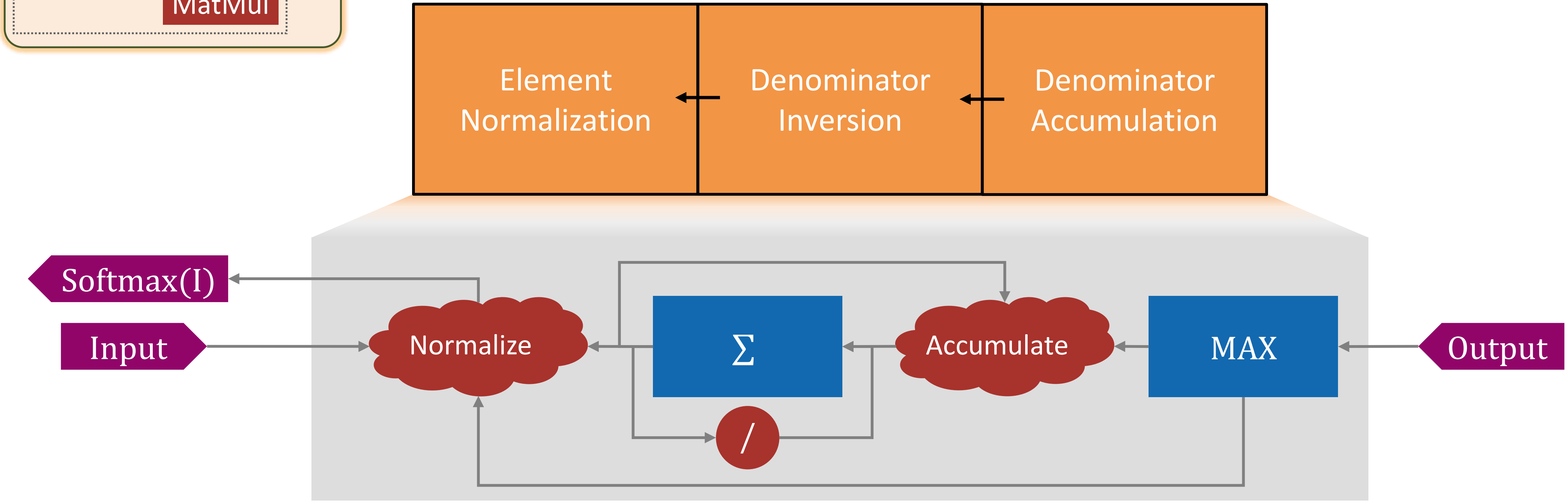
$$\mathrm{Softmax}(\mathbf{x})_i = \frac{1}{\sum_j^n 2^{(x_{qj} - \max(\mathbf{x}_q)) \gg 5}} 2^{(x_{qi} - \max(\mathbf{x}_q)) \gg 5}$$



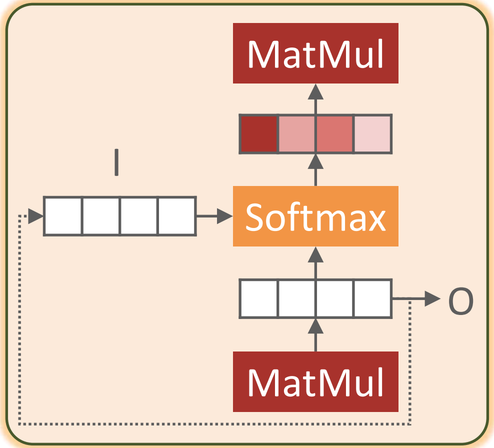
Hardware-friendly *Softmax*



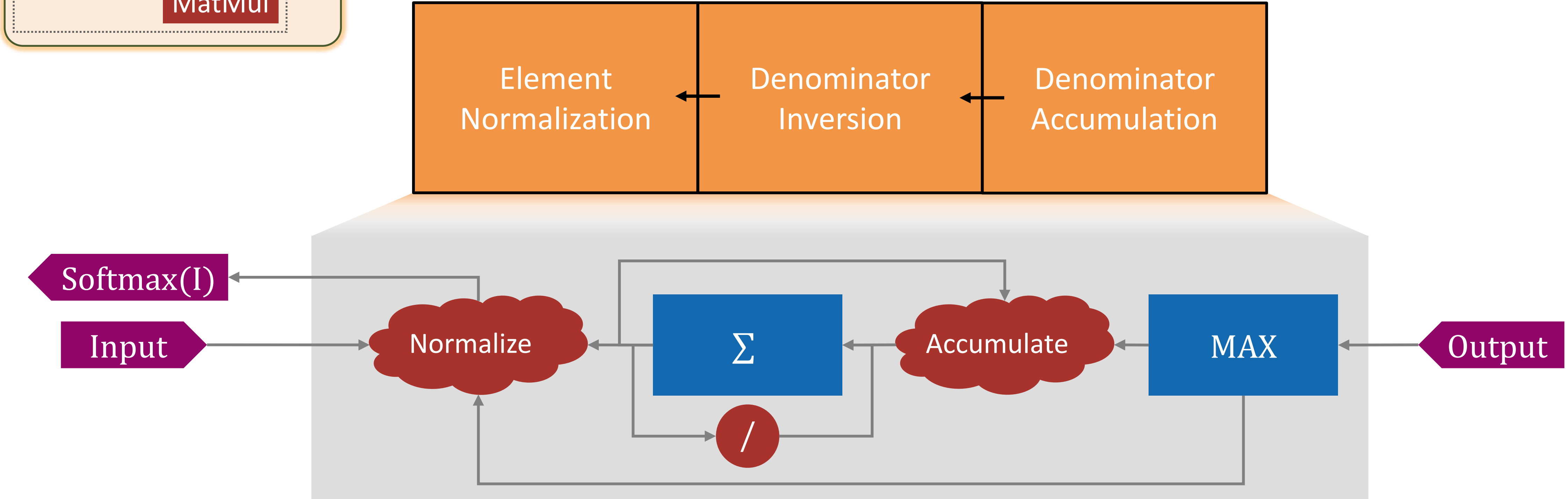
$$\mathrm{Softmax}(\mathbf{x})_i = \frac{1}{\sum_j^n 2^{(x_{qj}-\max(\mathbf{x}_q))\gg 5}} 2^{(x_{qi}-\max(\mathbf{x}_q))\gg 5}$$



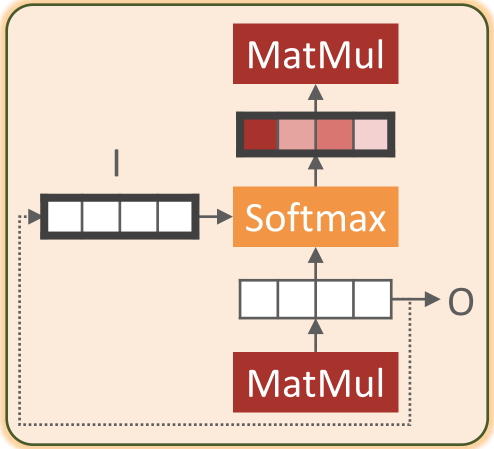
Hardware-friendly *Softmax*



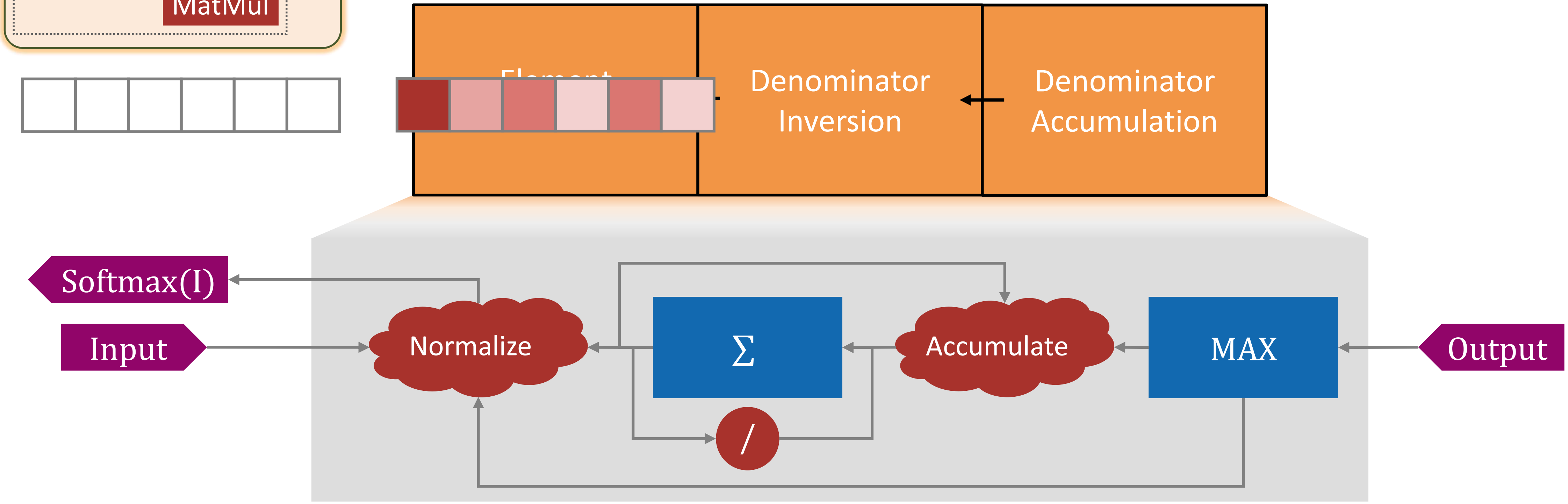
$$\mathrm{Softmax}(\mathbf{x})_i = \frac{1}{\sum_j^n 2^{(x_{qj}-\max(\mathbf{x}_q))\gg 5}} 2^{(x_{qi}-\max(\mathbf{x}_q))\gg 5}$$



Hardware-friendly *Softmax*

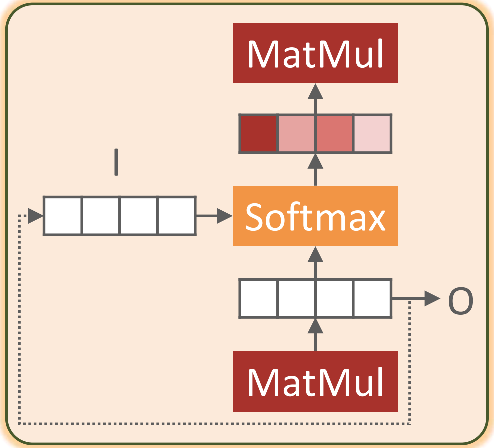


$$\mathrm{Softmax}(\mathbf{x})_i = \frac{1}{\sum_j^n 2^{(x_{qj} - \max(\mathbf{x}_q)) \gg 5}} \, 2^{(x_{qi} - \max(\mathbf{x}_q)) \gg 5}$$

