

**PULP**  
Parallel Ultra Low Power

# Extreme Edge AI on Open Hardware

*ACCML@HIPEAC Bologna*

*20.01.2020*

*Luca Benini<sup>1,2</sup>*



*<sup>1</sup>Department of Electrical, Electronic  
and Information Engineering*

**ETH zürich**

*<sup>2</sup>Integrated Systems Laboratory*



European Research Council



European  
Commission

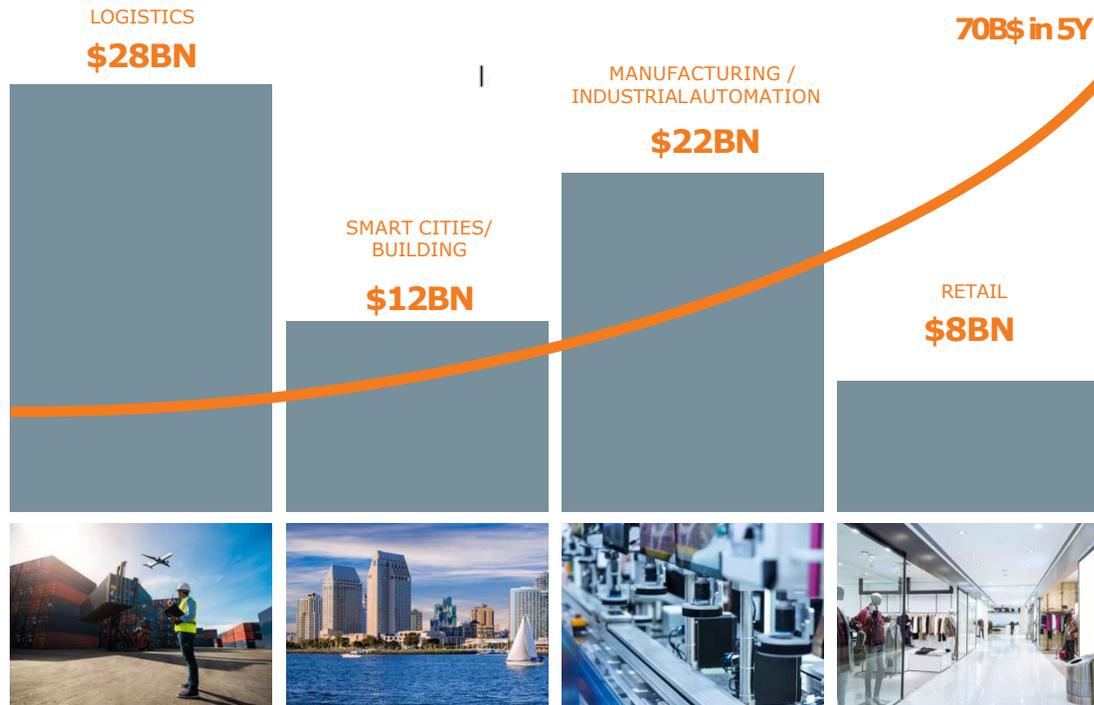
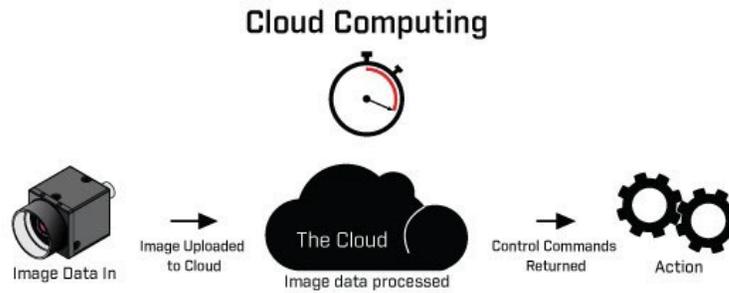
Horizon 2020  
European Union funding  
for Research & Innovation



FN-NF

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

# Cloud → Edge → Extreme Edge AI aka TinyML



70\$ in 5Y AVERAGE CAGR 273%

#1 Customer Question on Amazon.com (out of 1,000+):

1. I don't want any of my (private, personal) videos on any servers not in my control. Is this possible?

#2 Customer Question on Amazon.com (out of 1,000+):

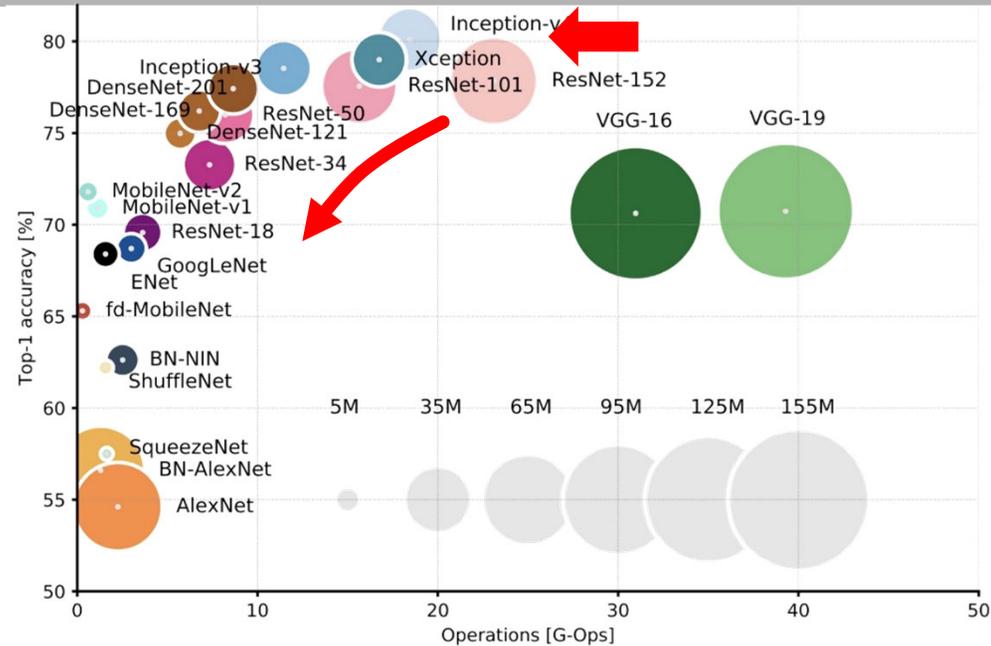
2. How long does the battery charge last?

Source: [www.amazon.com/ask/questions/asin/B01M3VHG87/](http://www.amazon.com/ask/questions/asin/B01M3VHG87/)

**Extreme edge AI challenge**  
**AI capabilities in the power envelope of an MCU: 100mW peak (1mW avg)**

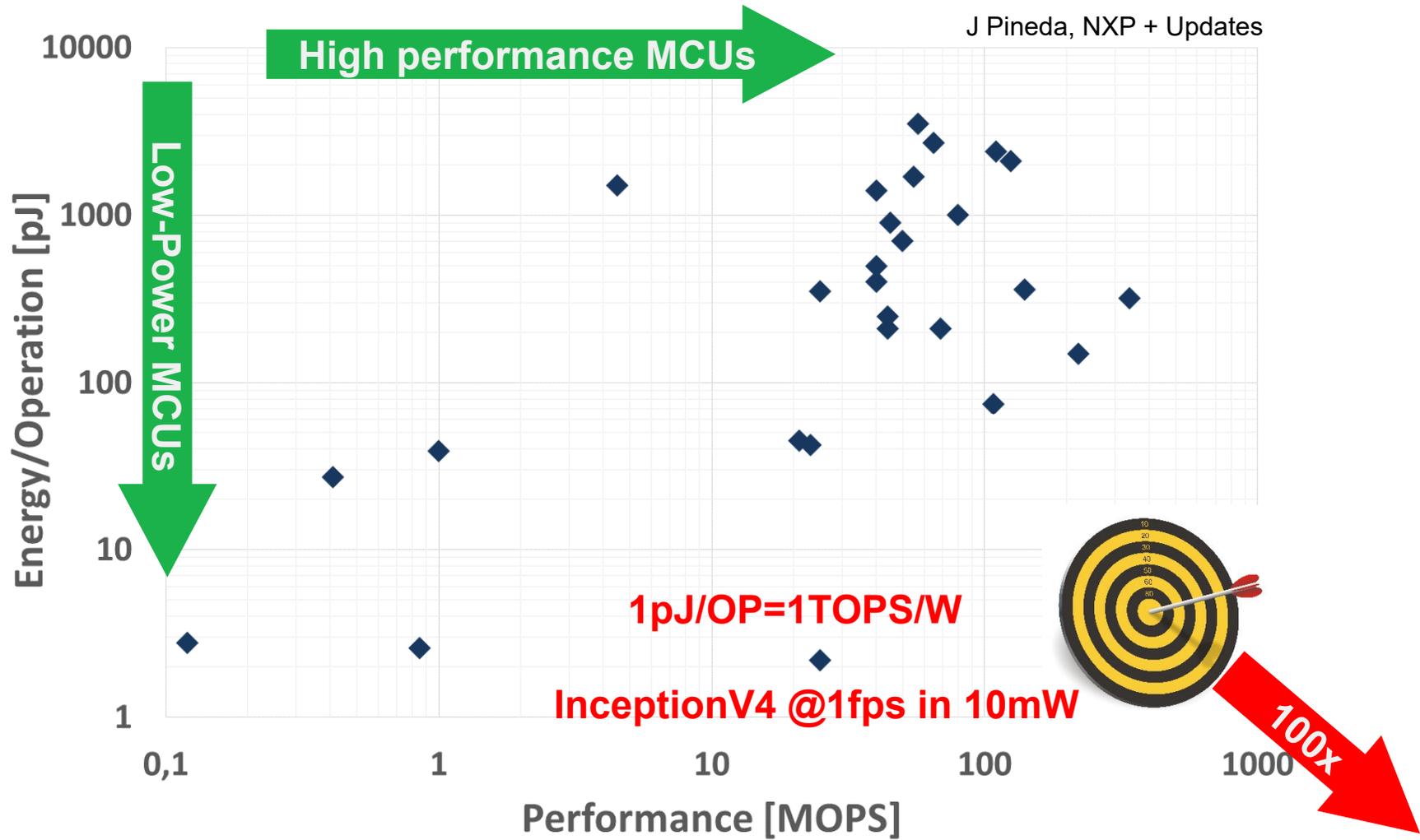
# AI Workloads from Cloud to Edge (Extreme?)

**GOP+**  
**MB+**



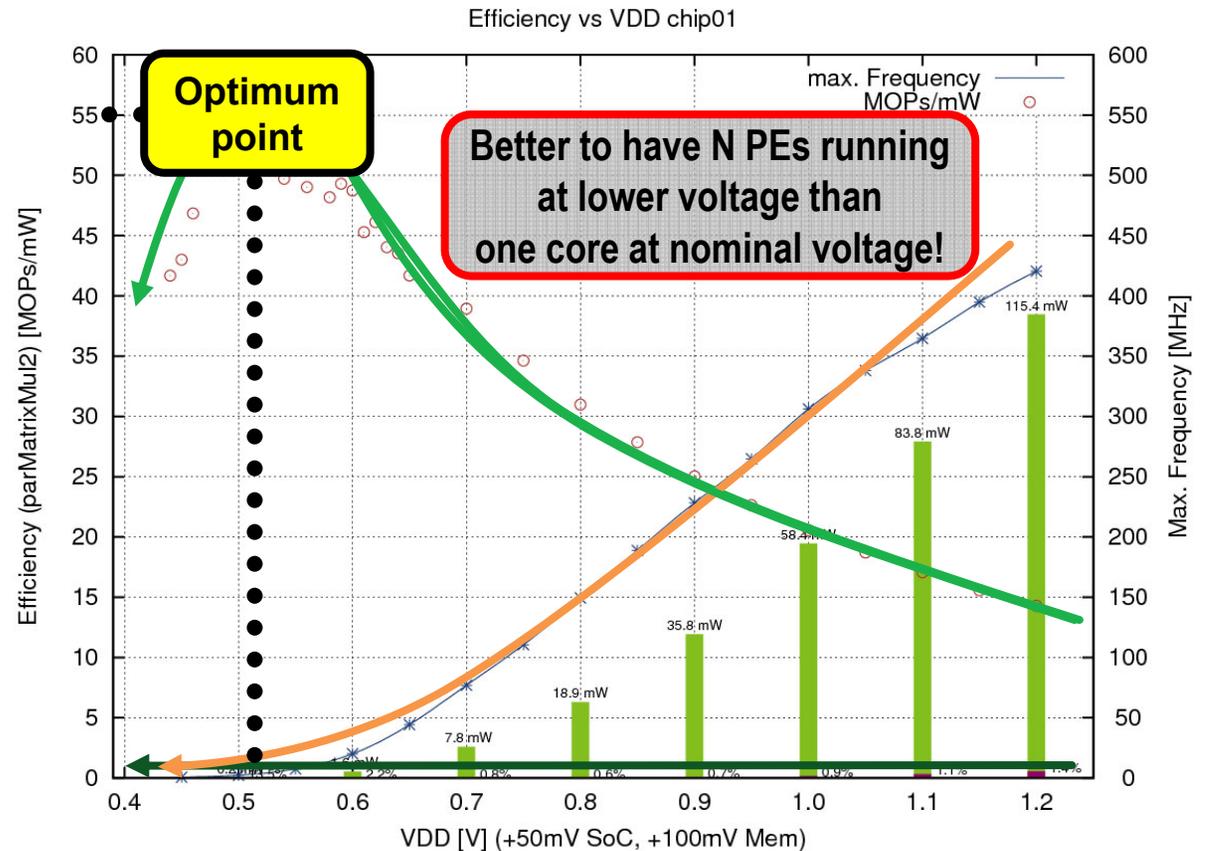
High OP/B ratio  
Massive Parallelism  
MAC-dominated  
Low precision OK  
Model redundancy