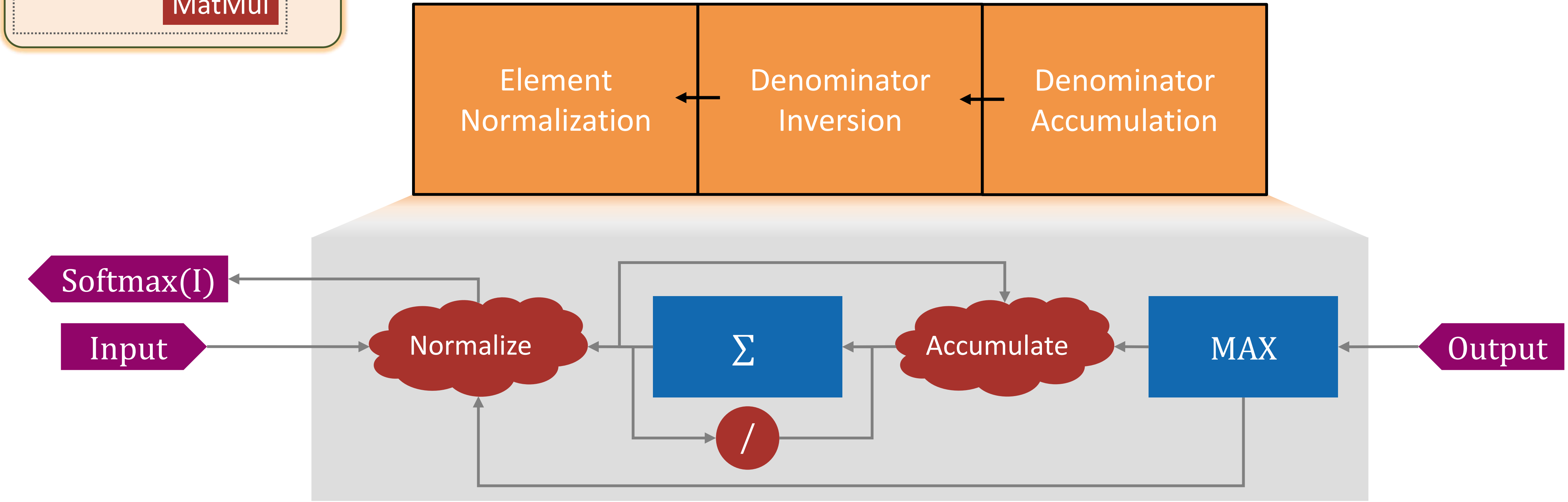


Hardware-friendly ***Softmax***

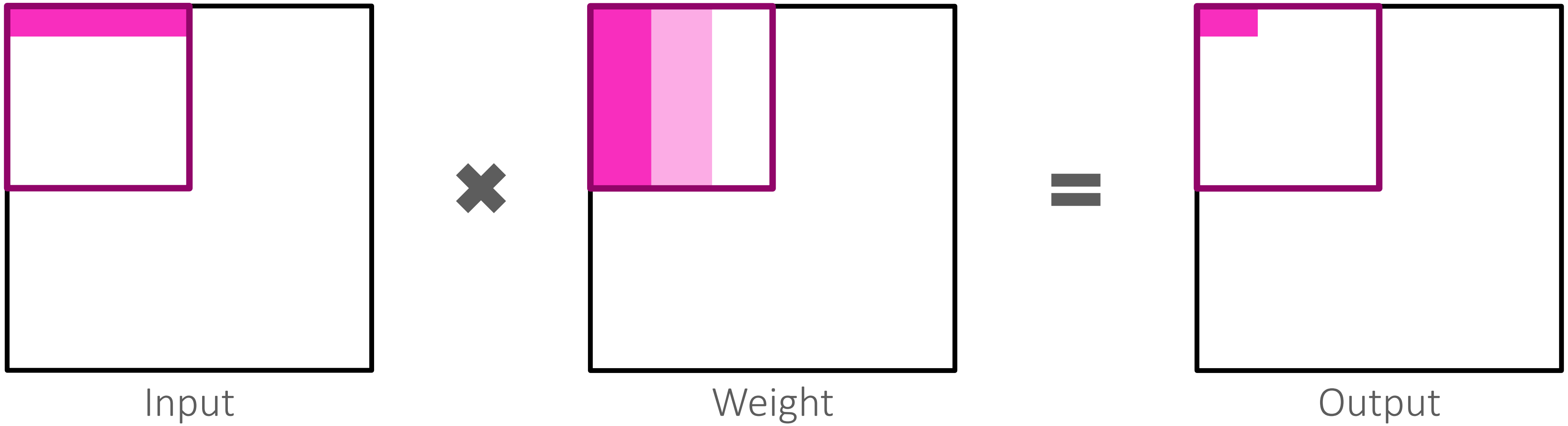
$MAE = \mathbf{0.46\%}$



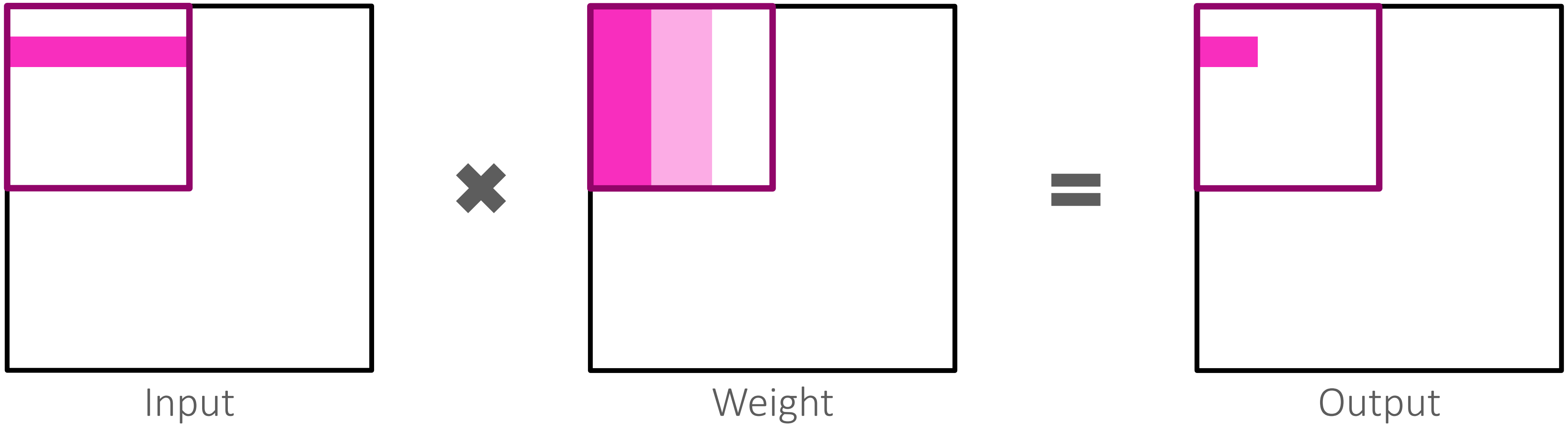
$$\text{Softmax}(\mathbf{x})_i = \frac{1}{\sum_j^n 2^{(x_{qj} - \max(\mathbf{x}_q)) \gg 5}} 2^{(x_{qi} - \max(\mathbf{x}_q)) \gg 5}$$



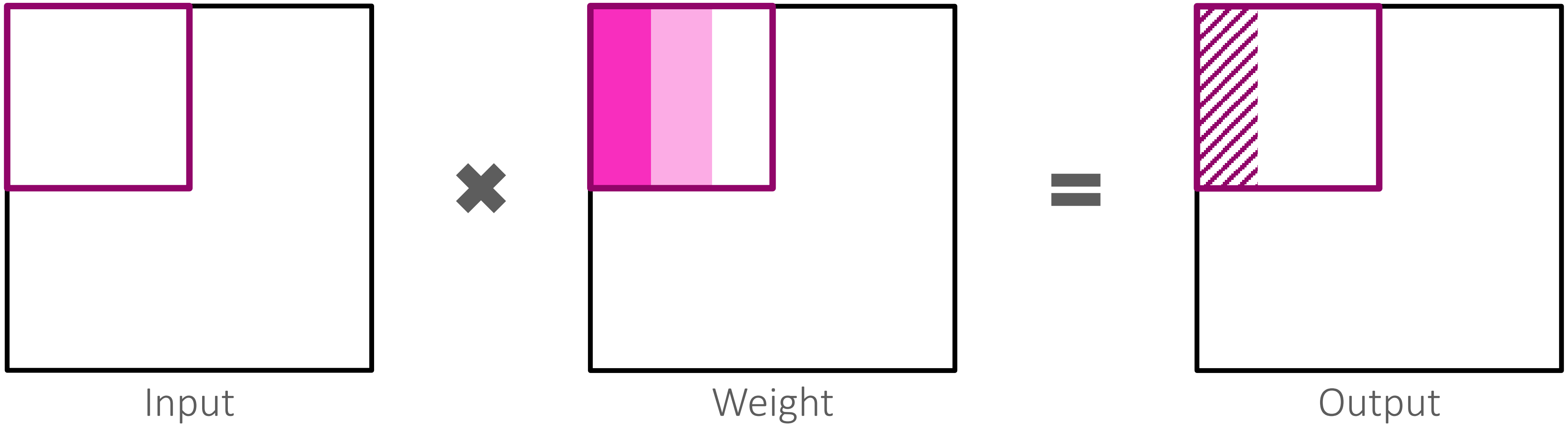
Output stationary - Local weight stationary



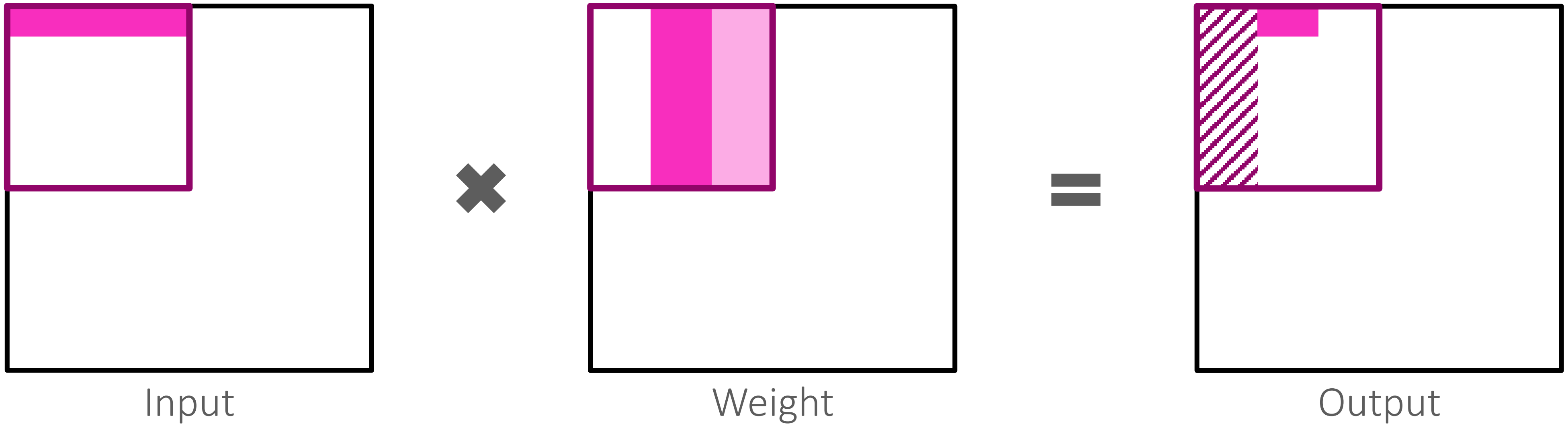
Output stationary - Local weight stationary



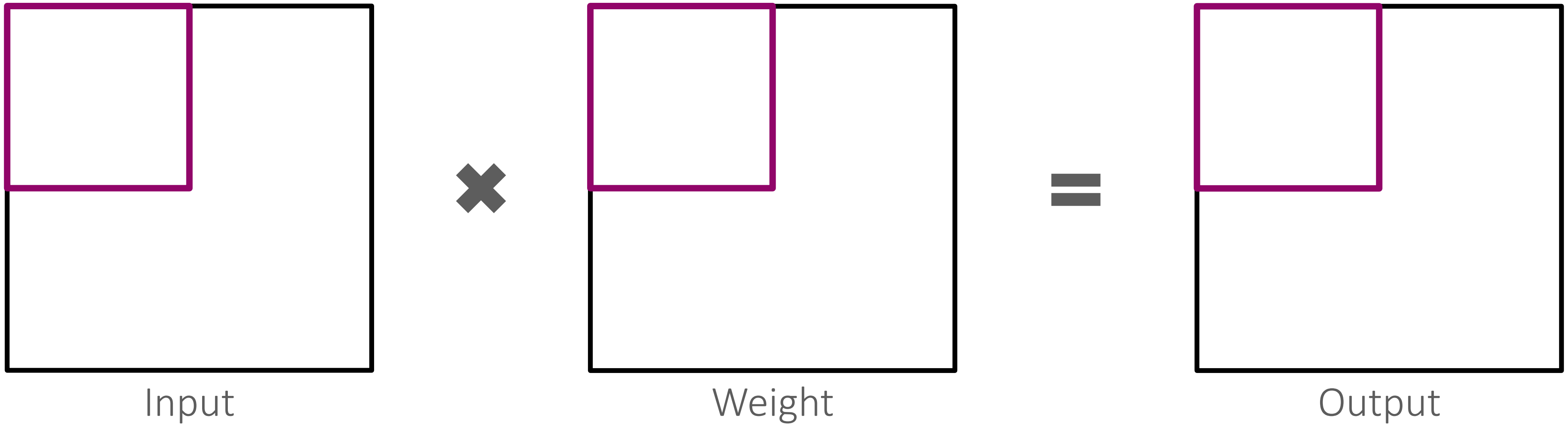
Output stationary - Local weight stationary



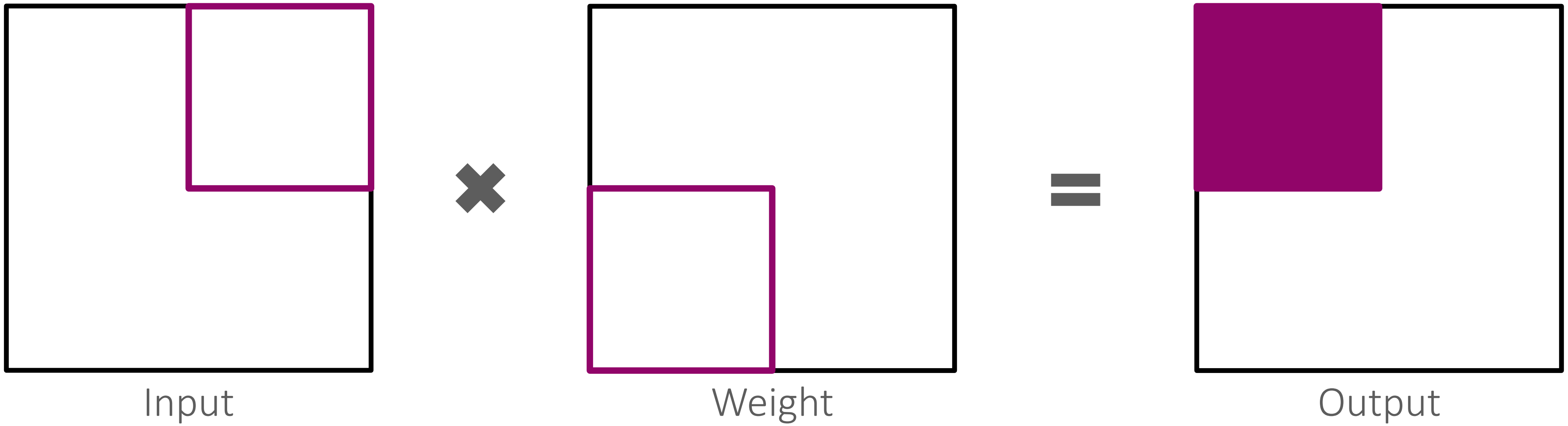
Output stationary - Local weight stationary



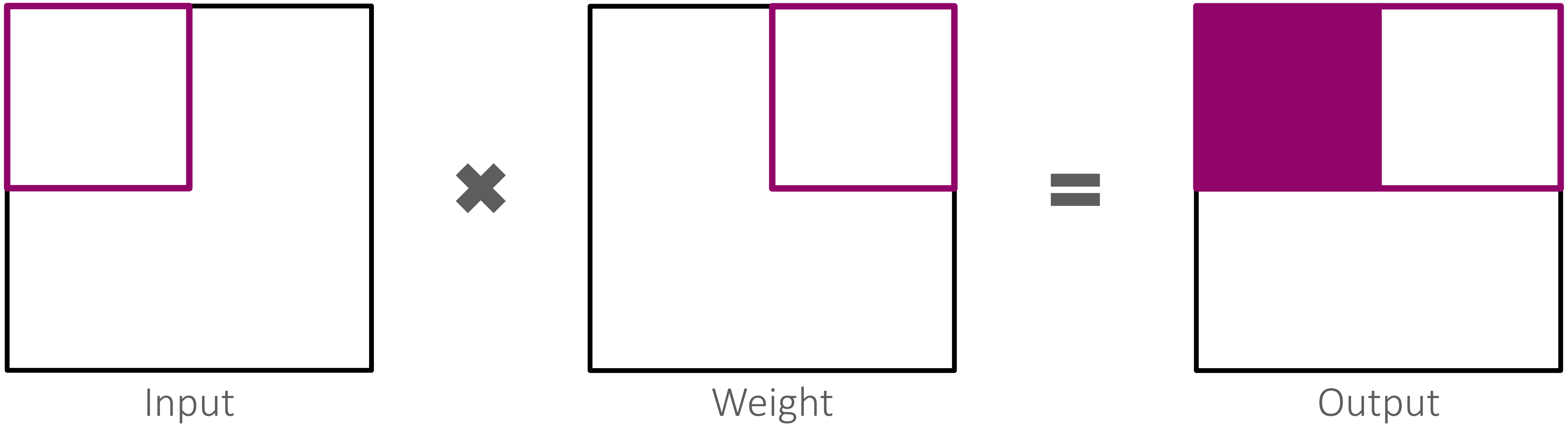
Output stationary - Local weight stationary



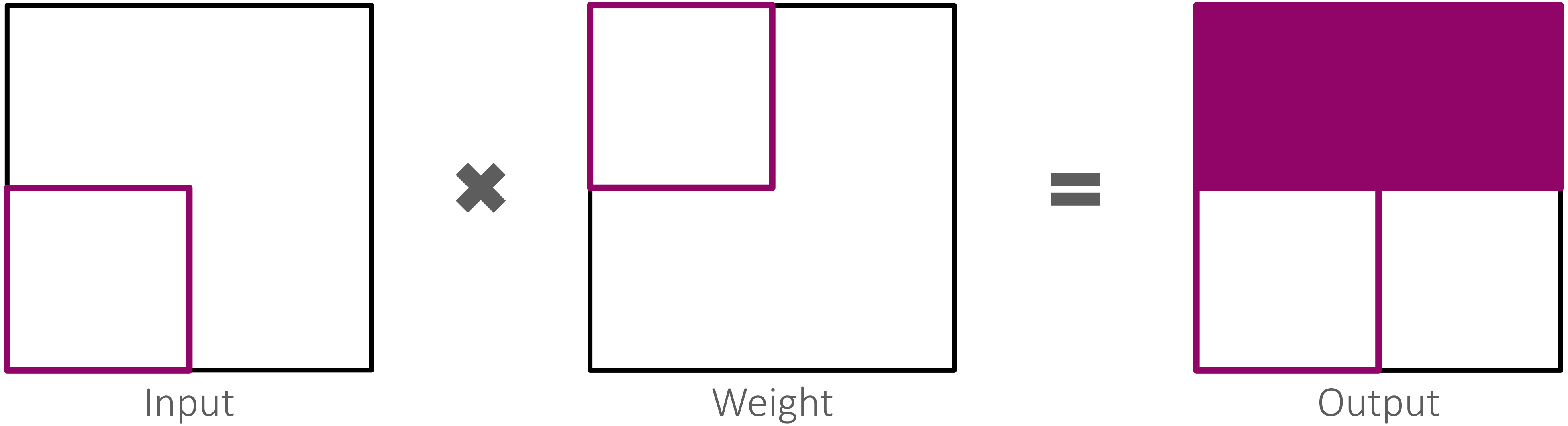
Output stationary - Local weight stationary



Output stationary - Local weight stationary

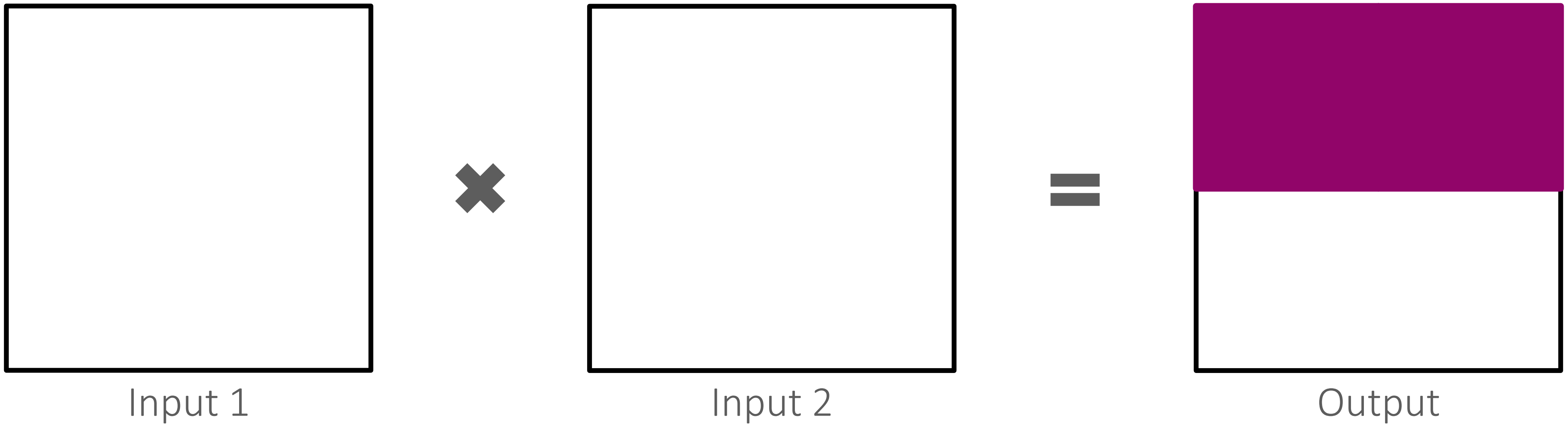


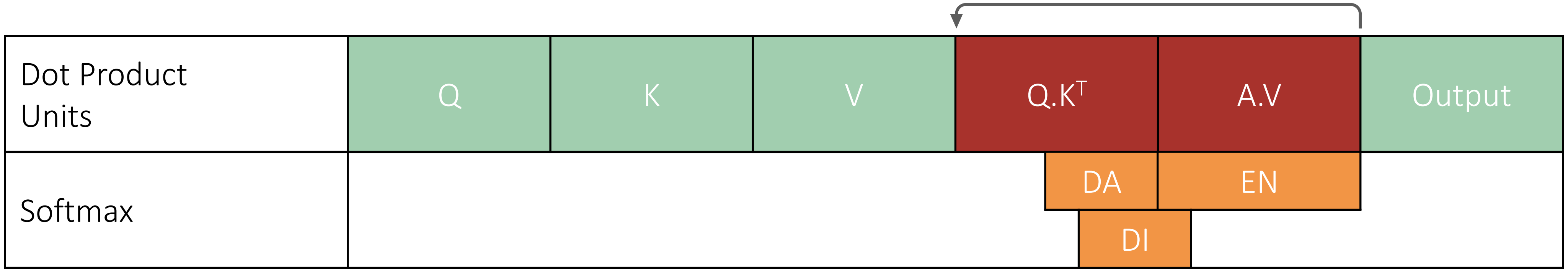
Output stationary - Local weight stationary