# Energy efficiency is THE Challenge



# ML & Near-threshold: a Marriage Made in Heaven

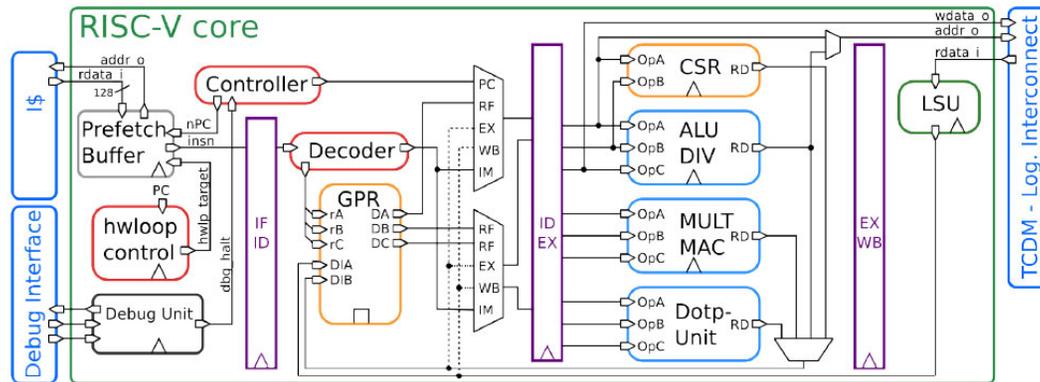
- As VDD decreases, operating speed decreases
- However efficiency increases → more work done per Joule
- Until leakage effects start to dominate
- Put more units in parallel to get performance up and keep them busy with a parallel workload



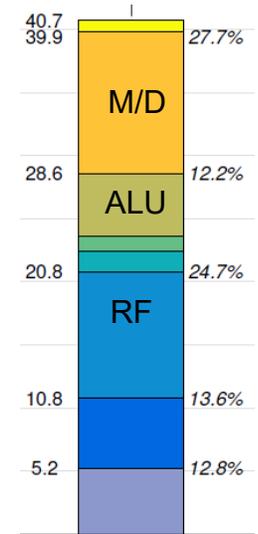
**ML is massively parallel and scales well (P/S ↑ with NN size)**

# The workhorse: A simple RISC-V pipeline + ISA Extensions

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

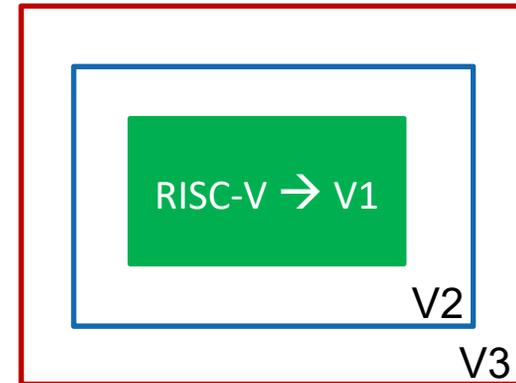


40KG  
70% RF+DP



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

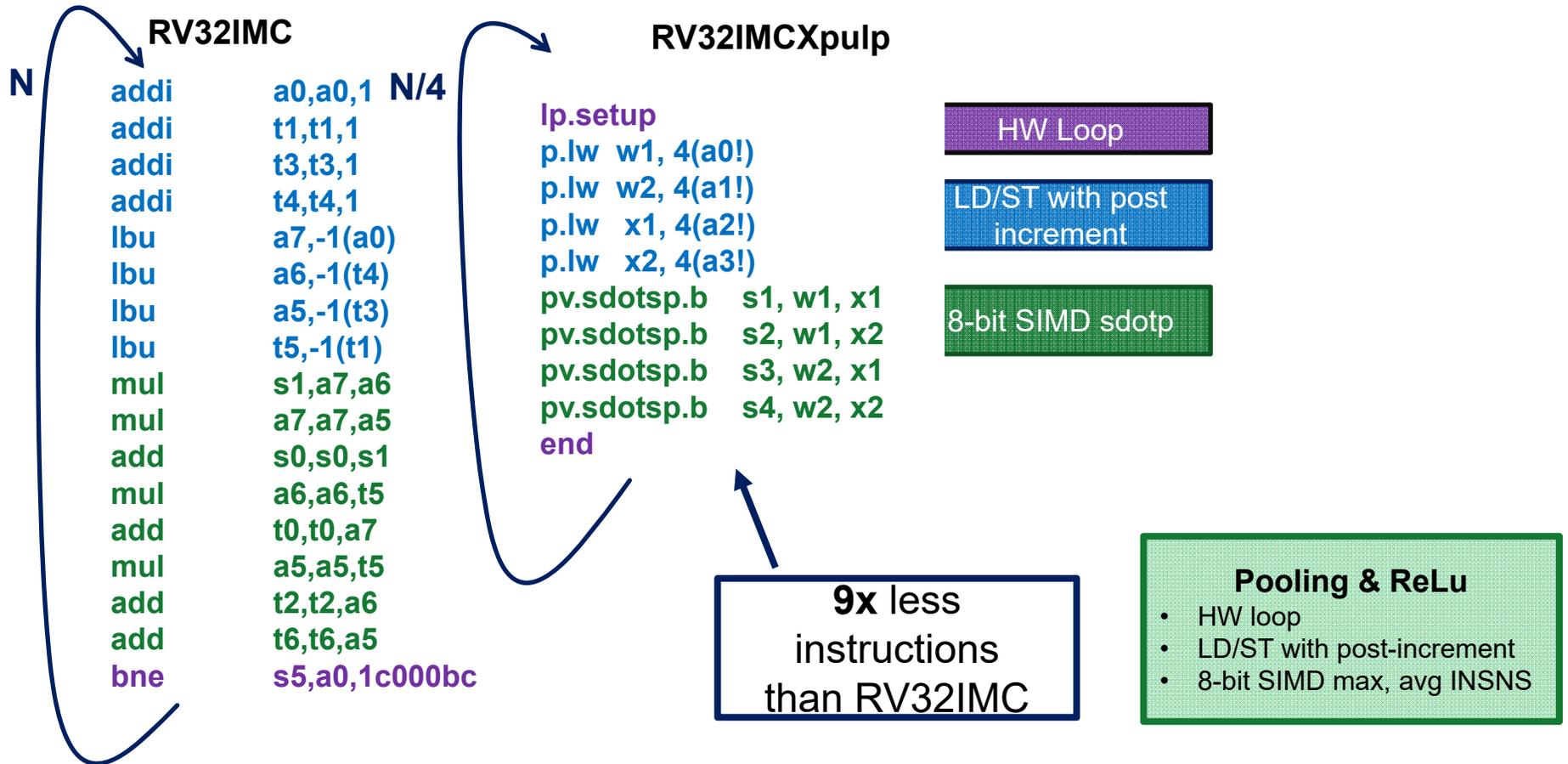
- V1** Baseline RISC-V RV32IMC  
HW loops
- V2** Post modified Load/Store  
Mac
- V3** SIMD 2/4 + DotProduct + Shuffling  
Bit manipulation unit  
Lightweight fixed point (**EML centric**)



**XPULP extensions: 25KG → 40KG (1.6x)**

# PULP-NN: Xpulp ISA exploitation

## 8-bit Convolution



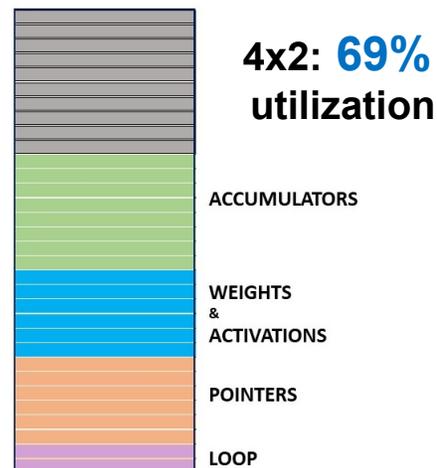
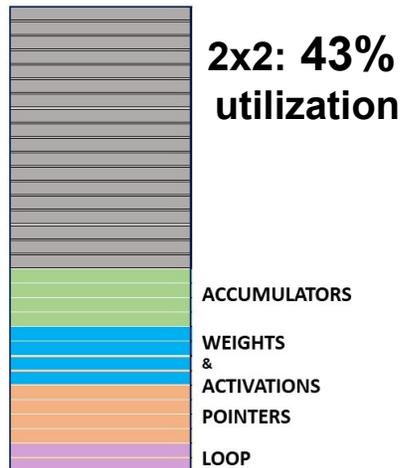
# PULP-NN: Data Reuse in the Register File

## 8-bit Convolution

CMSIS-NN based Matrix Multiplication Layout: 2x2 PULP-NN Matrix Multiplication Layout: 4x2



RegisterFile of the RI5CY core: 32 general purpose registers



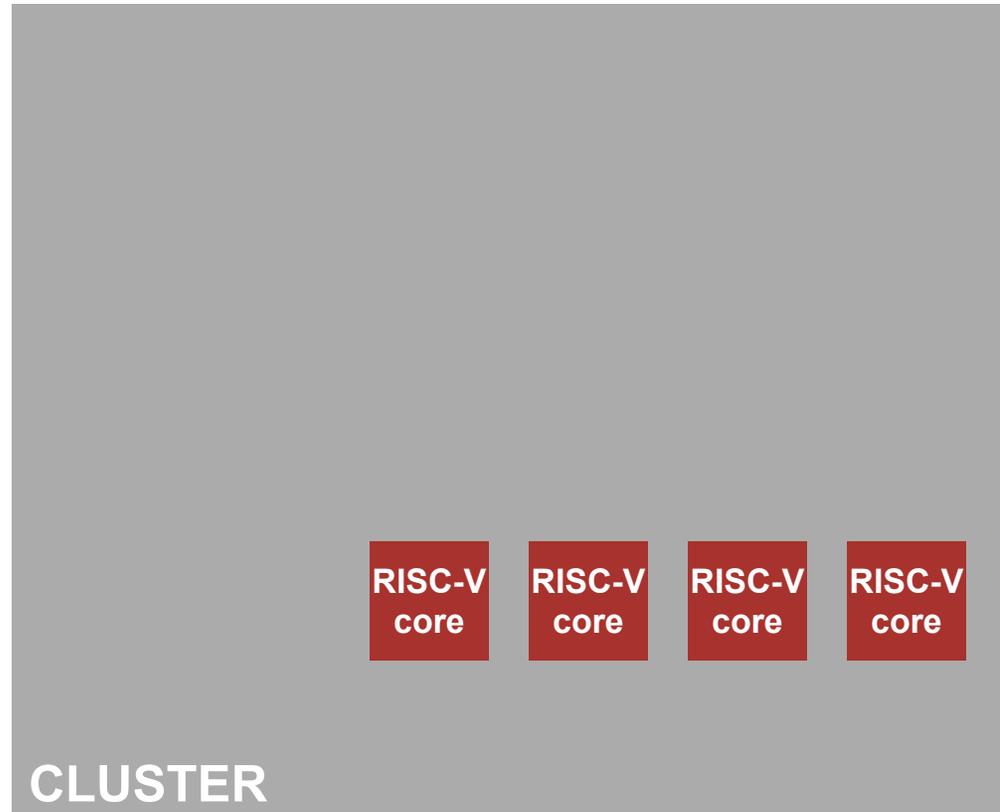
More Data Reuse & Higher utilization of the RF

Peak Performance (8 cores)

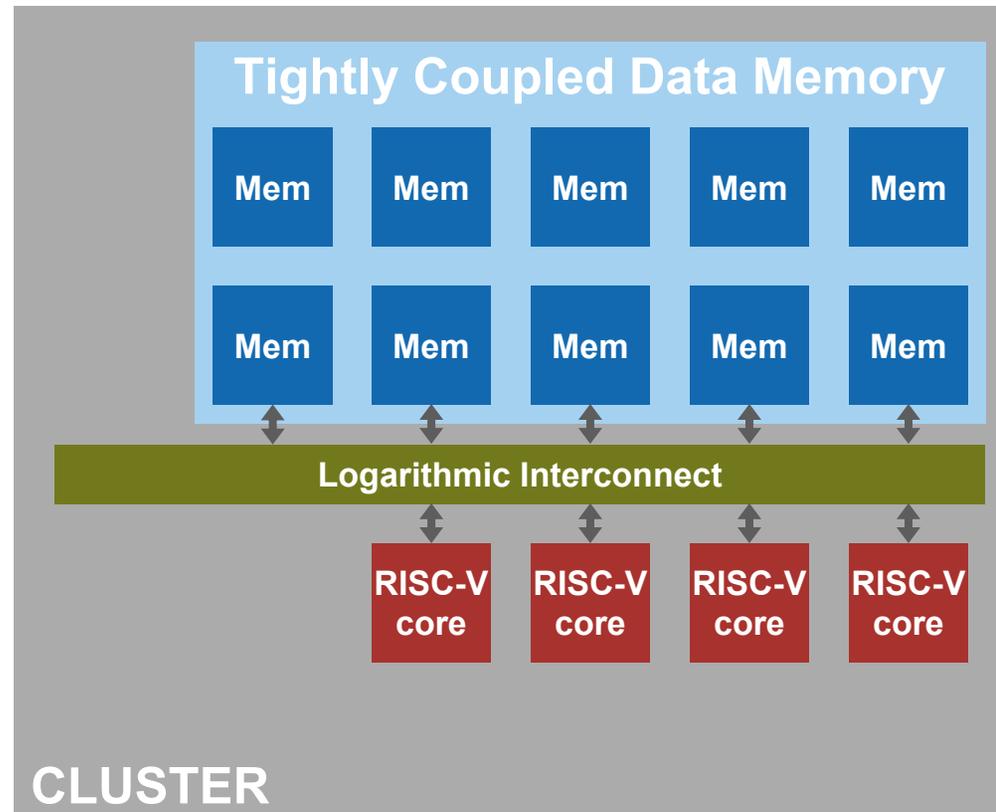
2x2 : 12.8 MAC/cyc

4x2 : 15.5 MAC/cyc

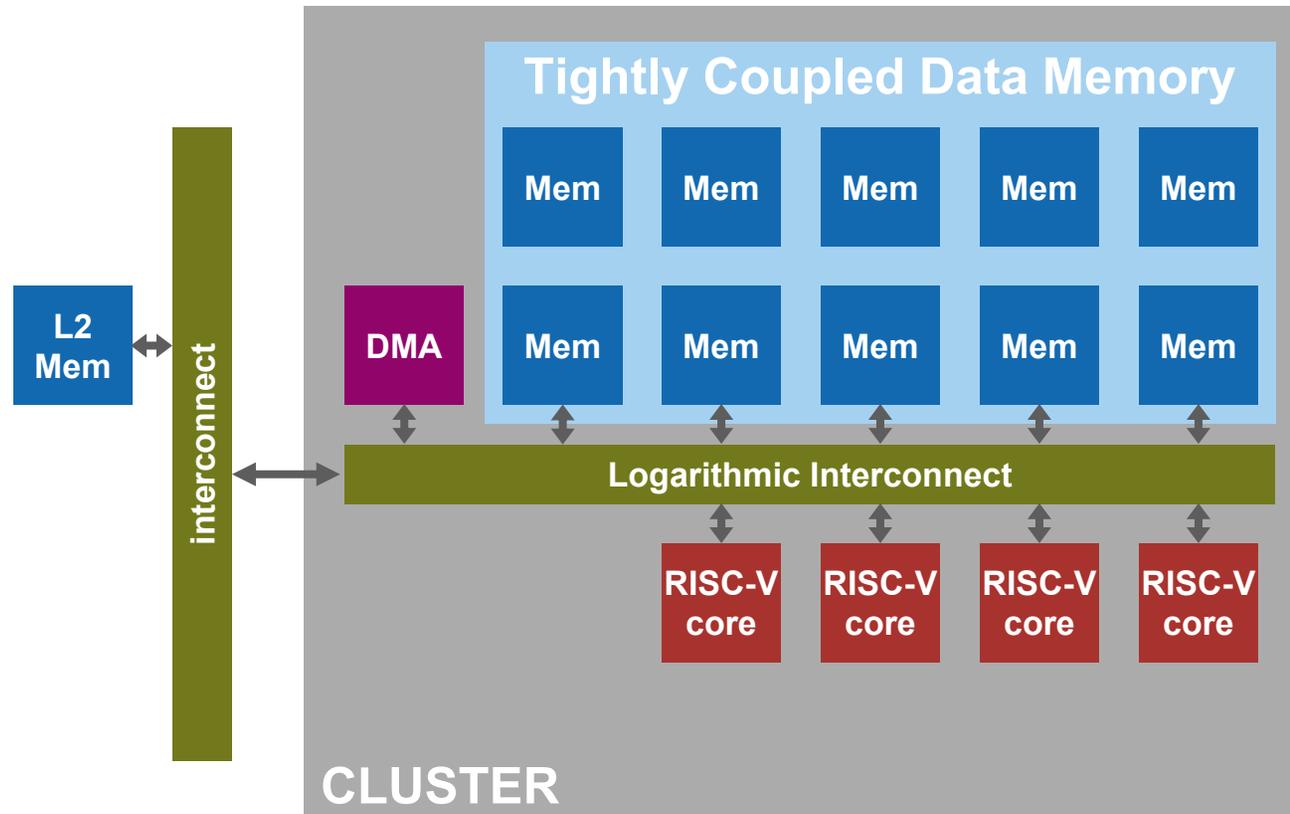
# Multiple RI5CY Cores (1-16)