Dot Product Units	Q	K	V			Output

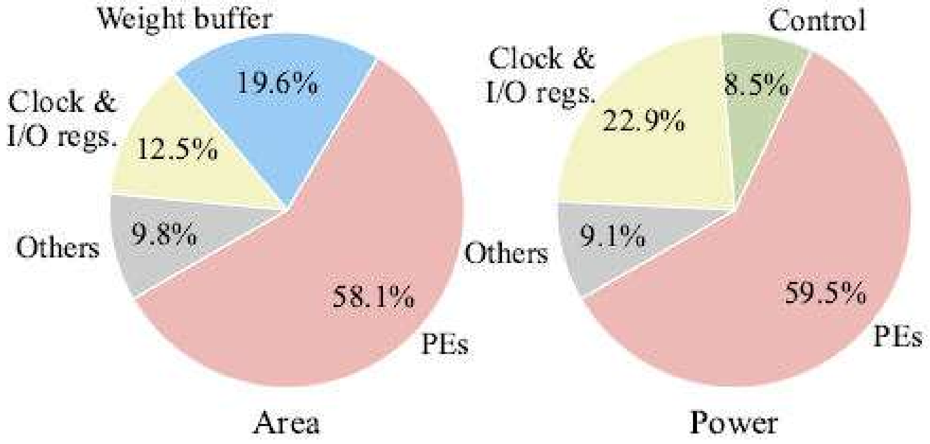
Fused Q.K^T and A.V computation

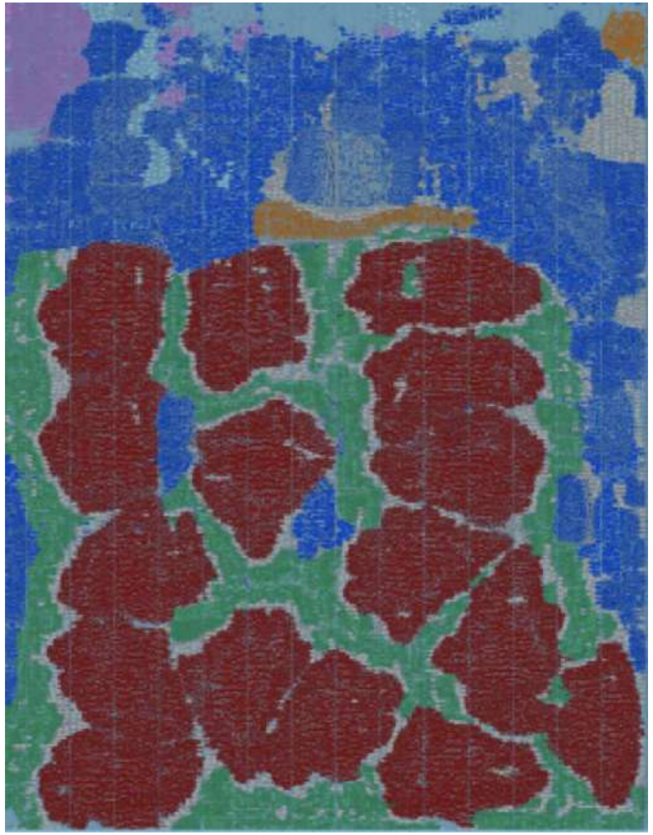




Physical Implementation

- Implemented in GF22FDX
  - Target frequency of 500 MHz (SS/0.72V/125°C)
- Area 0.17 mm^2
  - *Softmax* module has only **3.3%** area contribution, corresponding to 28.7 KGE.
- Power 60 mW (TT/0.80V/25°C)
  - *Softmax* module consumes **1.4%** of the power.

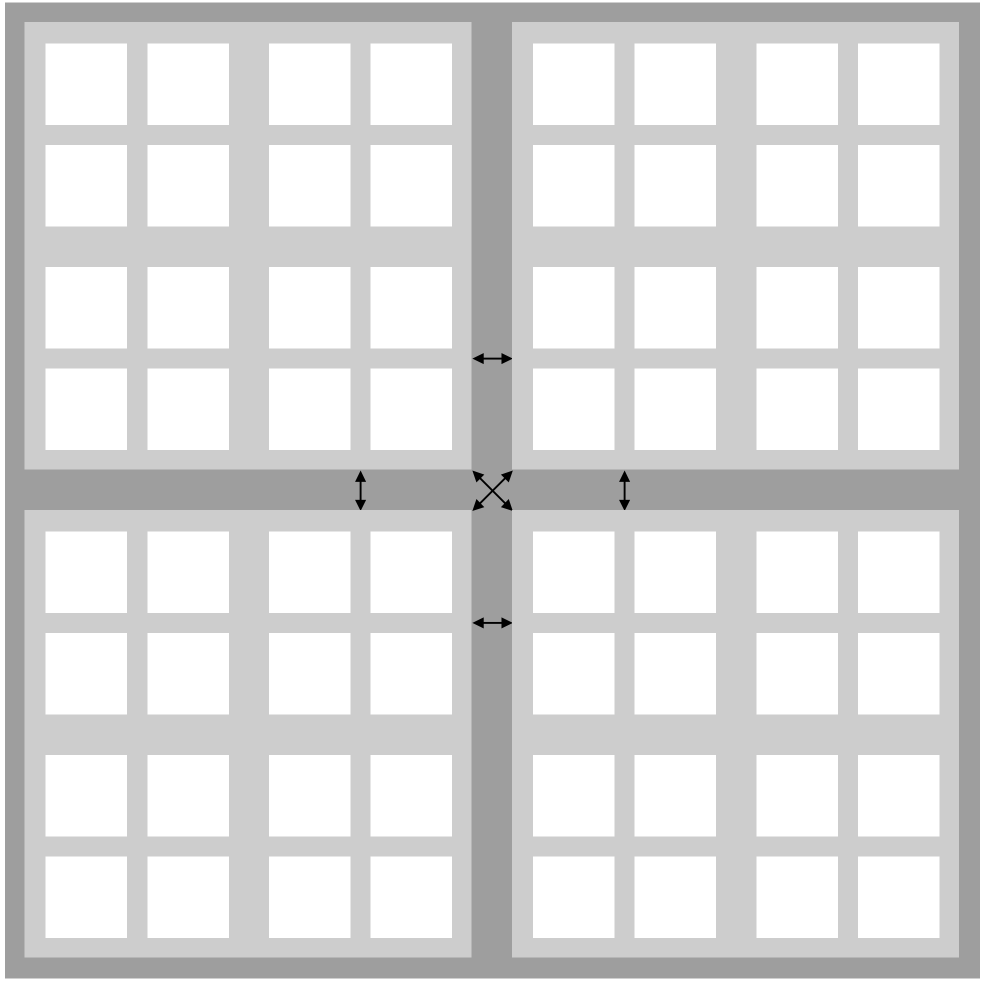




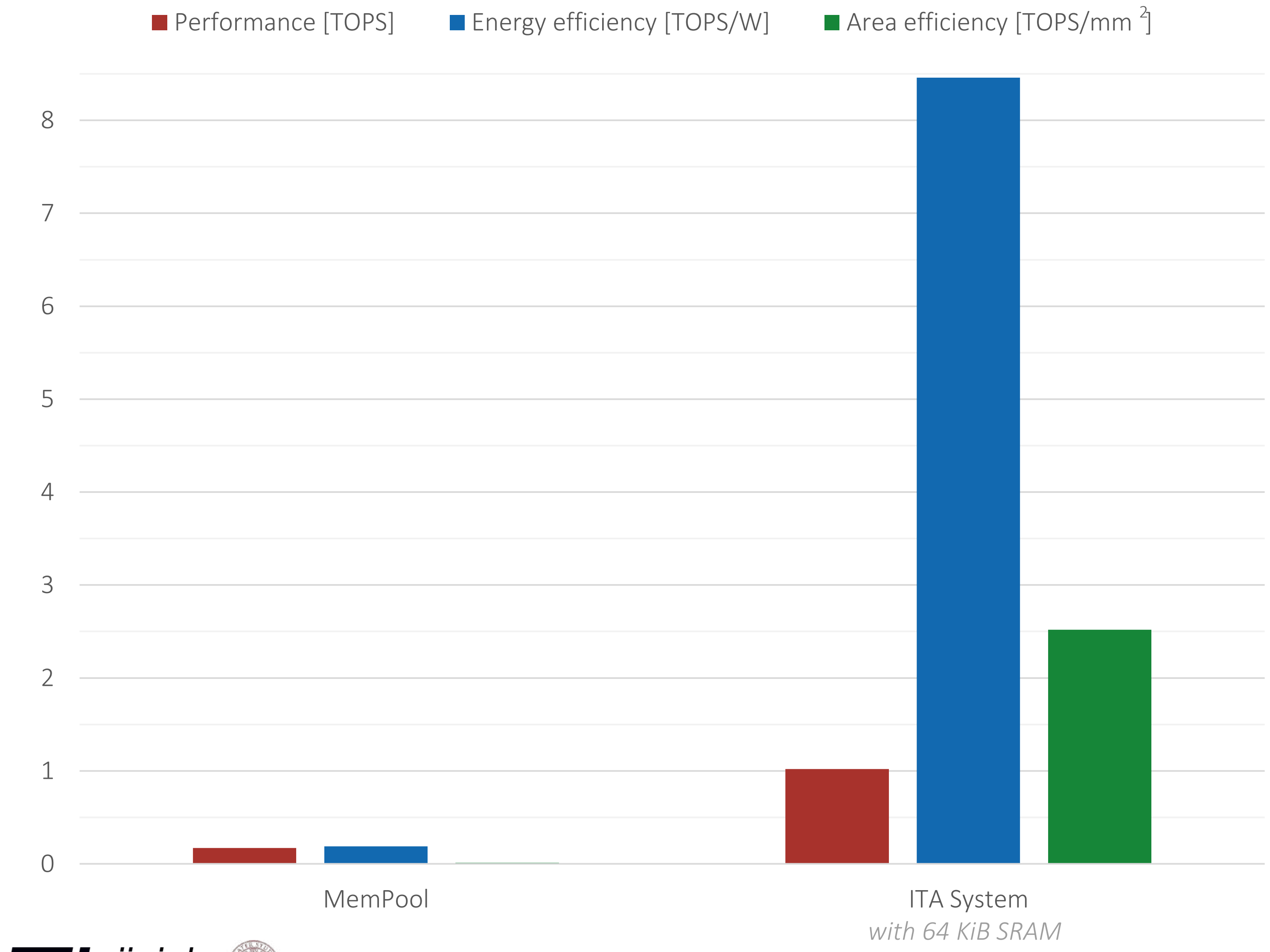
🟥 Dot product units
🟩 Weight memory
🟧 Softmax

Comparison to a software baseline on **MemPool**

- Many-core system with low-latency L1 memory
- Designed for highly parallel workloads
- 256 32-bit RISC-V cores
- 1 MiB L1 scratchpad memory