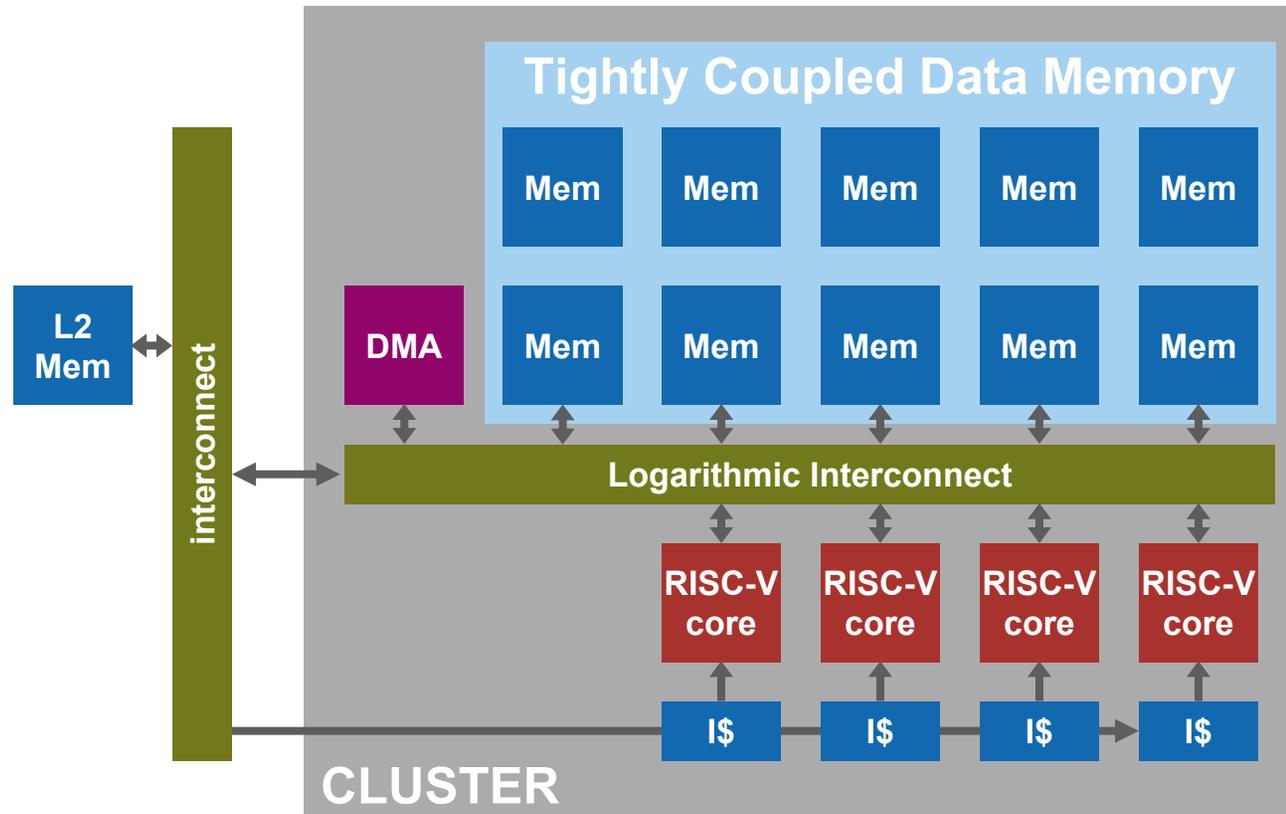
# Low-Latency Shared TCDM



# DMA for data transfers from/to L2

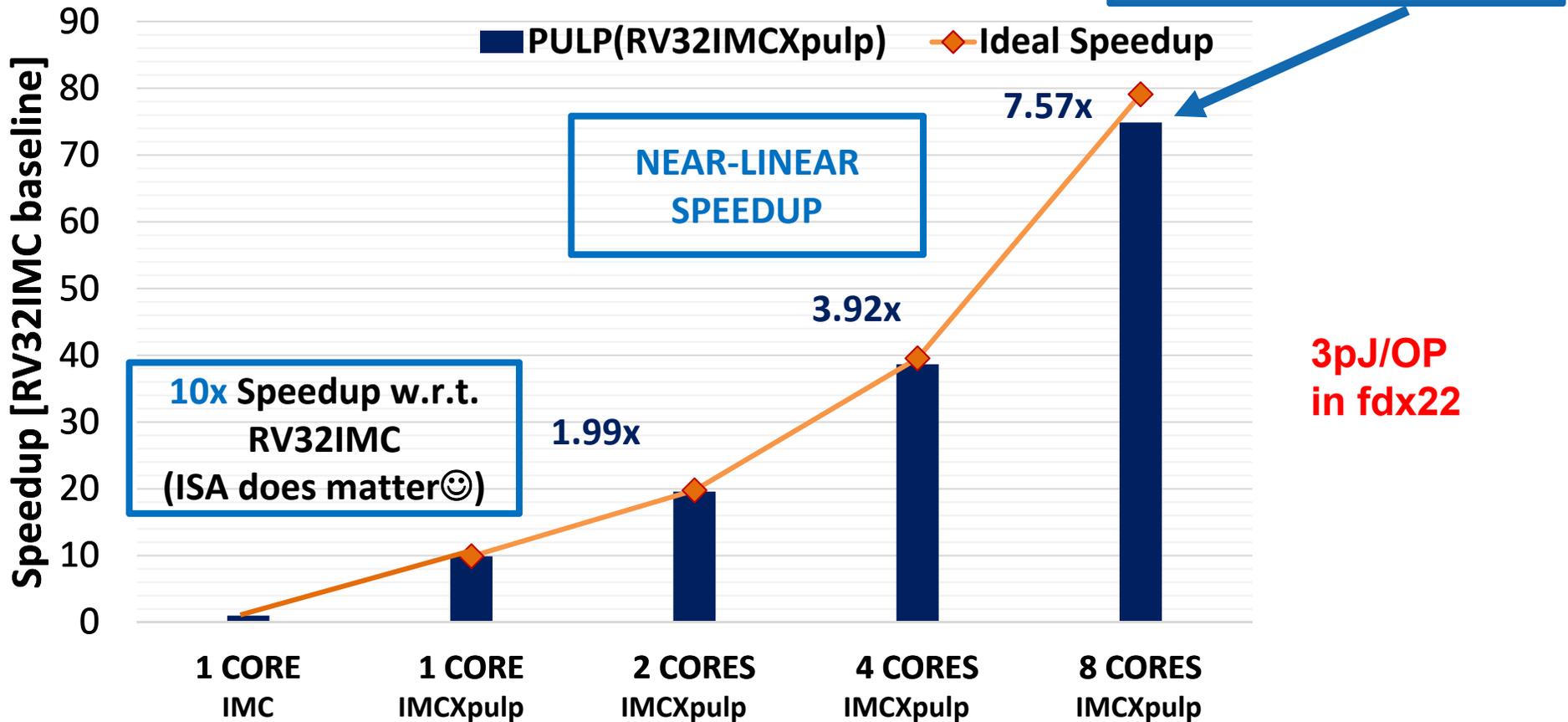


# Shared l2cached with private "loop buffer"



# Results: RV32IMCxpulp vs RV32IMC

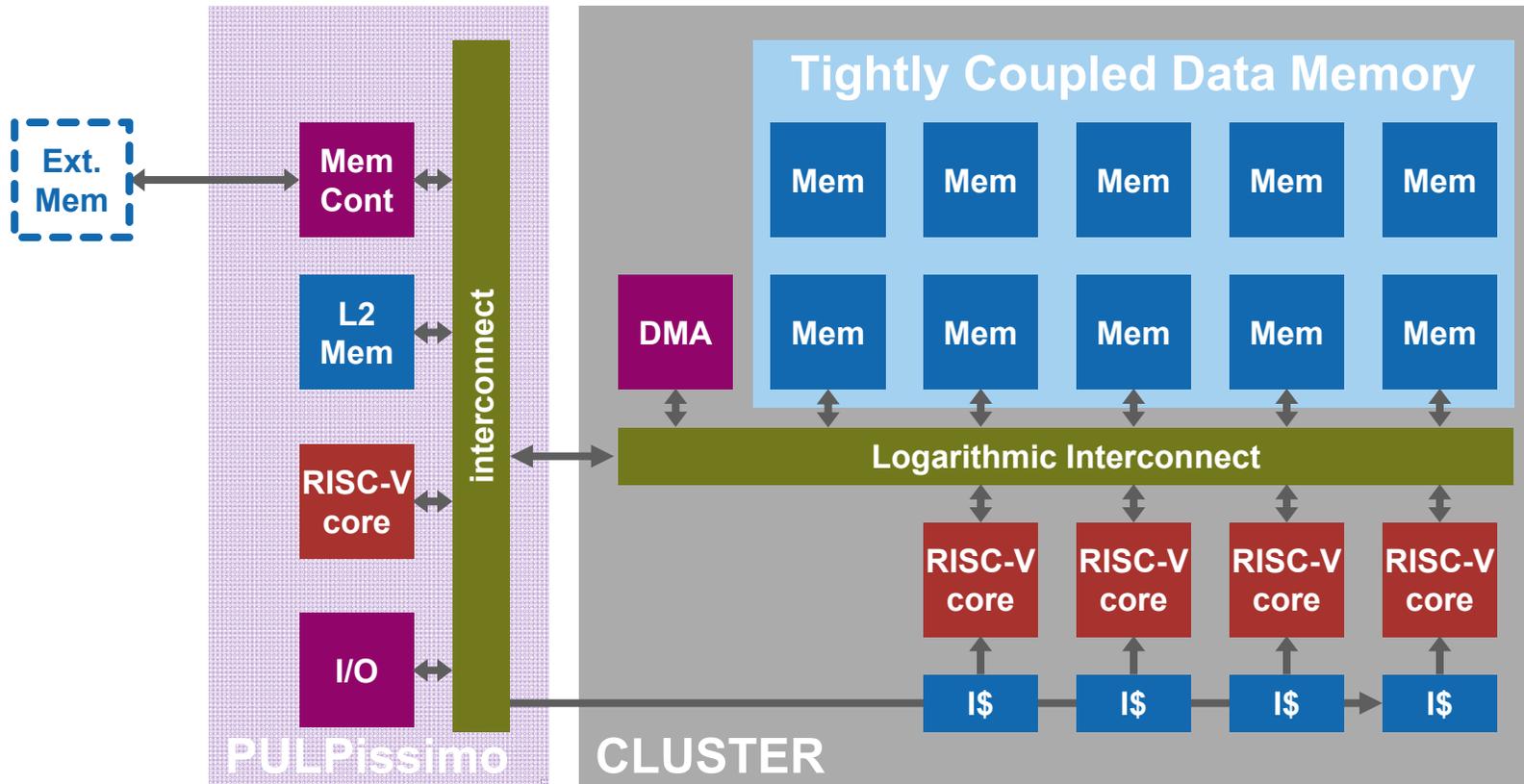
## 8-bit Convolution Results



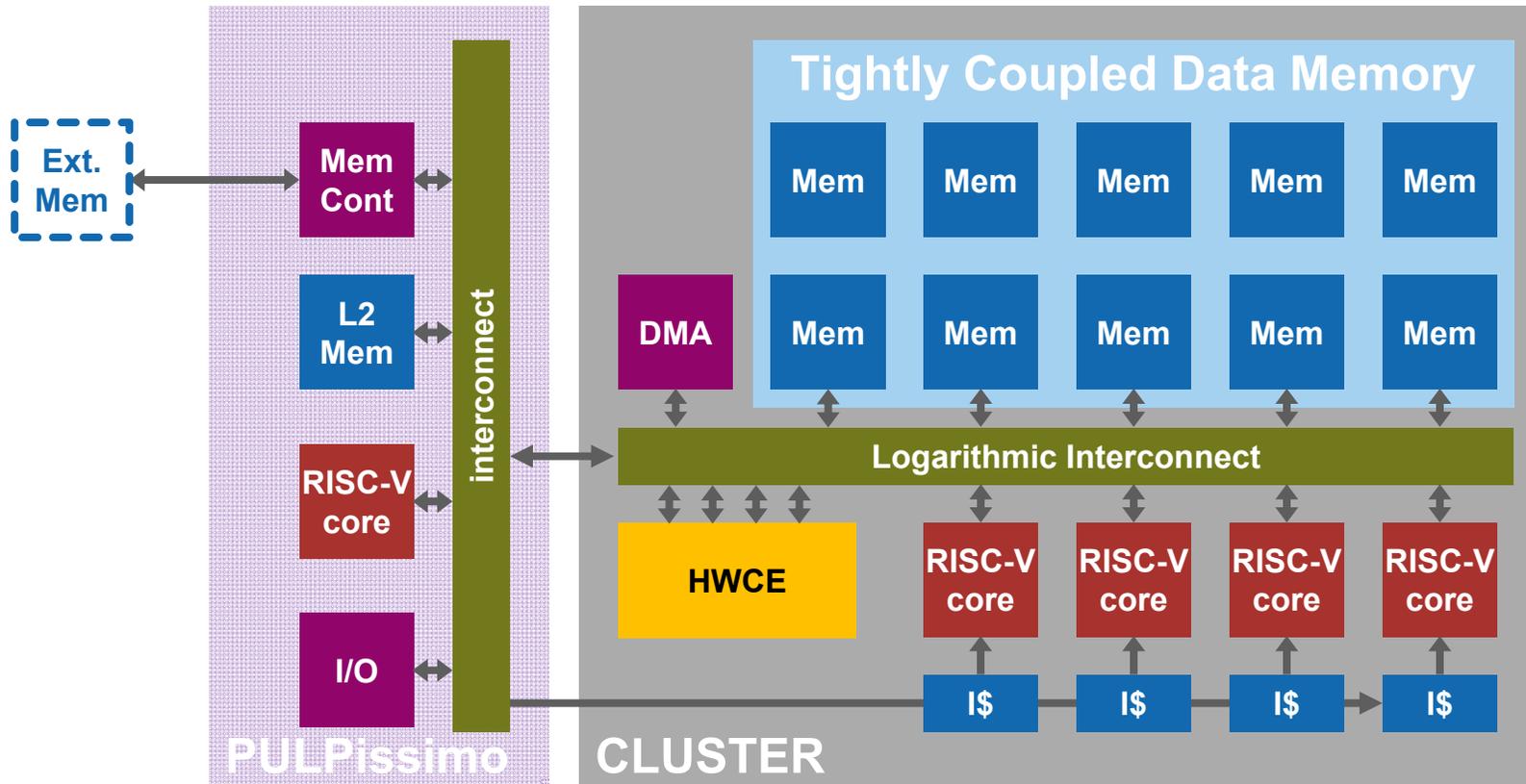
**PULP-NN: an open Source library for DNN inference on PULP cores**



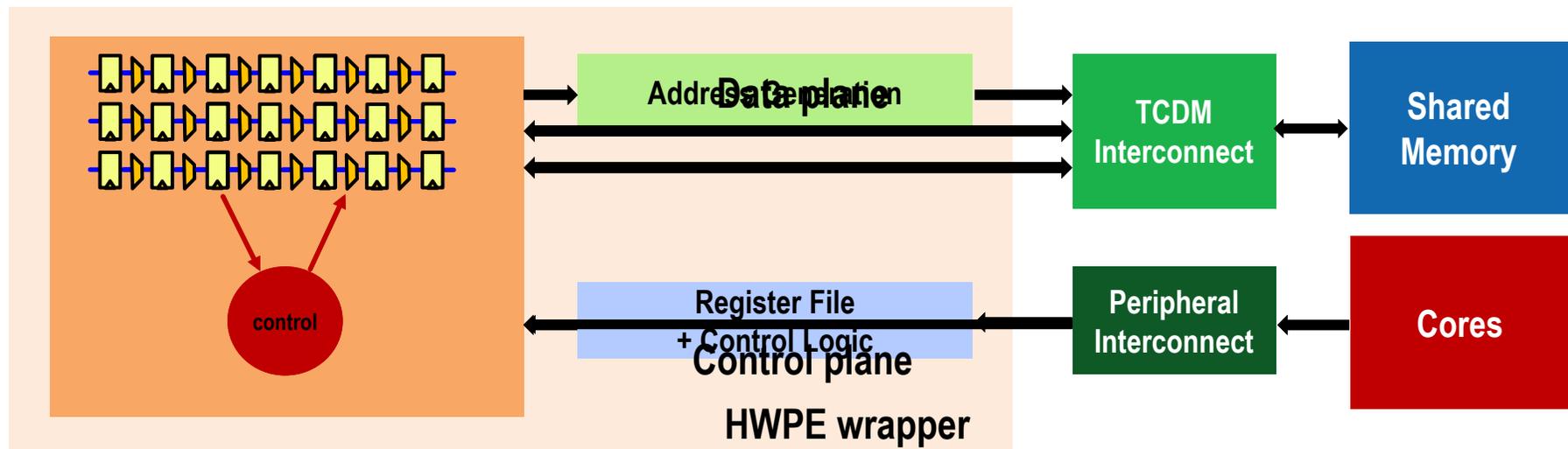
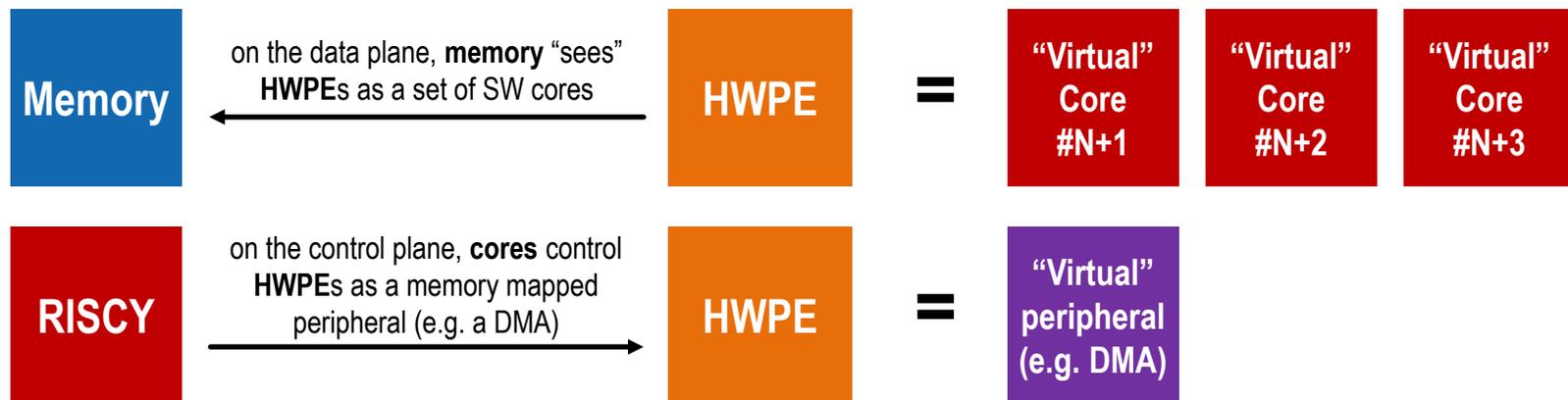
# An additional controller is used for I/O



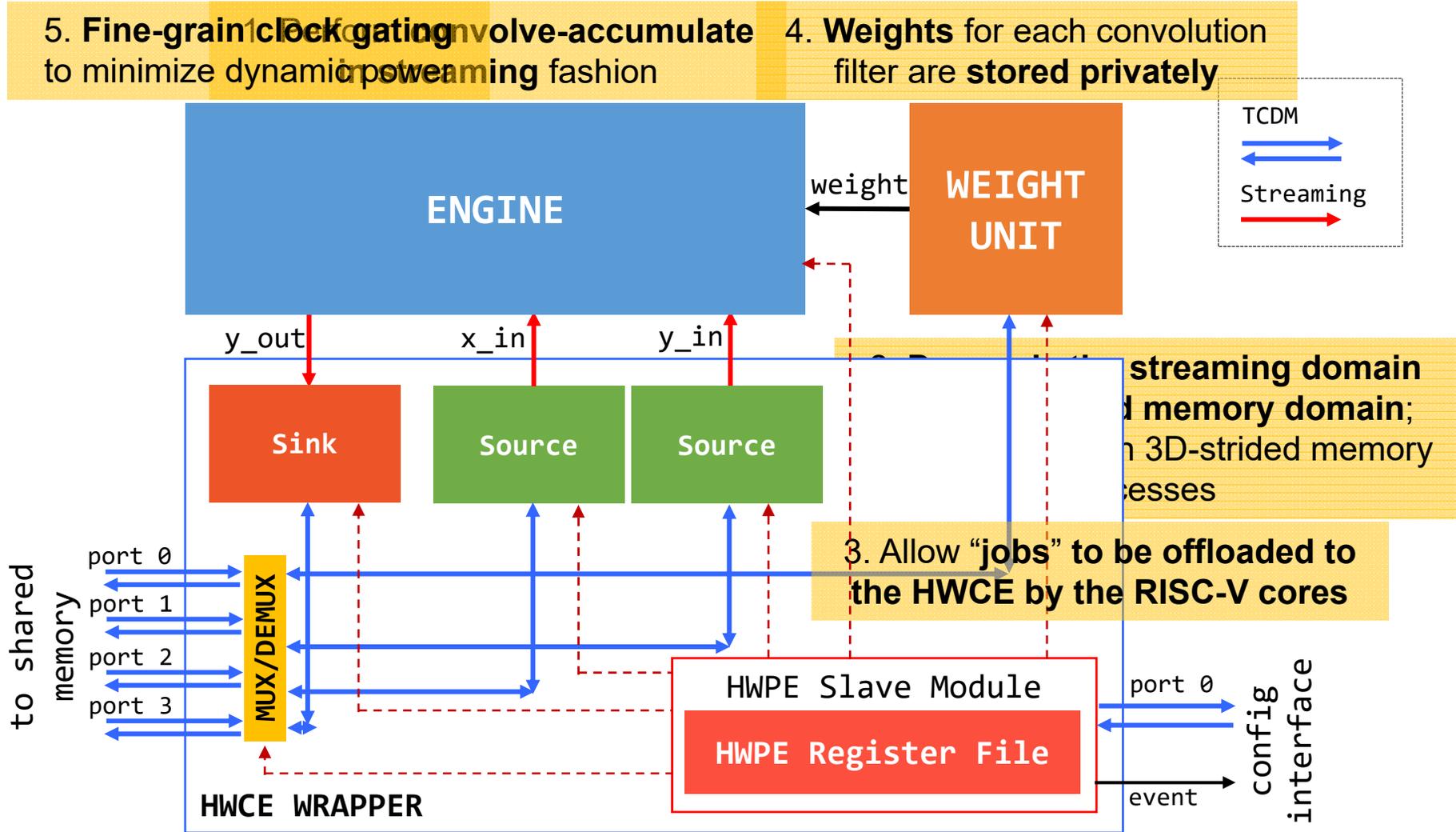
# Sub-pJ/W? Tightly-coupled HW Computing Engine



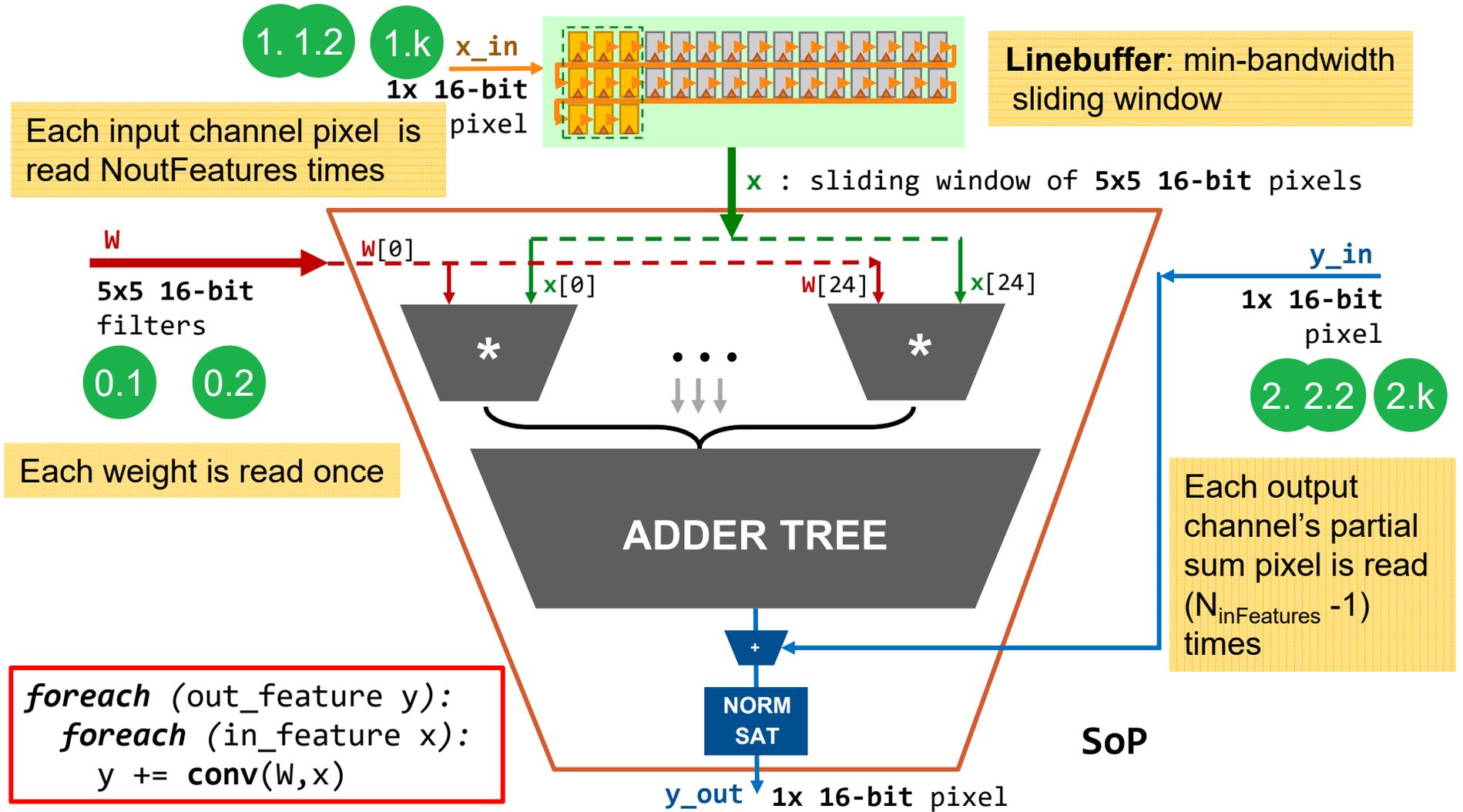
# Hardware Processing Engines (HWPEs)