Comparison to a software baseline on **MemPool**



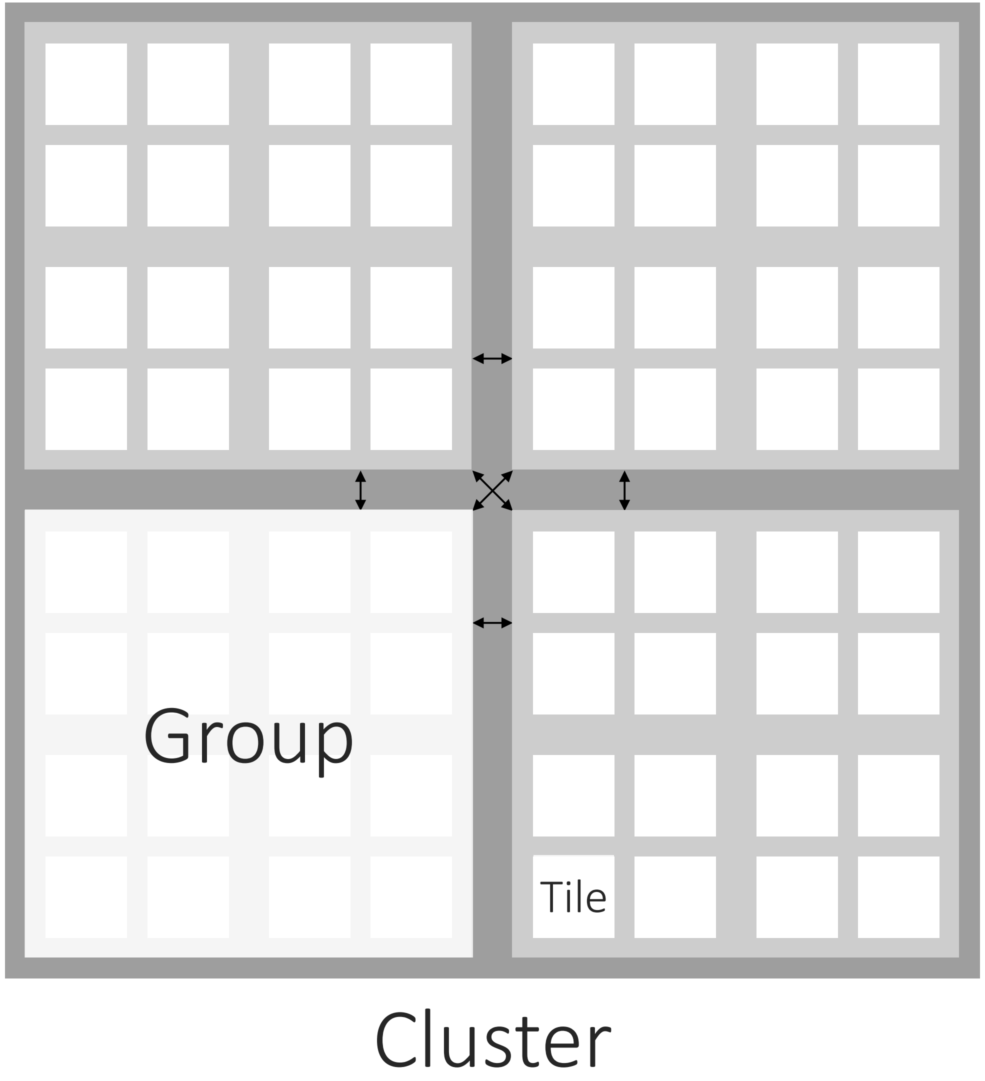
Performance
increase of **6x**

Energy Efficiency
increase of **45x**

Area Efficiency
increase of **220x**

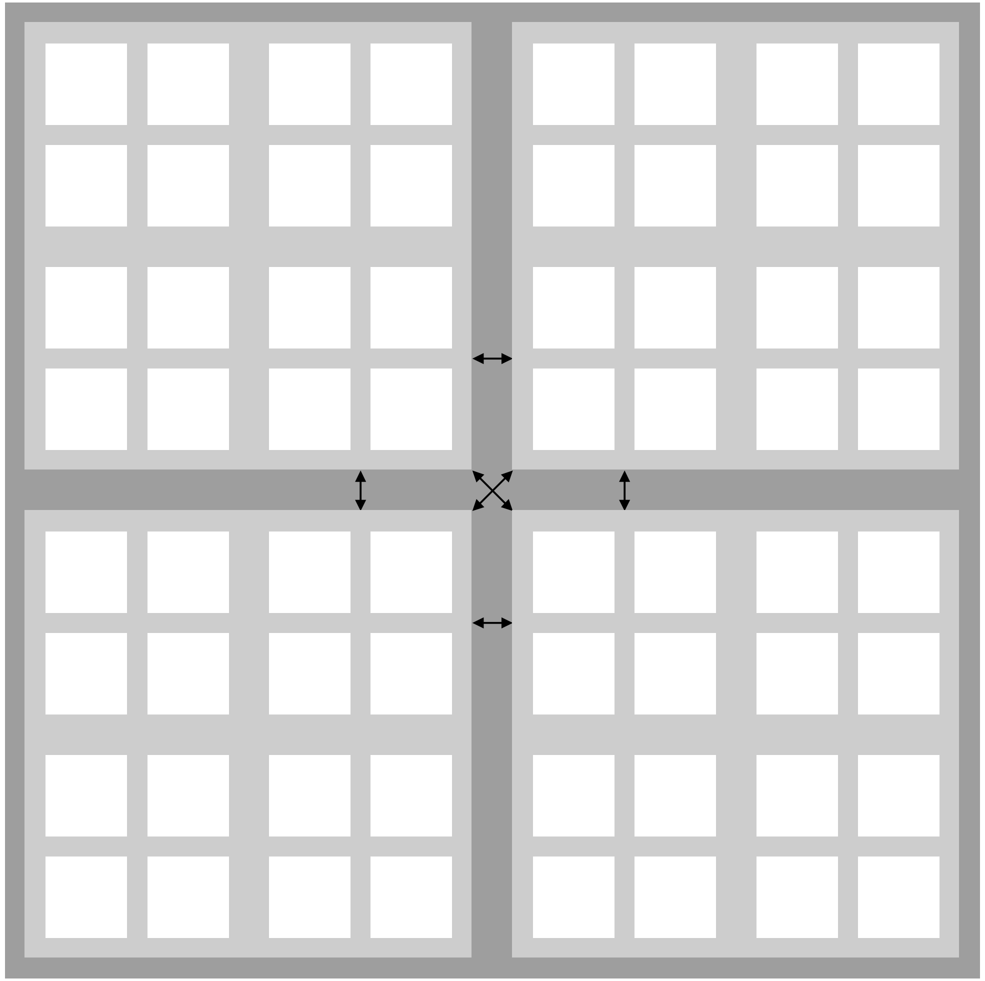
Integrating ITA into MemPool

- Not very straightforward!
- Hierarchical architecture
  - Cluster
  - Group
  - Tile



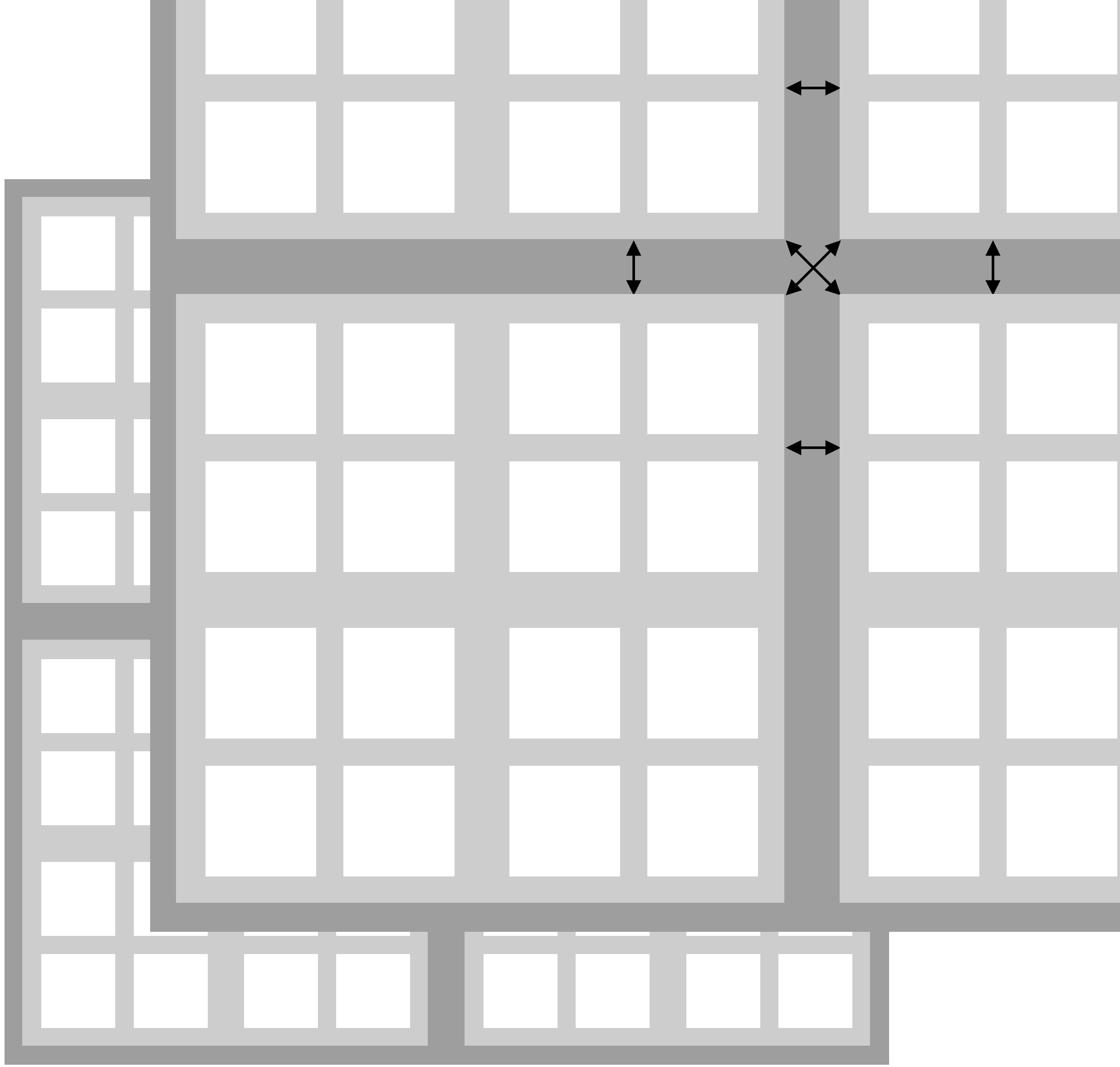
Integrating ITA into MemPool

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



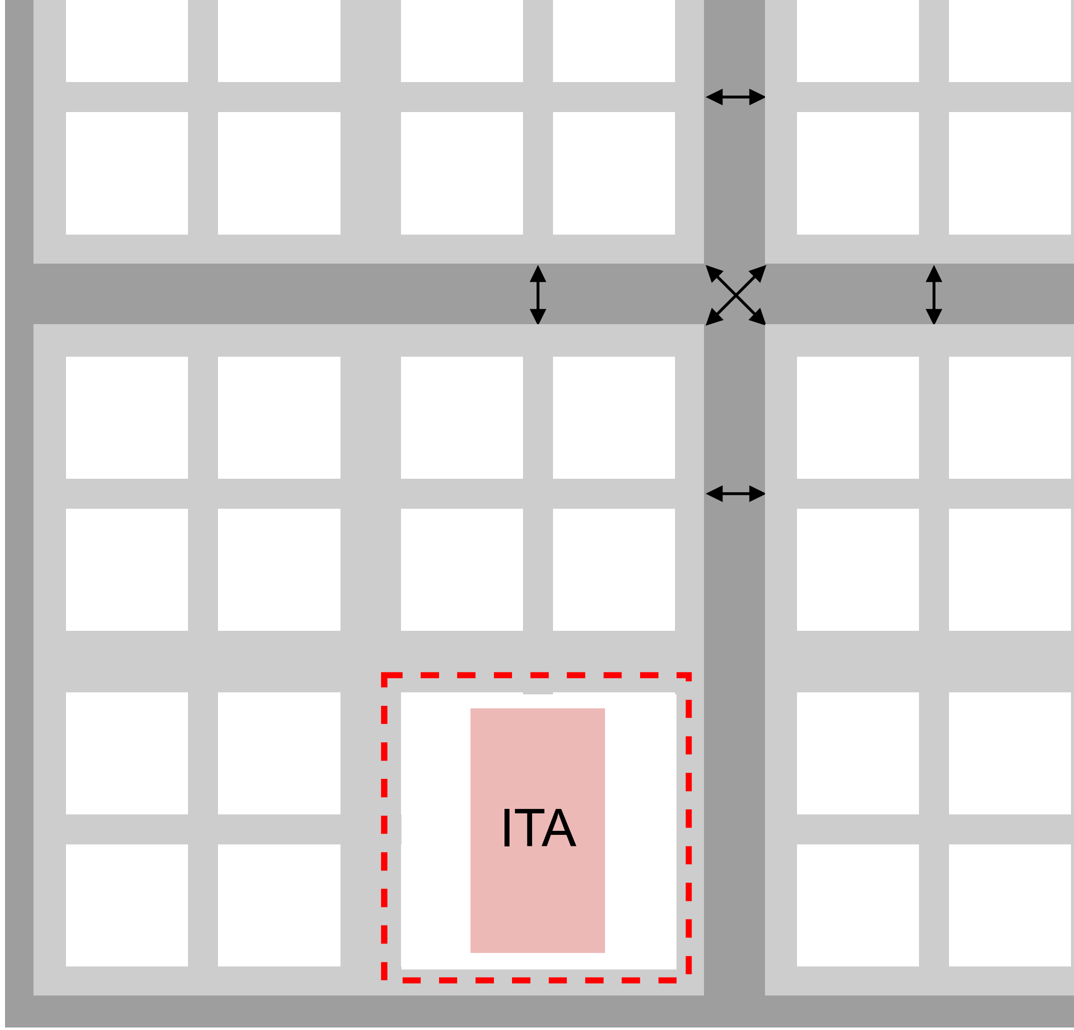
One ITA core per Group

- ITA fits **four tiles** of MemPool



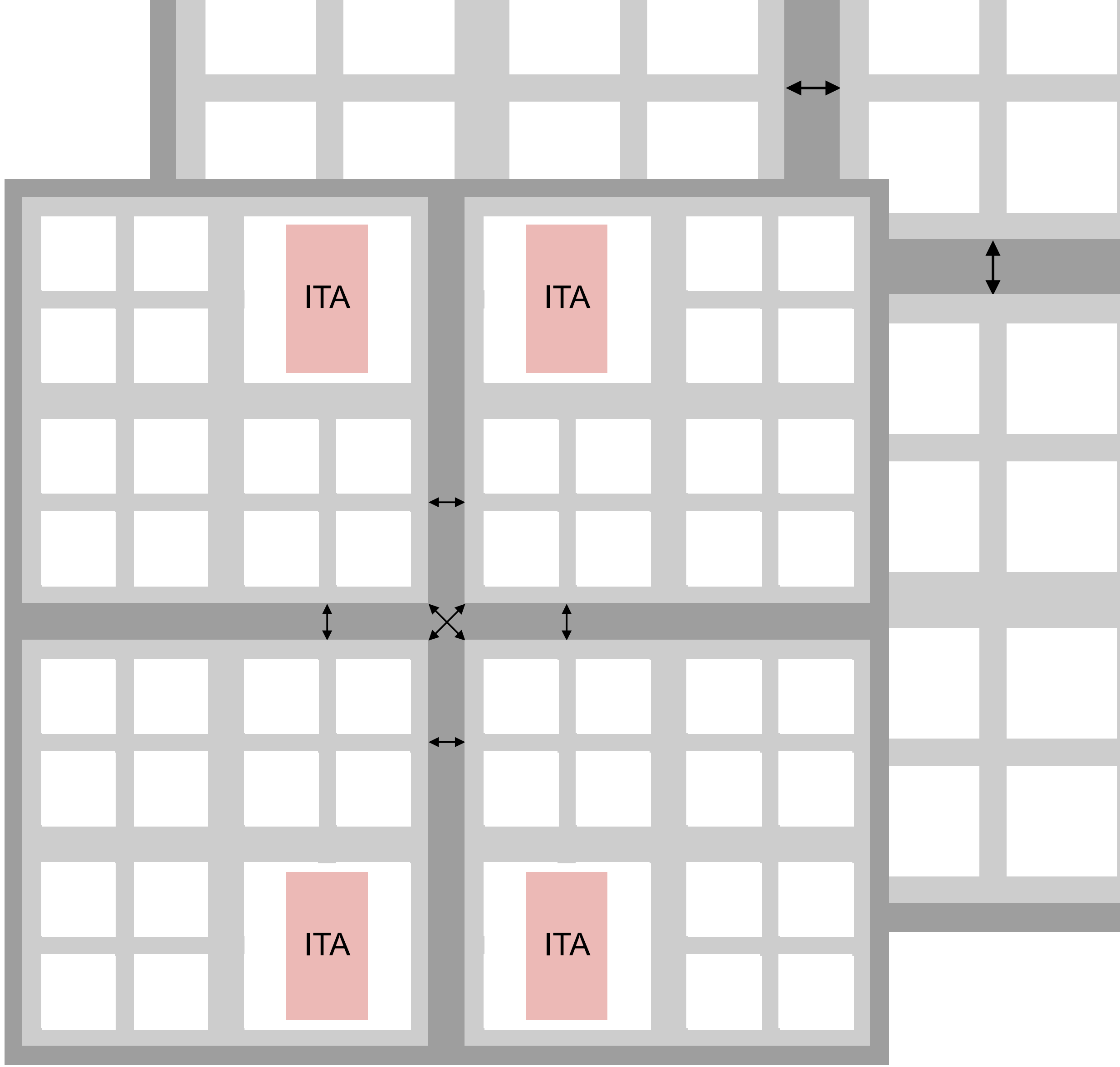
One ITA core per Group

- ITA fits **four tiles** of MemPool
- Bottom right
- Remove the cores
- Rearrange the banks



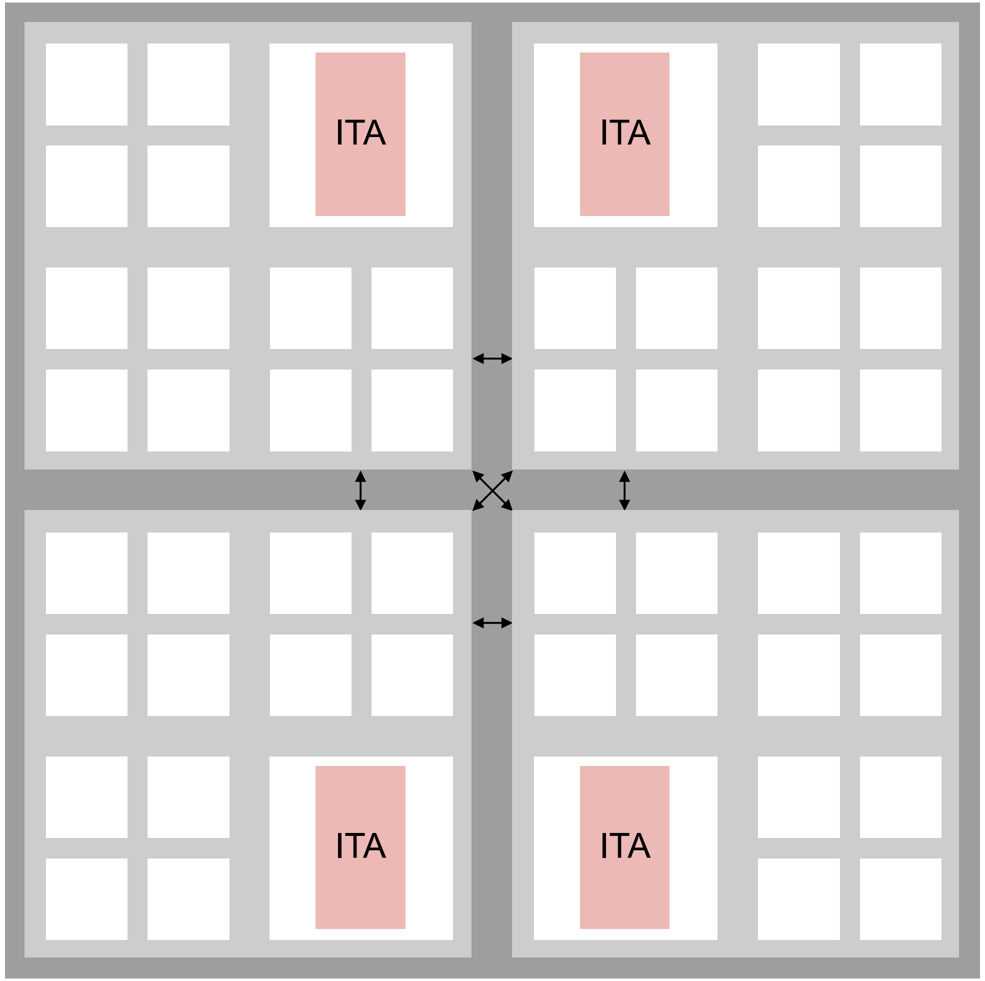
One ITA core per Group

- ITA fits **four tiles** of MemPool
- Bottom right
- Remove the cores
- Rearrange the banks
- Each ITA core works on one head of attention



Integrating ITA into MemPool

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



Modified Interconnect of Four Tiles

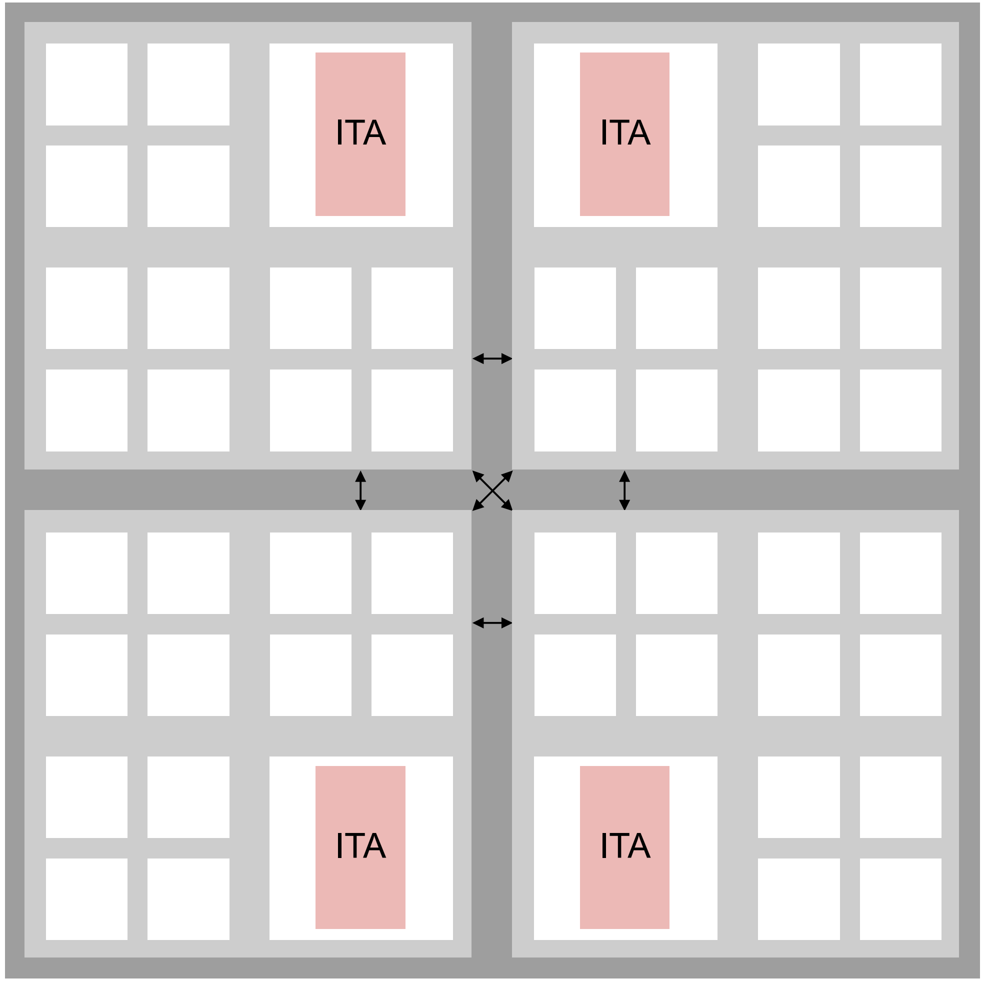
- 4 tiles = 64 banks
- ITA needs to access 28 banks per cycle
- 3 types of requests/responses
  - Core
  - ITA
  - DMA