# HW Convolution Engine



# HWCE Sum-of-Products

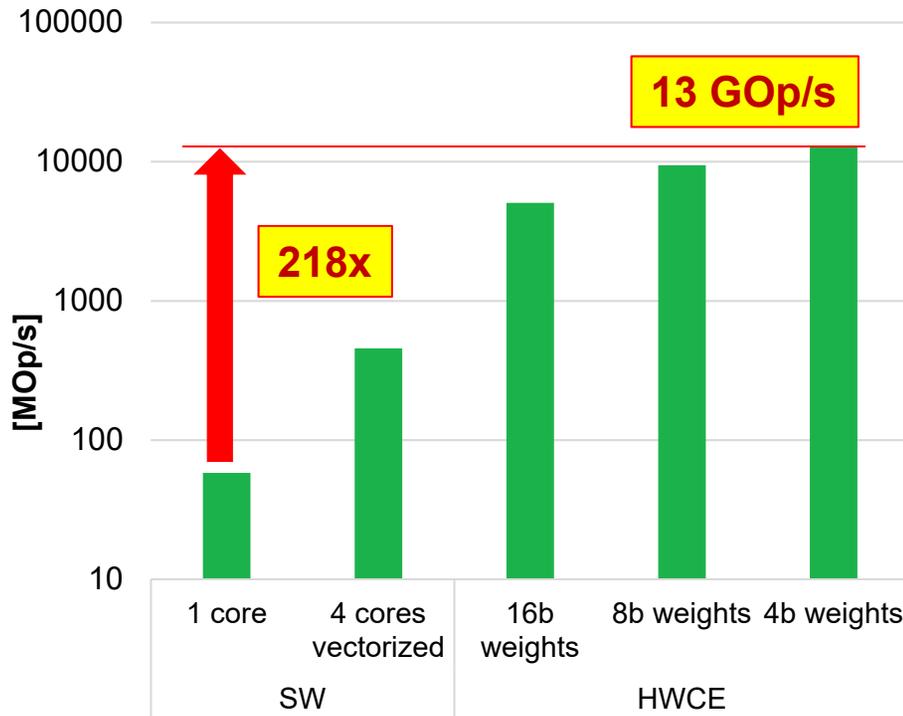


# Heterogeneous PULP CNN Performance

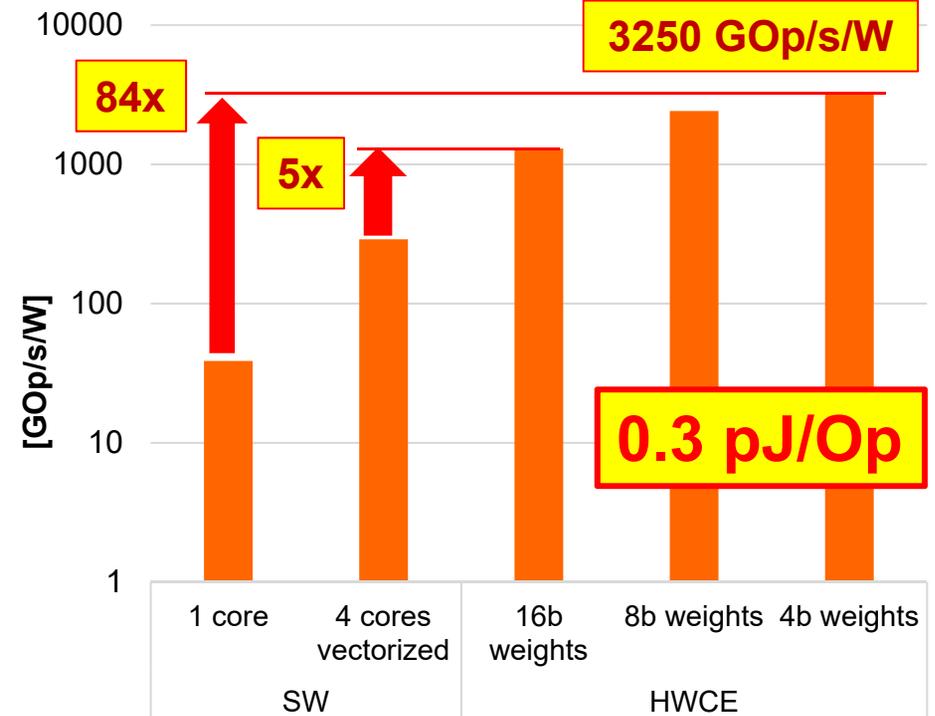
Cluster performance and energy efficiency on a 64x64 CNN layer (5x5 conv)

Scaled to ST FD-SOI 28nm @ Vdd=0.6V, f=115MHz

## PERFORMANCE



## ENERGY EFFICIENCY

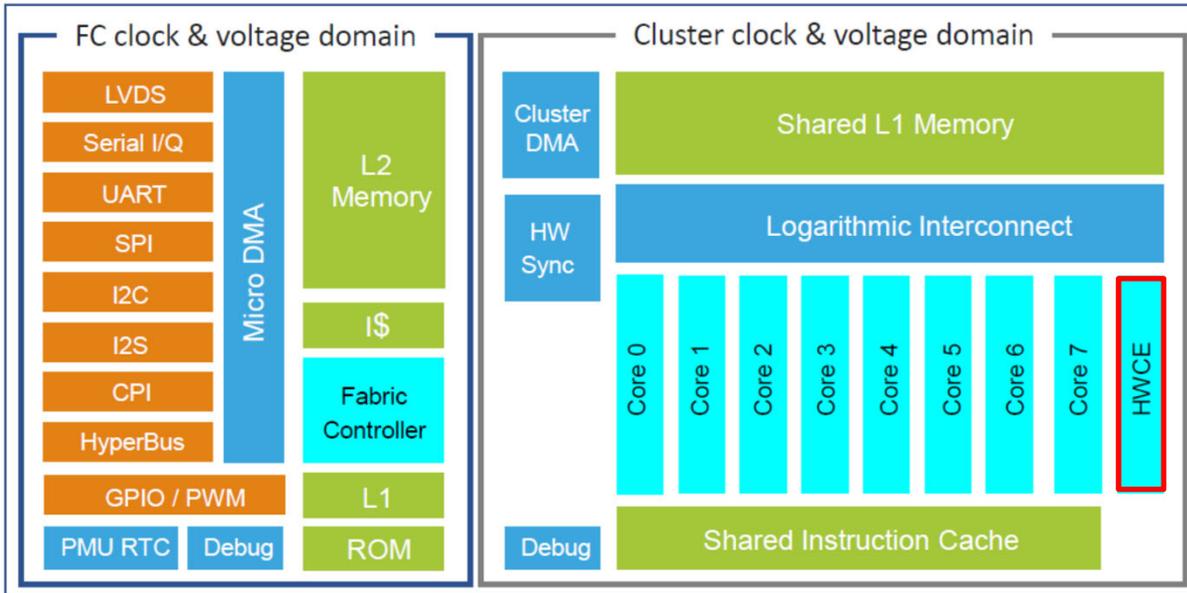


Now coming: HWCEv5.0 – improves scalability & flexibility @ 3TOPS/W



# PULP cluster+MCU+HWCE(V1) → GWT's GAP8 (55 TSMC)

Two independent clock and voltage domains, from 0-133MHz/1V up to 0-250MHz/1.2V



## MCU Function

- Extended RISC-V core
- Extensive I/O set
- Micro DMA
- Embedded DC/DC converter
- Secured execution

## Computation engine

- 8 extended RISC-V cores
- Fully programmable
- Efficient parallelization
- Shared instruction cache
- Multi channel DMA
- HW synchronization
- HW convolution Engine



What	Freq MHz	Exec Time ms	Cycles	Power mW
40nm Dual Issue MCU	216	99.1	21 400 000	60
GAP8 @1.0V	15.4	99.1	1 500 000	3.7
GAP8 @1.2V	175	8.7	1 500 000	70
GAP8 @1.0V w HWCE	4.7	99.1	460 000	0.8

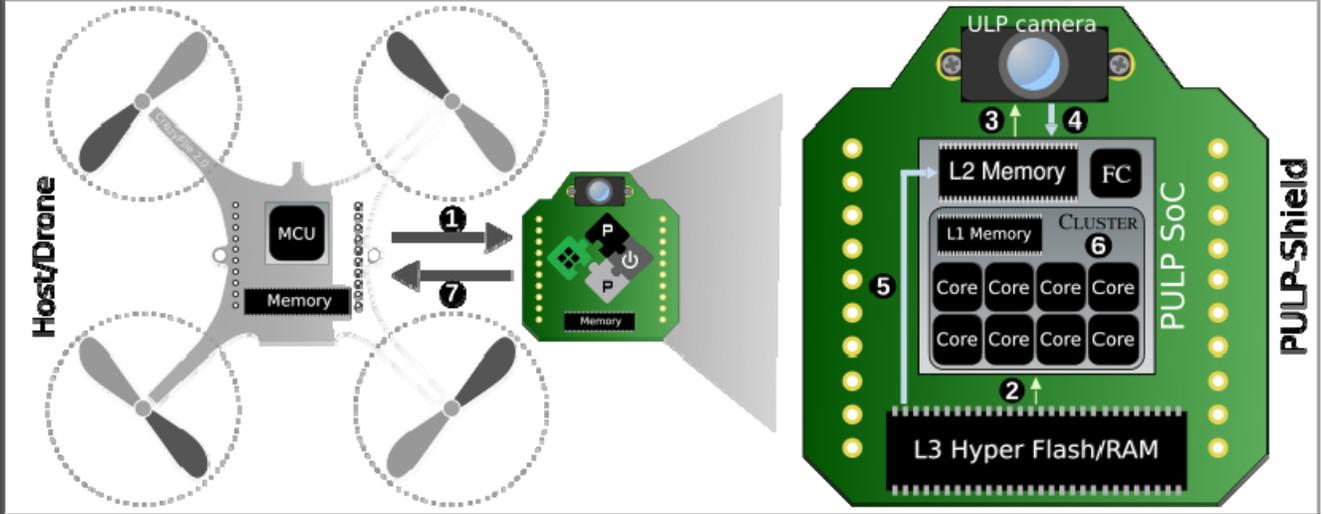


**4x More efficiency at less than 10% area cost**



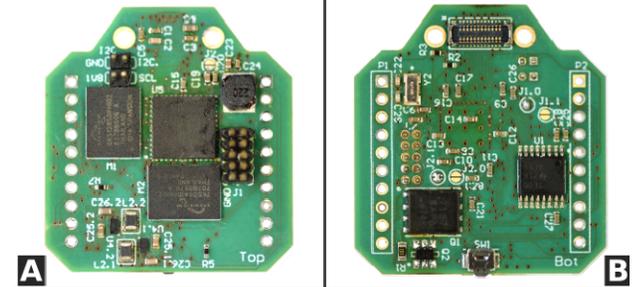
# New Application Frontiers: DroNET on NanoDrone

- 1 Init interrupt (GPIO)
- 2 Load binary (HyperBus)
- 3 Configure camera (I2C)
- 4 Grab frames (µDMA)
- 5 Load weights (HyperBus)
- 6 PULP computation
- 7 Write-back results (SPI)



Pluggable PCB:  
PULP-Shield

- ~5g,  
30×28mm
- GAP8 SoC
- 8 MB  
HDRAM
- 16 MB  
HFlash
- QVGA ULP  
HiMax  
camera
- Crazyflie 2.0  
nano-drone  
(27g)



**Only onboard computation for autonomous flight + obstacle avoidance  
no human operator, no ad-hoc external signals, and no remote base-station!**



# More Efficiency (2): Extreme Quantization

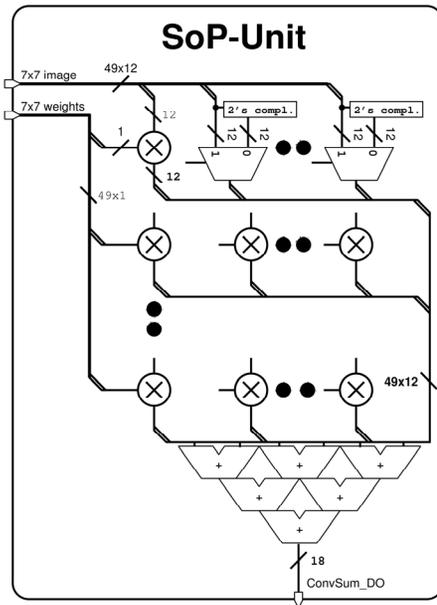
Low(er) precision: 8 → 4 → 2

Model	Bit-width	Top-1 error
ResNet-18 ref	32	31.73%
INQ	5	31.02%
INQ	4	31.11%
INQ	3	31.92%
INQ	2 (ternary)	33.98%

SOA INQ retraining

2.2% loss → 0% with 20% larger net

MULT → MUX



Equivalent for 7x7 SoP

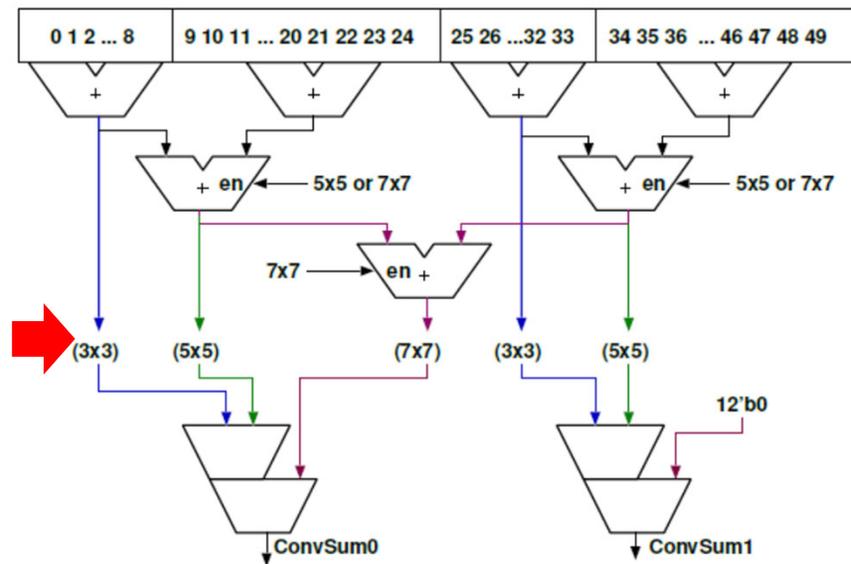
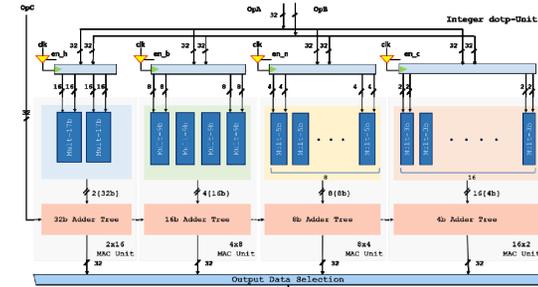
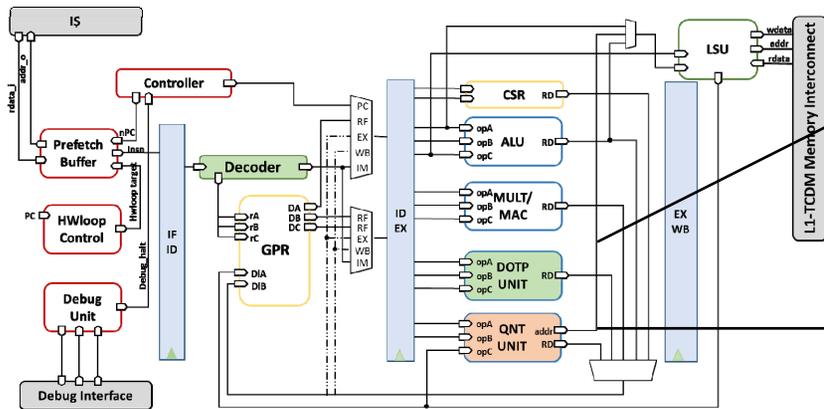