ITA > DMA > Core

ITA

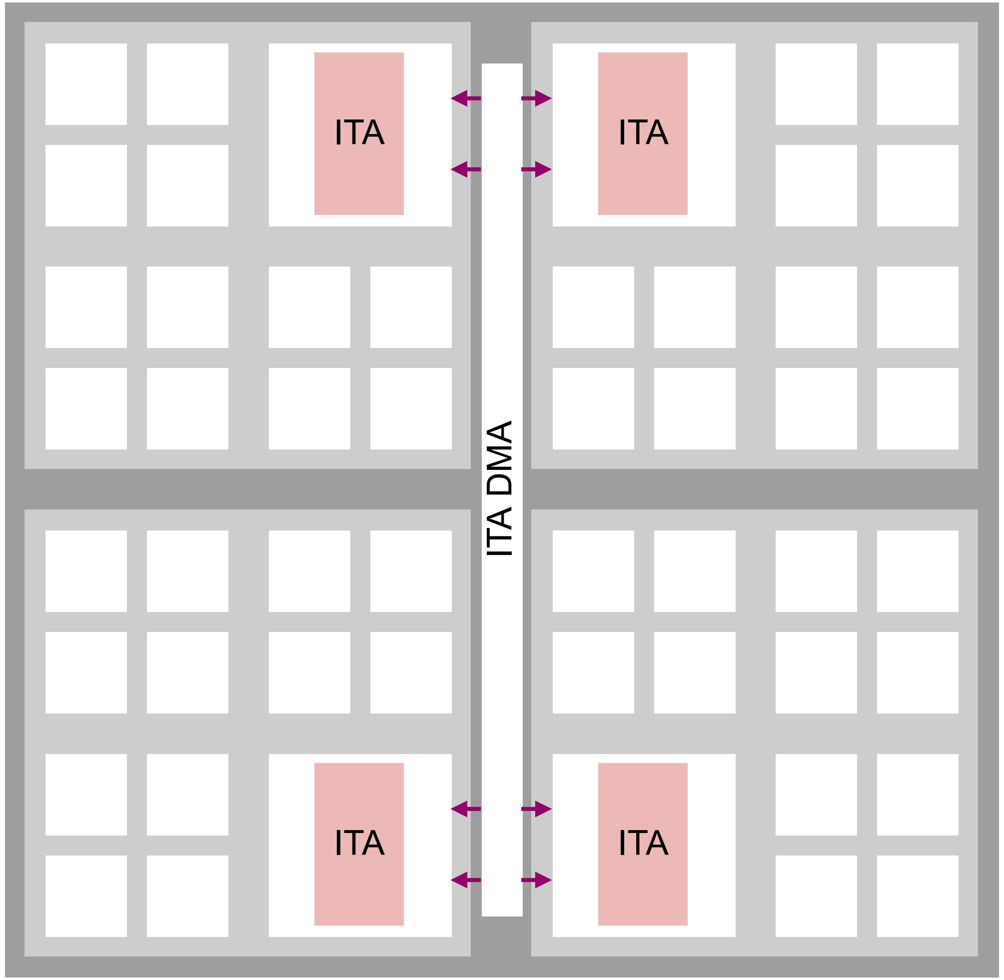
Integrating ITA into MemPool

- ✓ 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

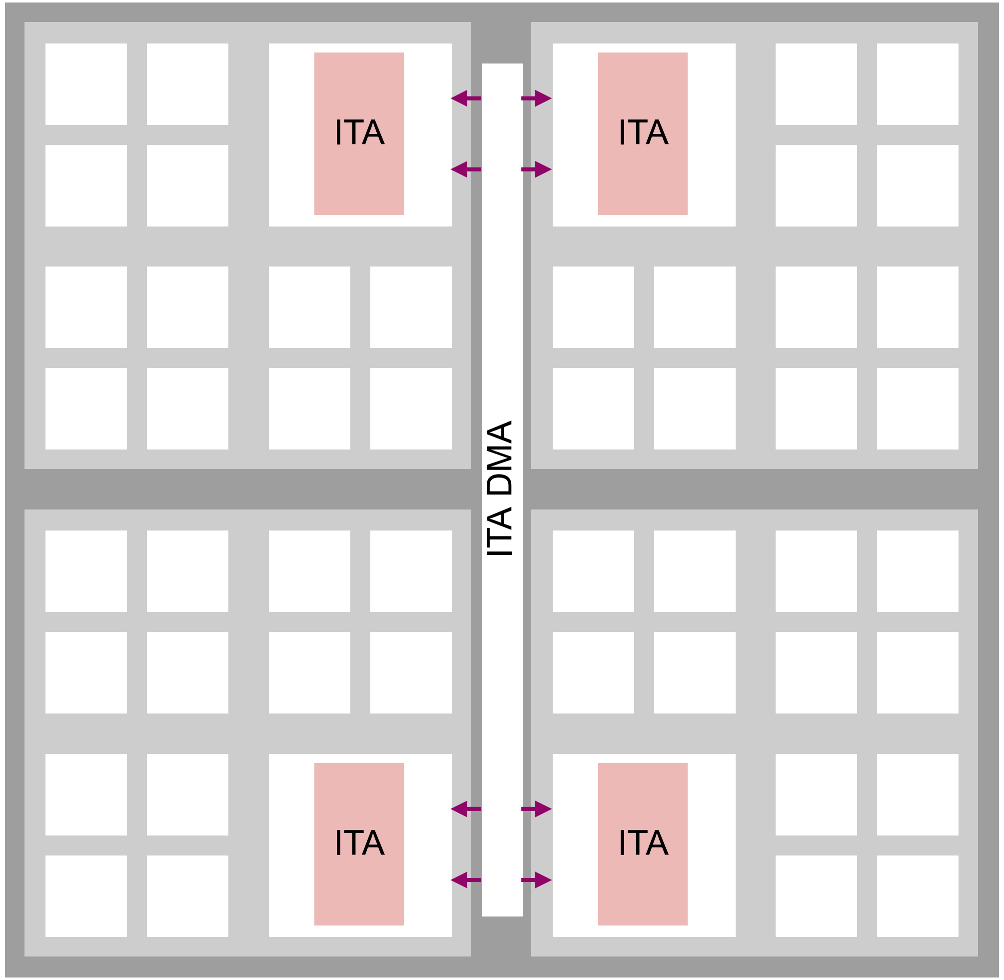
- Moves transformer data from L2 to L1 memory
- Inputs are broadcasted to all groups
- Two 16 bytes/cycle ports per group



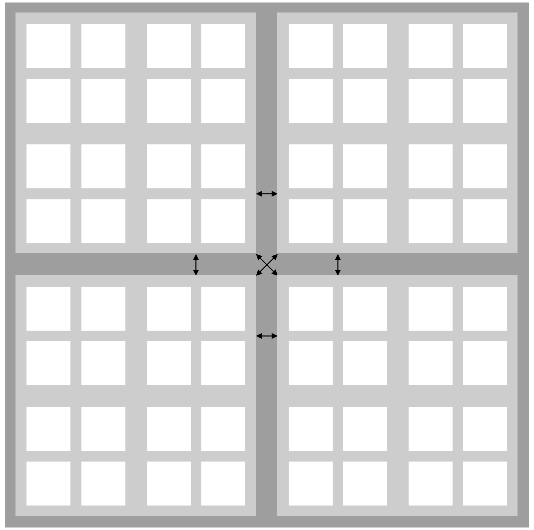
Integrating ITA into MemPool

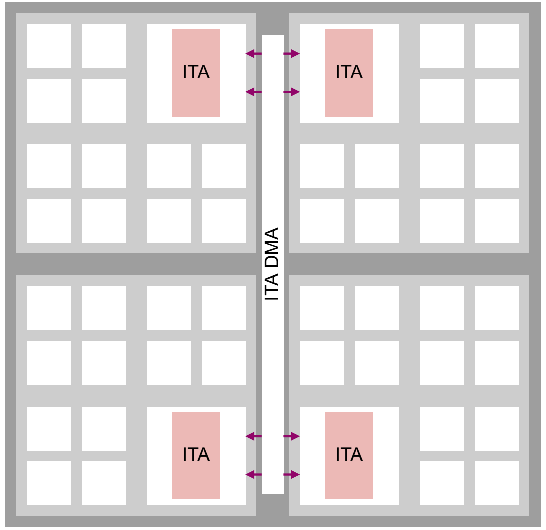
- ✓ Where to put ITA?
- ✓ How to connect ITA to L1 memory?
- ✓ How to refill L1 from L2 memory for ITA?

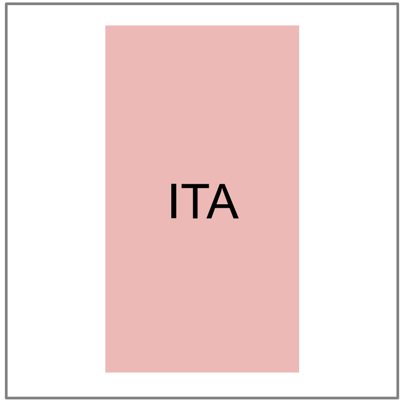
End-to-end heterogeneous collaborative Transformer deployment



Comparison to MemPool and ITA System

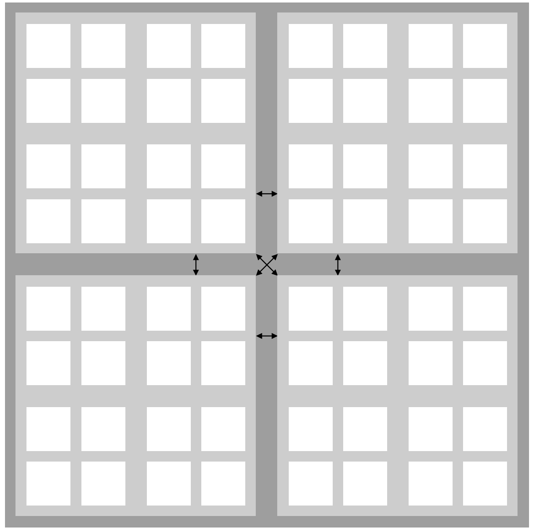


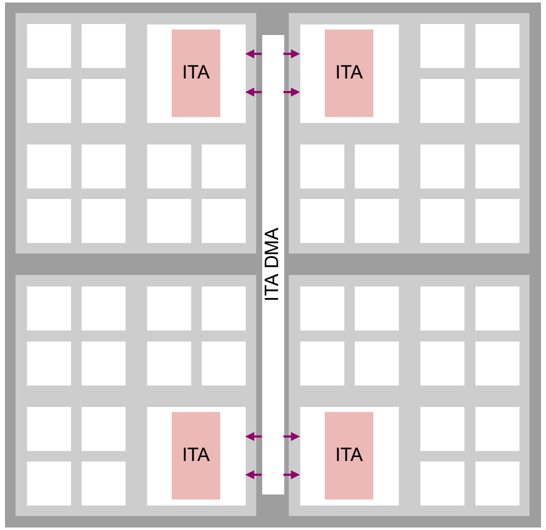


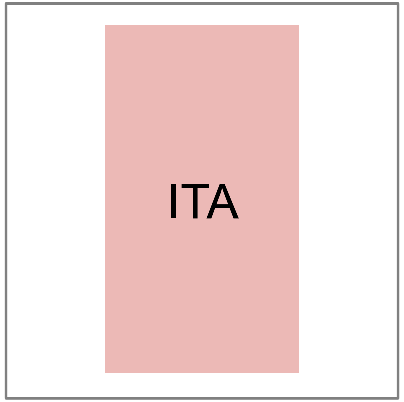


Throughput [TOPS]
Energy efficiency [TOPS/W]
Area efficiency [TOPS/mm^2]

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/mm²]	0.0114	2.10	**5.02**	2.52