Image Mapping (3x3, 5x5, 7x7)

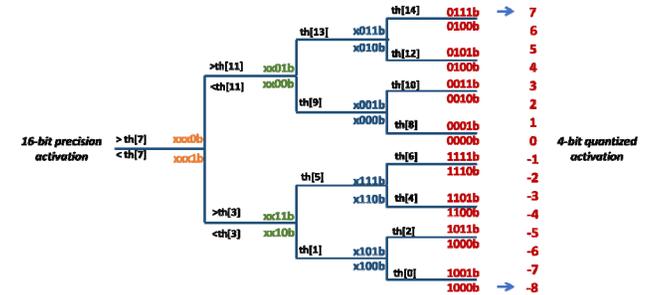
1 MAC Op = 2 Op (1 Op for the “sign-reverse”, 1 Op for the add).

# RISC-V ISA Extensions for extreme quantization

## RI5CY microarchitectural extensions



## 2-bit & 4-bit SIMD DOTP + OP Isolation



## QNT UNIT: 2 Quantizations in 9 Cycles

- Overheads (28nm FDX PULPissimo impl.):
  - Area: ~11% (vs. Ri5CY)
  - Timing Overhead: negligible (integrated in PULPissimo)
  - 8-bit MatMul power overhead: 1.8% (integrated in PULPissimo)
  - GP-app power overhead: 3.5% (integrated in PULPissimo)

# From +/-1 Binarization to XNORs

$$y(k_{out}) = \text{binarize}_{\pm 1} \left( \mathbf{b}_{k_{out}} + \sum_{k_{in}} \left( \mathbf{W}(k_{out}, k_{in}) \otimes \mathbf{x}(k_{in}) \right) \right)$$

**XNOR**

$$\text{binarize}_{\pm 1}(t) = \text{sign} \left( \gamma \frac{t - \mu}{\sigma} + \beta \right)$$

$$\text{binarize}_{0,1}(t) = \begin{cases} 1 & \text{if } t \geq -\kappa/\lambda \doteq \tau, \text{ else } 0 & (\text{when } \lambda > 0) \\ 1 & \text{if } t \leq -\kappa/\lambda \doteq \tau, \text{ else } 0 & (\text{when } \lambda < 0) \end{cases}$$

$$y(k_{out}) = \text{binarize}_{0,1} \left( \sum_{k_{in}} \left( \mathbf{W}(k_{out}, k_{in}) \otimes \mathbf{x}(k_{in}) \right) \right)$$

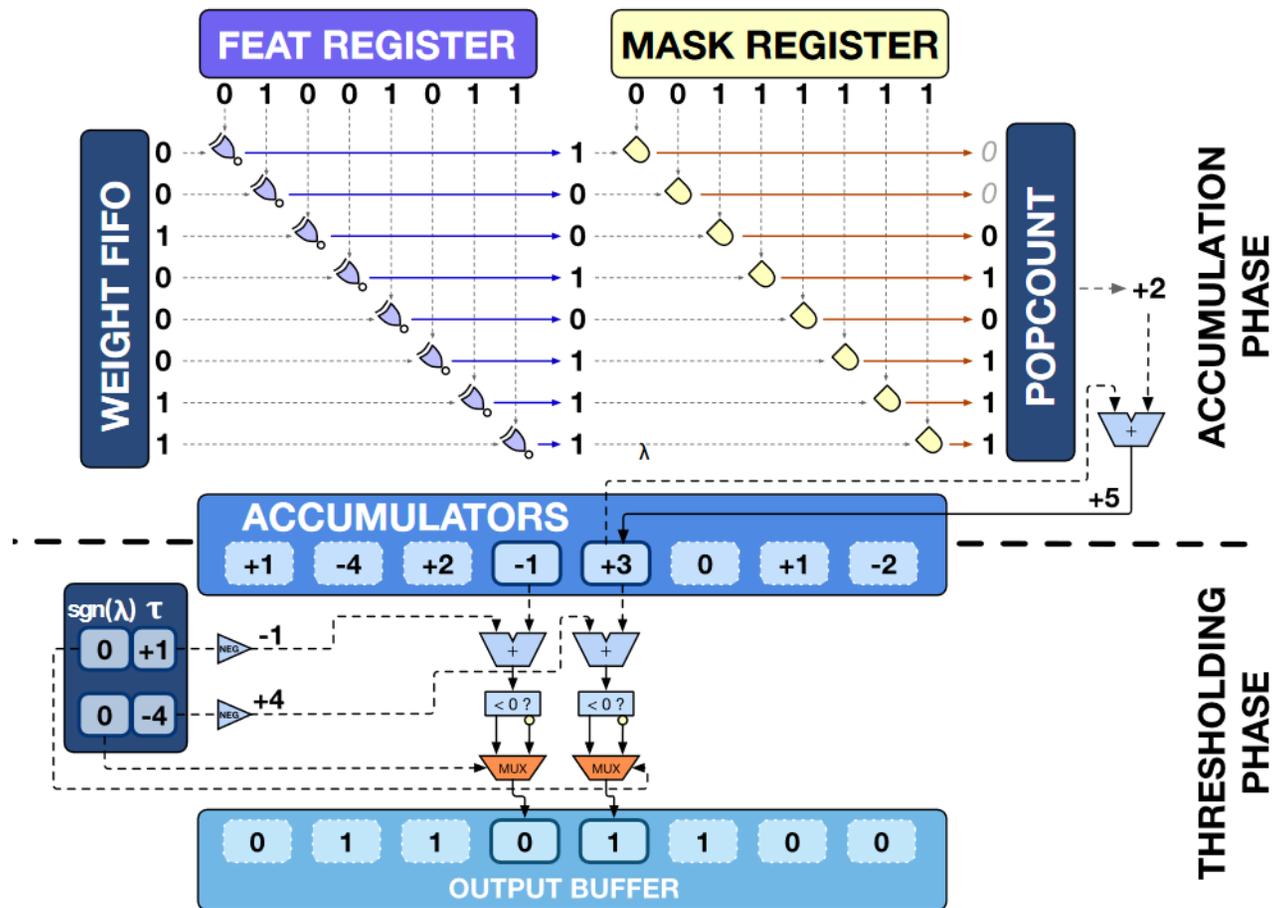
Thresholding

Multi-bit accumulation

Binary product → XOR

A	B	ou t	A	B	ou t
-1	-1	+1	0	0	1
-1	+1	-1	0	1	0
+1	-1	-1	1	0	0
+1	+1	+1	1	1	1

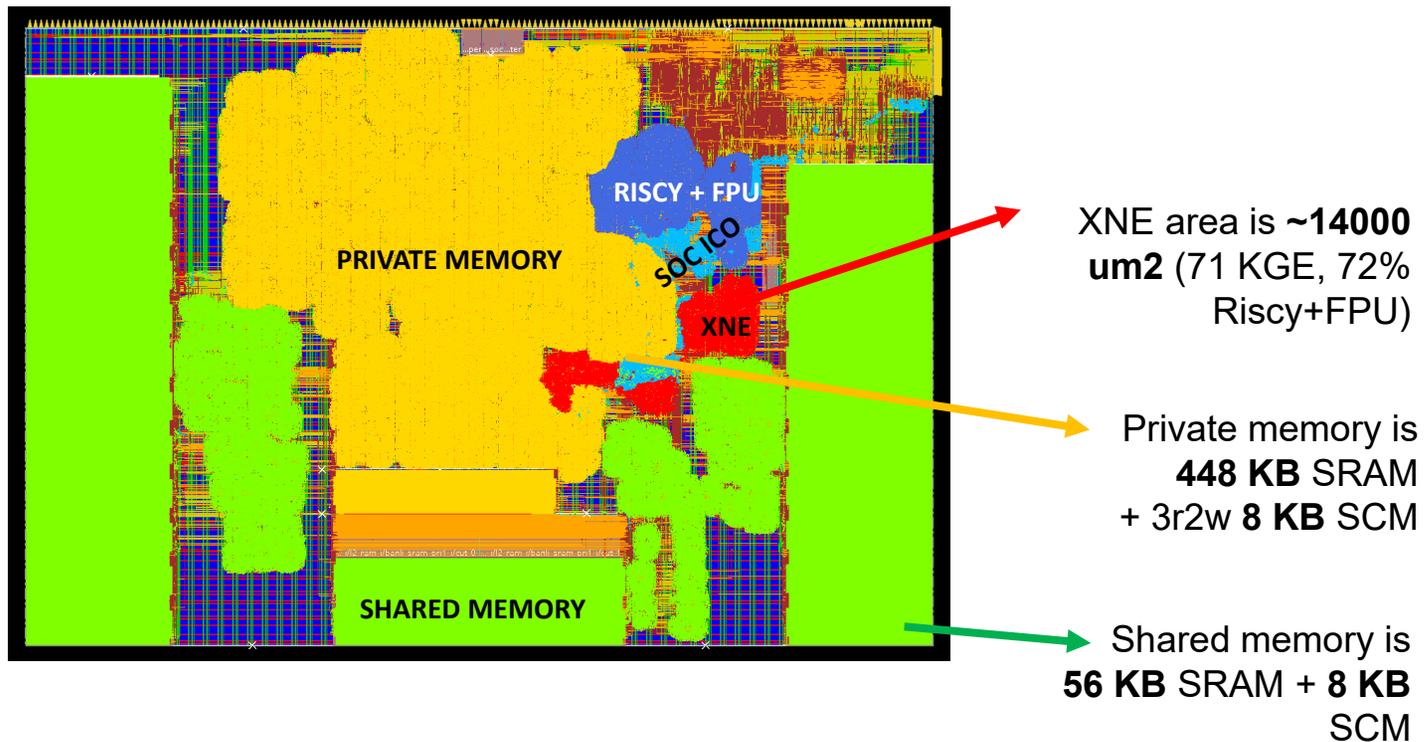
# XNE: XNOR Neural Engine



Main unit: binary dot-product and thresholding

# Quentin: a XNE-accelerated microcontroller

## Quentin in GlobalFoundries 22FDX

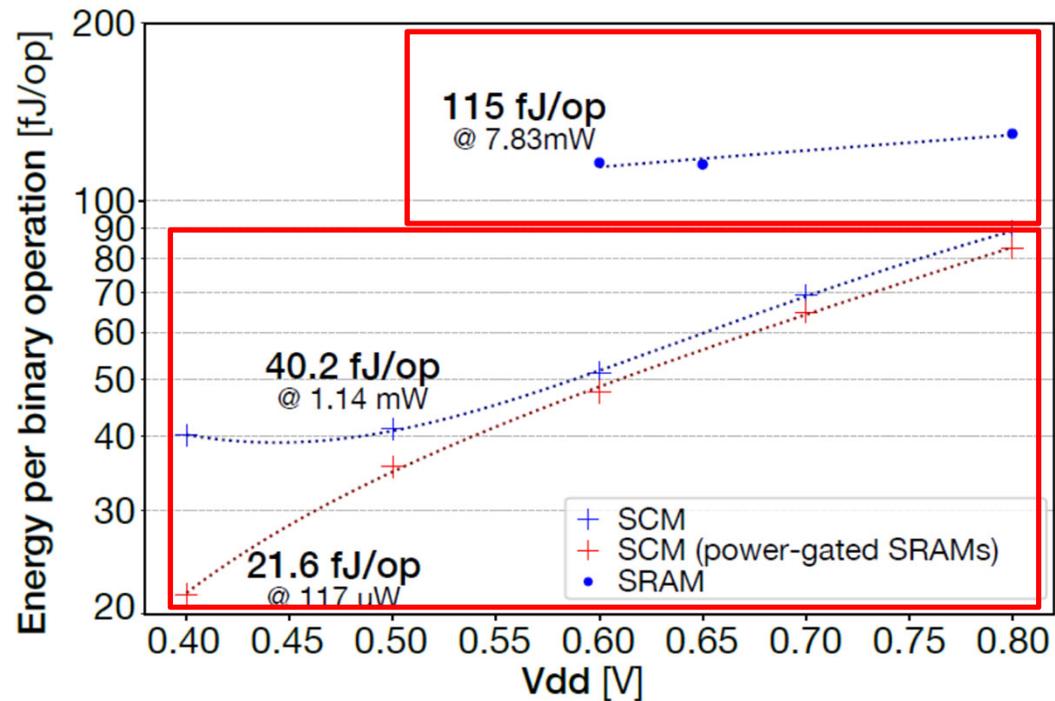


F. Conti, P. D. Schiavone and L. Benini, "XNOR Neural Engine: A Hardware Accelerator IP for 21.6-fJ/op Binary Neural Network Inference," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 11, pp. 2940-2951, Nov. 2018.