25x

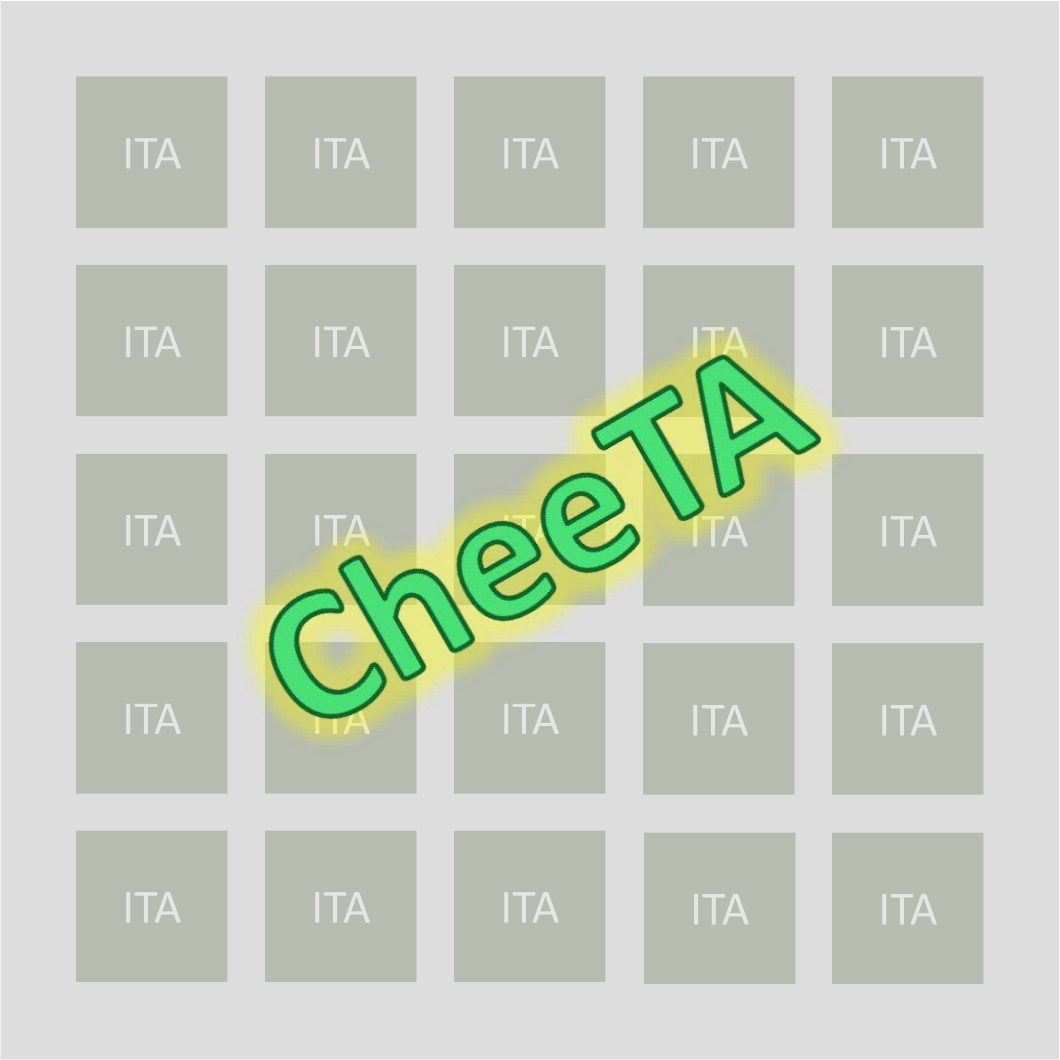
45x

2x

Future of ITA: Scaling up further

- Target workloads like GPT
- Floating-point capability

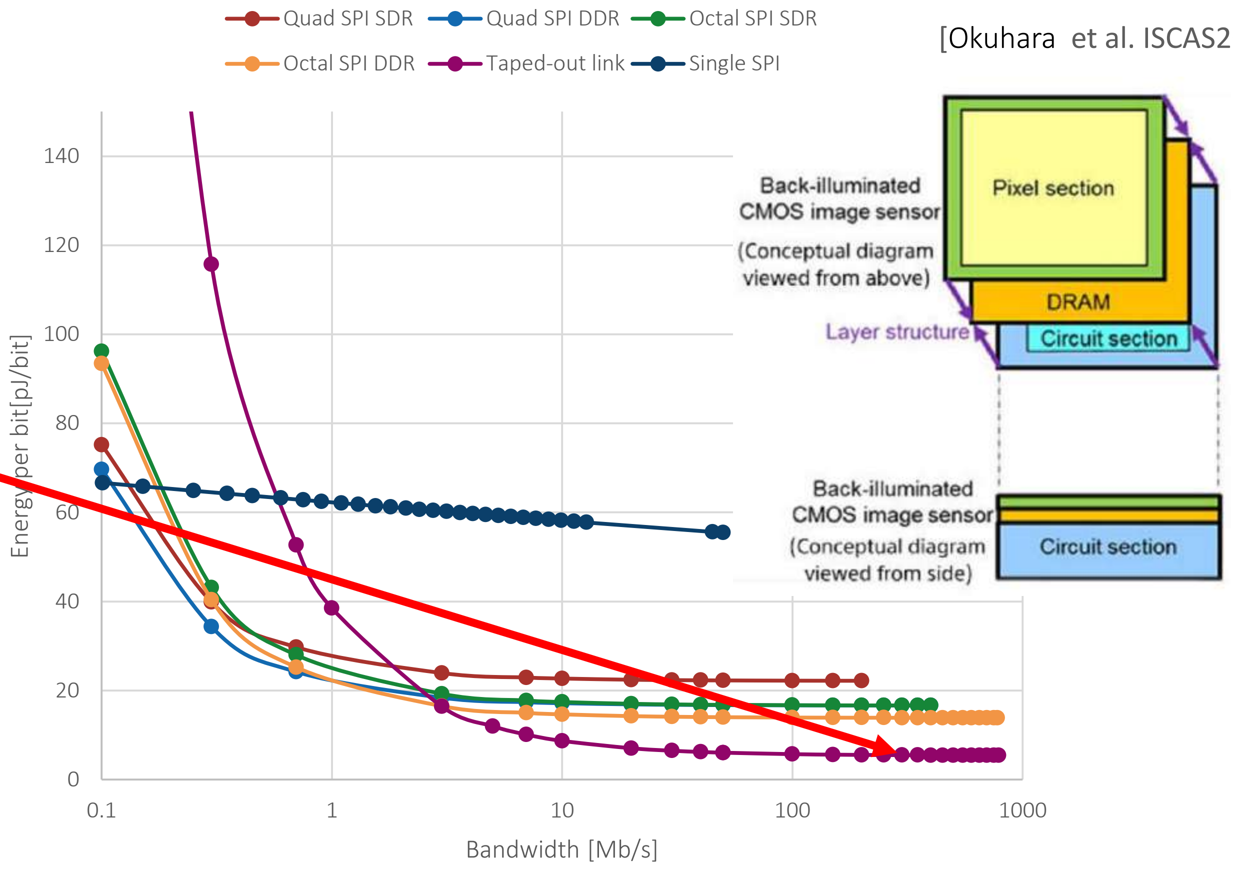
Accelerate **LLMs** and reach **100 TFLOPS or higher!**



Off-chip (Memory & Sensors): Feel the (IO) Pain!

No good solutions to curtail IO power for extreme edge ML

- SPIs
  - I/O VDD=1.8V
  - fspi-max=50MHz,
  - Assuming duty-cycled operation @ various bandwidths
- ULP serial link (duty-cycled)
  - 10.2x less energy and 15.7x higher maximum BW compared to single SPI
  - 2.56x higher efficiency than the DDR Octal SPI @787Mbps
  - 5 → 3pJ/bit
  - **However it's still 2mW@ 500Mbps**
- 3D integration: 0.15pJ/bit and below



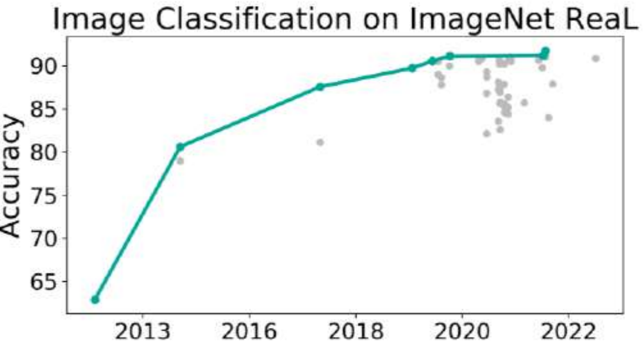
[Okuhara et al. ISCAS20]

2.5D and 3D coming fast even for extreme edge!

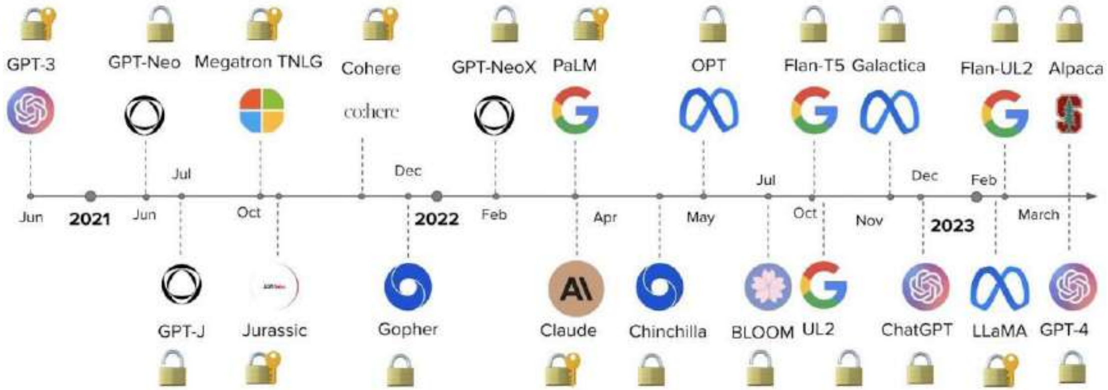
Closing thoughts

- Edge Computing requires flexibility, efficiency and *energy proportionality*
  - Multiple efficiency boosters (accelerators): ISA extensions, HWCEs: 1-4 OoM!
  - On-chip non-volatile memory, event-based processing for proportionality
- Moving forward – proliferation of acceleration engines
  - For tuning accelerator to sensor (like the brain)
  - For high-level sensor fusion, control, planning
- Managing inter-accelerator dataflows – low latency & high-bandwidth
  - Low-latency interconnects are key
  - Processors, Memory and interconnect fabrics need to be co-designed
  - Scaling to full-die and (heterogeneous) chiplets – next generation NoCs
- **Scale-up for foundation models**

Perceptive → Generative → Embodied AI

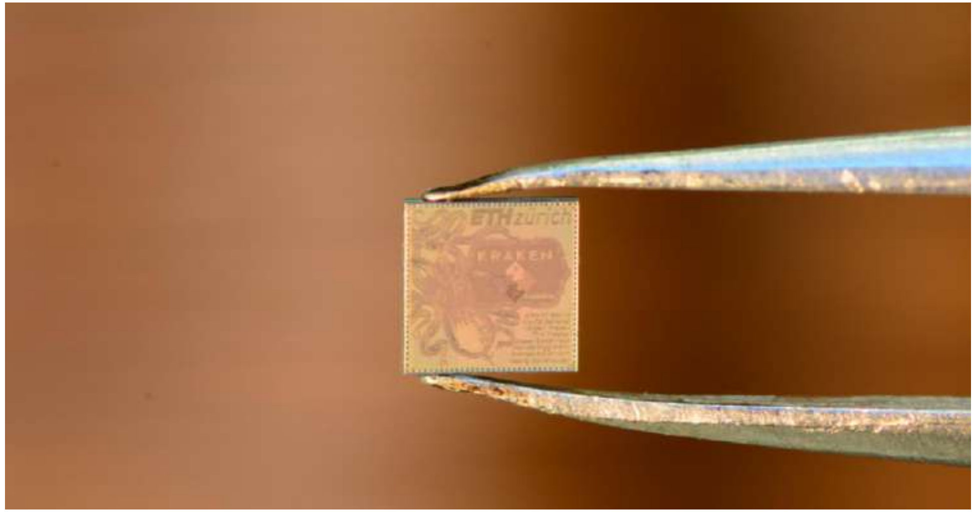


Precise



Interactive, creative

Efficient, RT-safe, secure



[8] pulp-platform/sne (github.com)

[9] pulp-platform/CUTIE (github.com)

[10] Kraken: A Direct Event/Frame-Based Multi-sensor Fusion SoC for Ultra-Efficient Visual Processing in Nano-UAVs
https://www.research-collection.ethz.ch/handle/20.500.11850/565105

Thanks!