# XNE Energy Efficiency

With SRAMs, max eff  
@ 0.65V 8.7 Top/s/W

With SCMs, max eff  
@ 0.5V 46.3 Top/s/W



Accuracy Loss is high even with retraining (10%+) → mixed precision  
TWN & TCN are also a very appealing alternative (under design)

# Binary-based Quantization (BBQ)

$Q_W$ : weight quantization level

$Q_A$ : activation quantization level

**Normal NN layer:** 
$$\mathbf{y}(k_{out}) = \mathbf{b}(k_{out}) + \sum_{k_{in}} (\mathbf{W}(k_{out}, k_{in}) \otimes \mathbf{x}(k_{in}))$$

*Inspired by ABC-Net:*

**BBQ NN layer:**

$$\mathbf{y}(k_{out}) \approx \mathbf{b}(k_{out}) + \sum_{i=0..Q_W} \sum_{j=0..Q_A} \sum_{k_{in}} \alpha_i \beta_j \left[ \mathbf{W}_{bin}(k_{out}, k_{in}) \otimes \mathbf{x}_{bin}(k_{in}) \right]$$

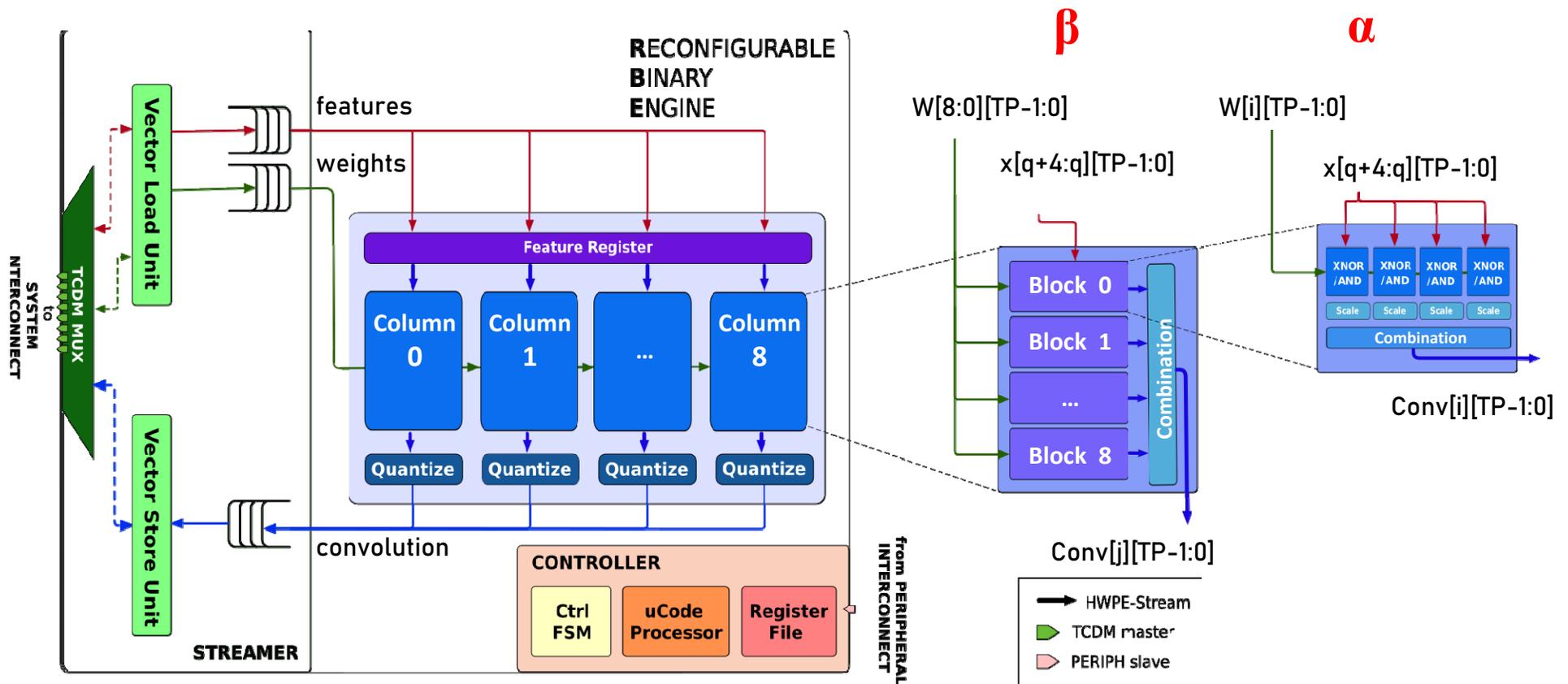
$\alpha_i, \beta_j$ : power-of-2  
Scale Module

Binary NN  
BinConv  
module

*One quantized NN can be emulated by superposition of power-of-2 weighted  $Q_A \times Q_W$  binary NN*

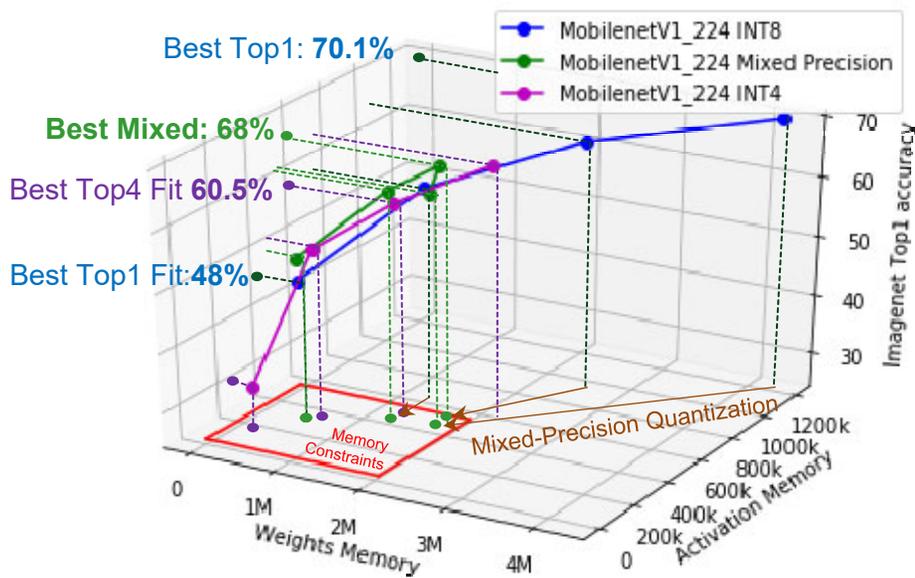
# Reconfigurable Binary Engine

$$y(k_{out}) \approx \mathbf{b}(k_{out}) + \sum_{i=0..Q_W} \sum_{j=0..Q_A} \sum_{k_{in}} \alpha_i \beta_j \mathbf{W}_{bin}(k_{out}, k_{in}) \otimes \mathbf{x}_{bin}(k_{in})$$

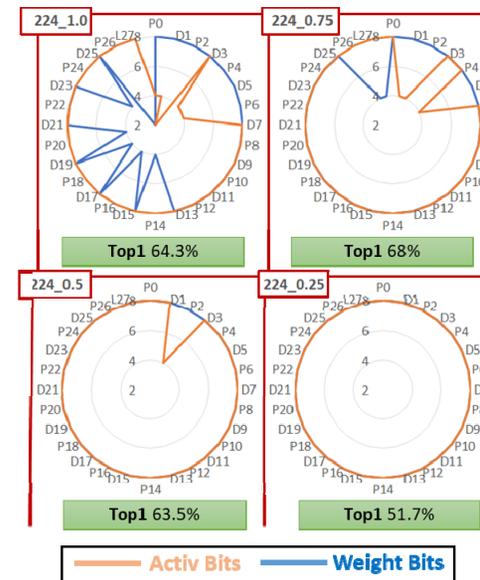


# Mixed-Precision Quantized Networks

Apply minimum tensor-wise quantization to fit the **memory constraints** with very-low accuracy drop

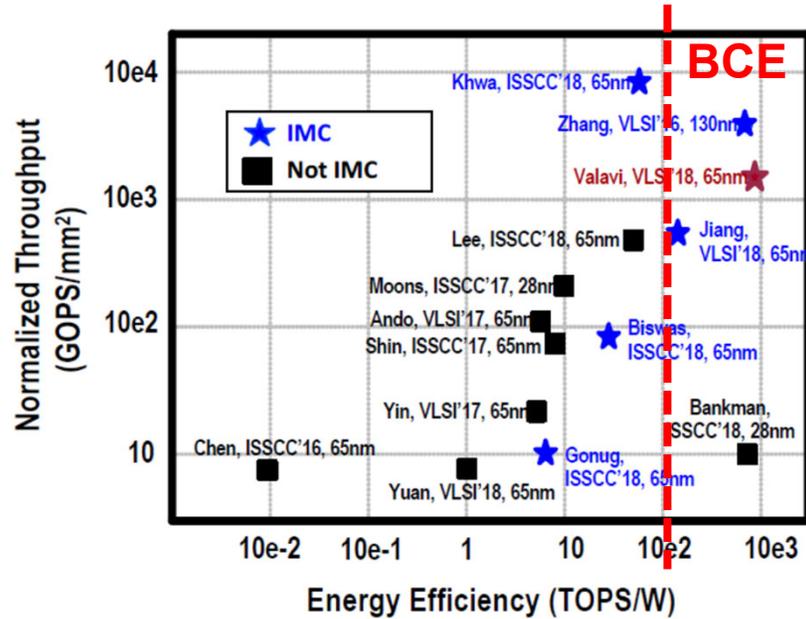


**mixed-precision** quantization  
 rule-based bit precision selection based on  
 memory constraints



- Only -2% wrt most accurate INT8 mobilenet (224\_1.0) which does not fit on-chip
- +8% wrt most accurate INT8 mobilenet fitting on-chip (192\_0.5)
- +7.5% wrt most accurate INT4 mobilenet (224\_1.0) fitting on chip

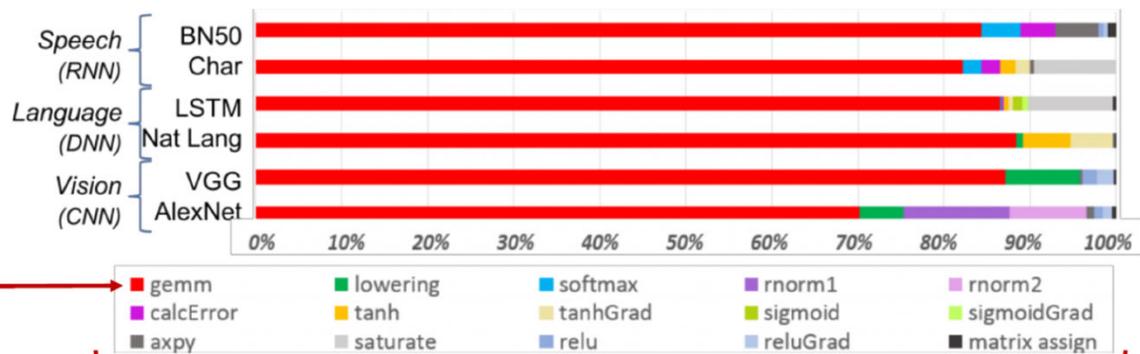
# Diversion: why not a fully hardwired engine?



➔ 1000TOPS/W...

**Flexibility is essential!**

However....



➔ Amdhal's effect!  
+  
Mixed precision  
+  
NN "zoo"

**General Matrix Multiply**  
(~256×2300=590k elements)

**Single/few-word operands**  
(traditional, near-mem. acceleration)

[B. Fleischer, VLSI'18]

# What about $\mu$ W «sleep»?

Small always-on network  $\rightarrow$  triggers alarm and video capture/streaming for cloud-based forensics

**Camera**

**Digital Processor**

- DC/DC
- Power Manager
- L2 Memory
- Smart Periph
  - Camera IF
  - SPI
  - I2S
  - I2C
  - Micro DMA
- L1
- Fabric Controller Core

**Radio Subsystem**

- LoRa Semtech 1276
- Power Management
- Wi-Fi/Bluetooth CYW4343W

Cluster Region

- Core 0
- Core 1
- Core 2
- Core 3
- Shared L1

JPEG encoding on c

Mcycles

Method	Mcycles
Fixed Point DCT	~100
Fast-DCT	~14.1
8 Cores & Optimizations	~2.6

7.1x (Fixed Point DCT vs Fast-DCT)

3.9x (Fast-DCT vs 8 Cores & Optimizations)

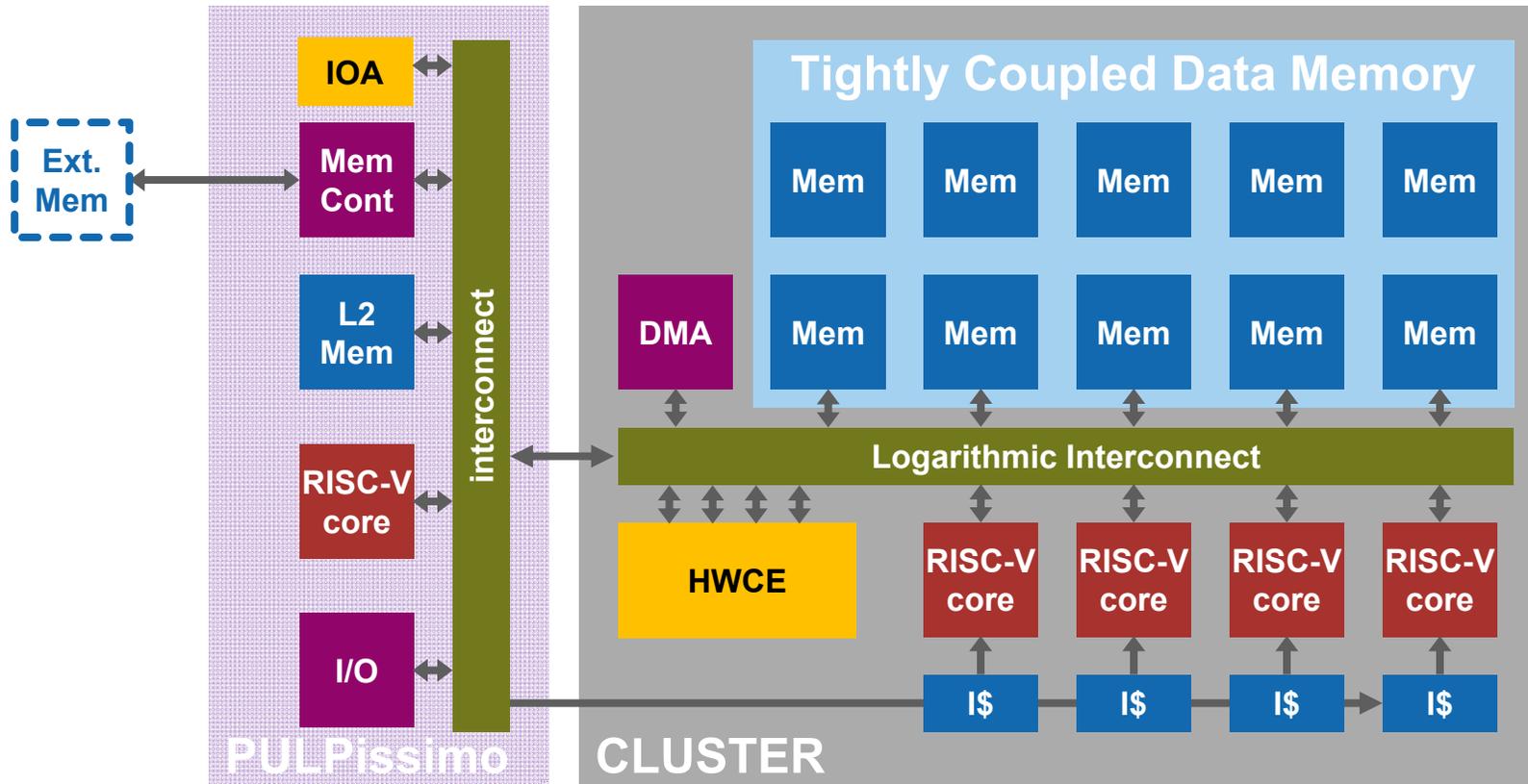
~20fps@50MHz

~20fps@50Mhz

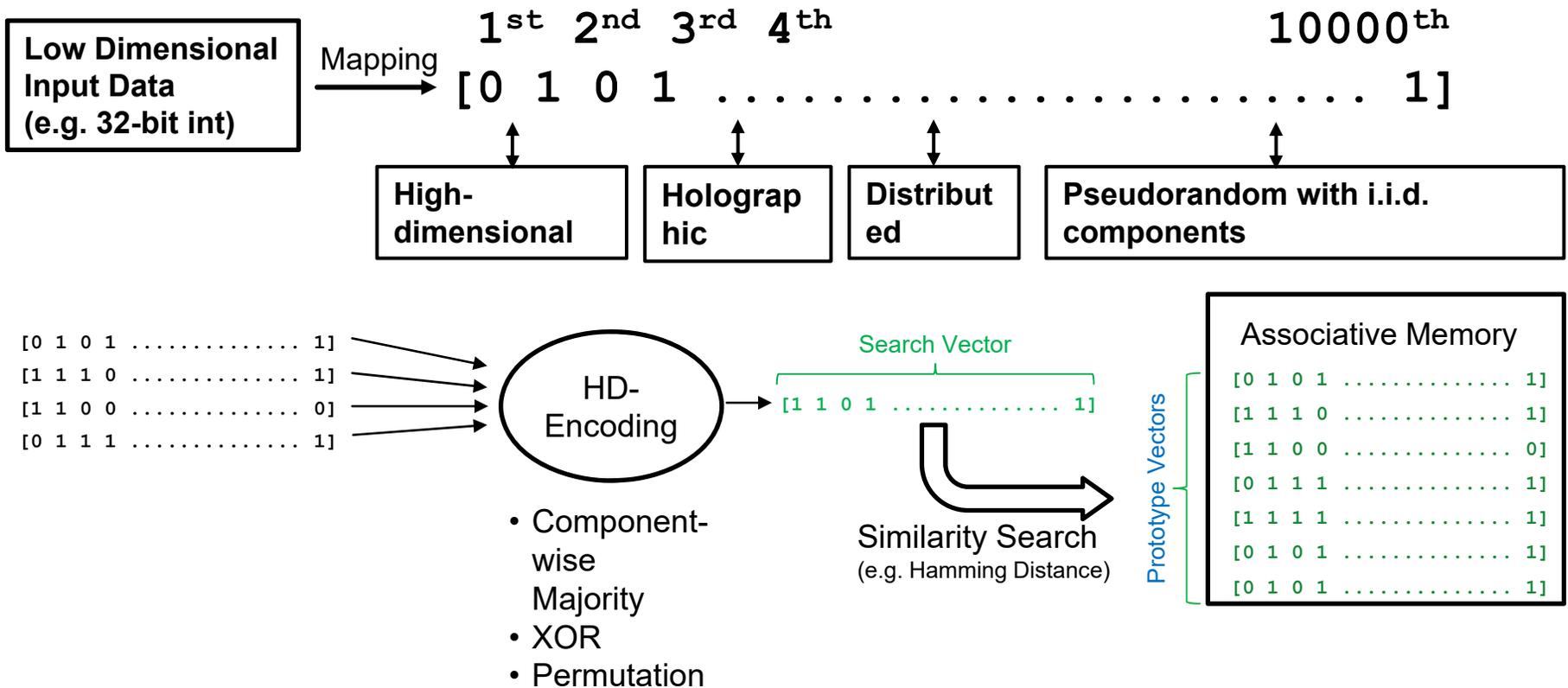
Target 30 fps ...

# Need uW always-on Intelligence

## Always-on IO Accelerator!



# Not Only CNNs: Hyper-Dimensional Computing

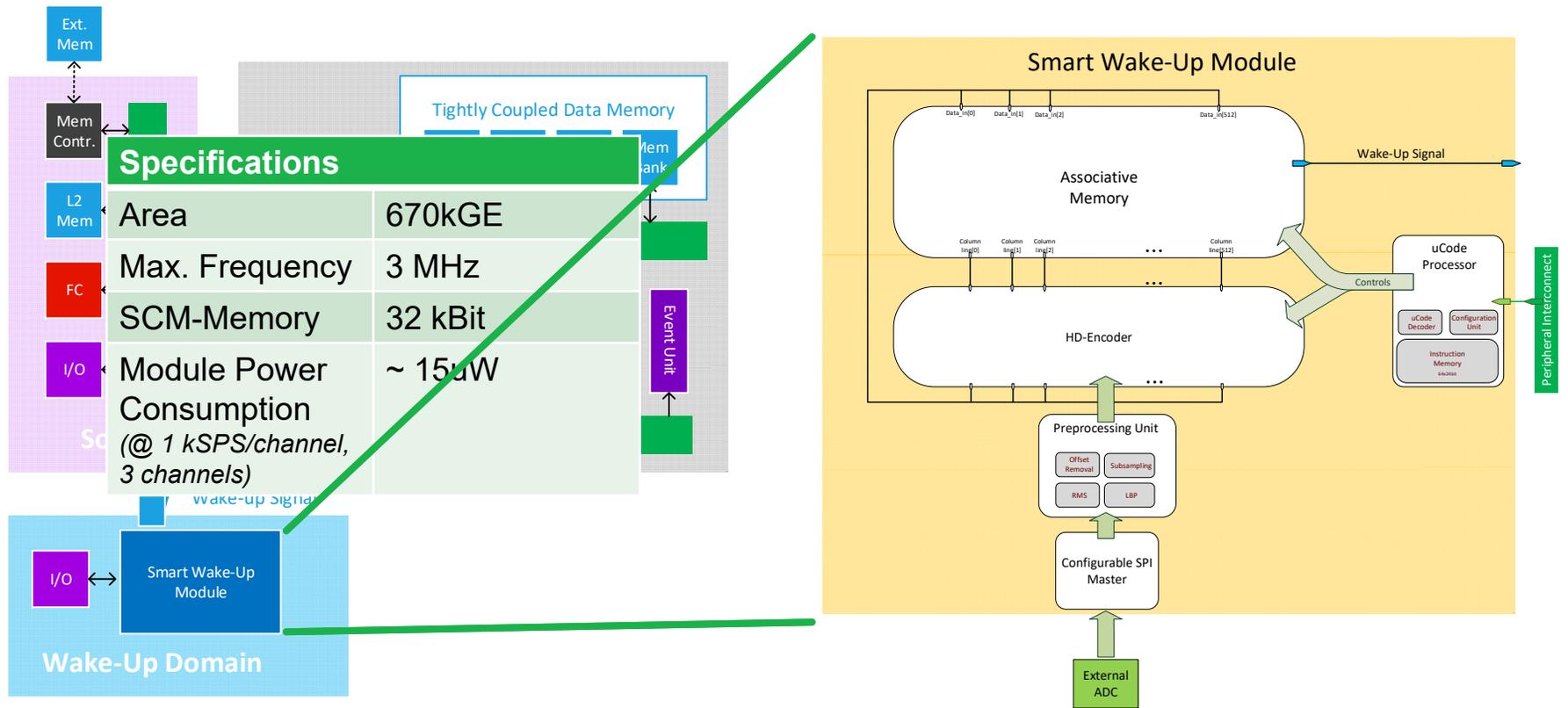


Highly parallel, fault-tolerant binary operators, assoc-min-distance search



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

# More efficiency (3): HD-Based smart Wake-Up Module

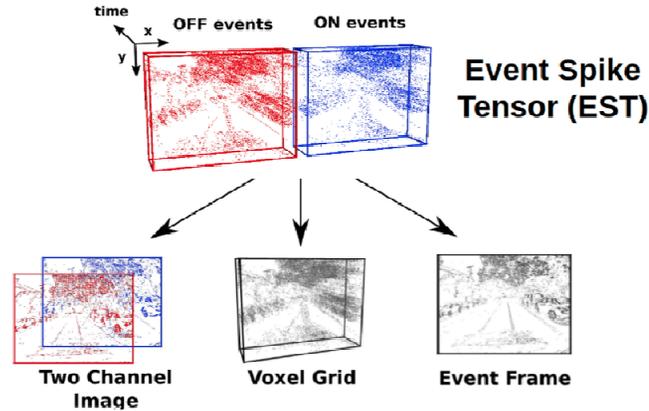


**Taped out in 22fdx**

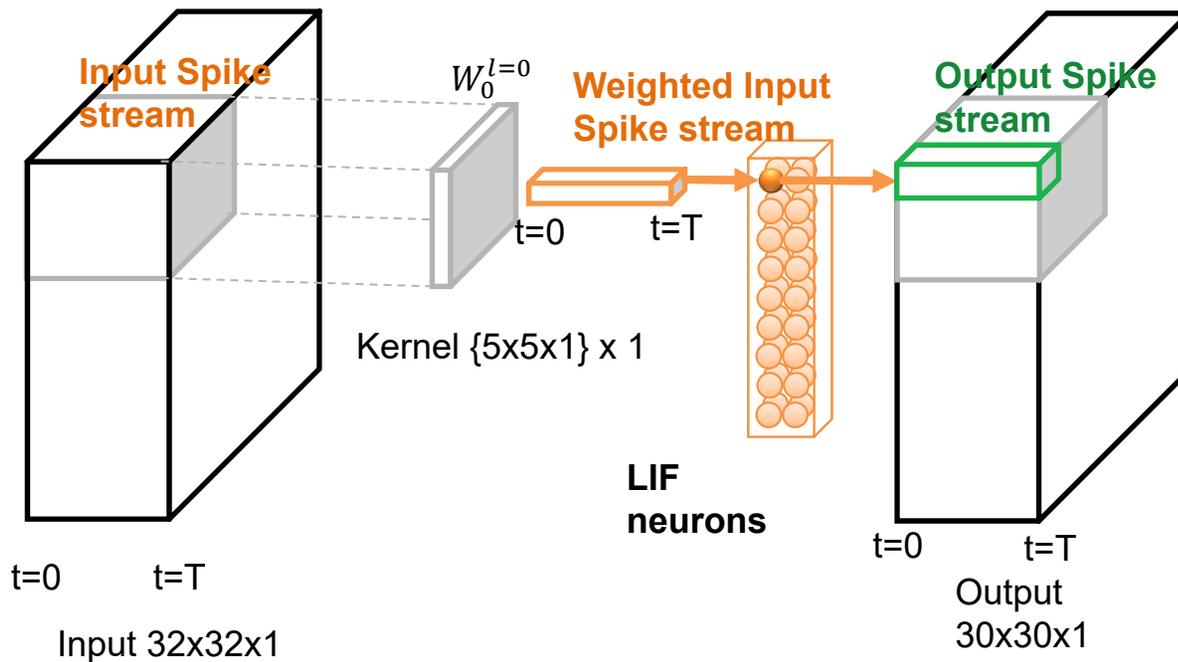
# Spiking Convolutional Neural Network



DVS camera

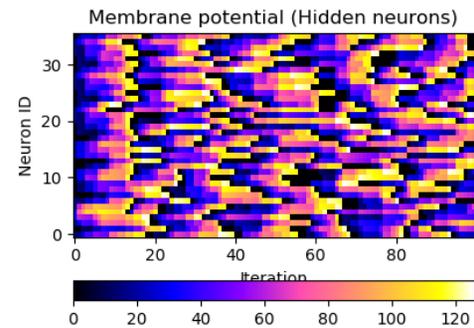
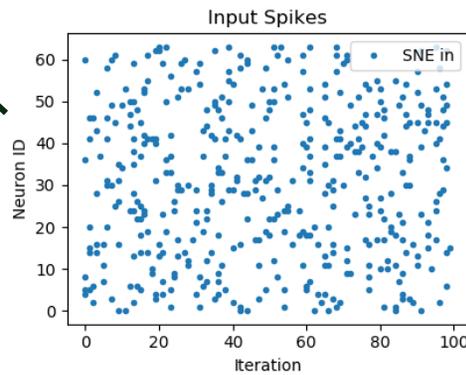


- Activity-driven computation
- Event-like (sparse) feature representation
- No processing in absence of input events
- Lightweight pre-processing required on emerging Event-based sensor data (e.g. DVS cameras)



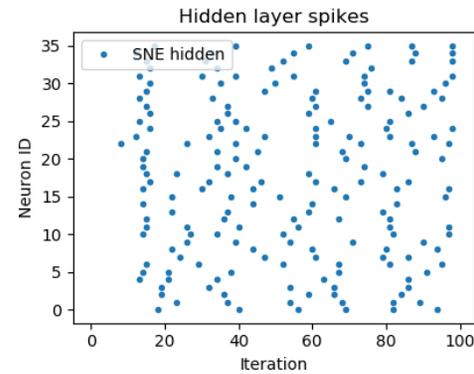
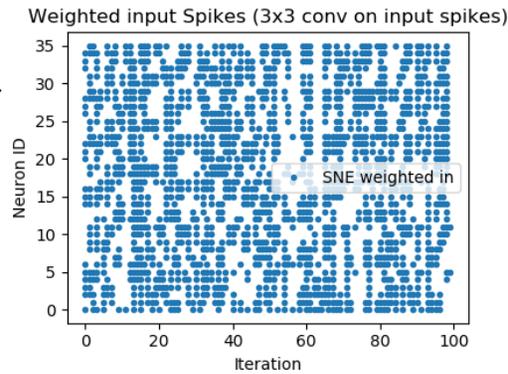
# Leaky Integrate&Fire CSNN layer

**Input spike stream**  
64 channel  
(8x8 matrix)  
100 time intervals



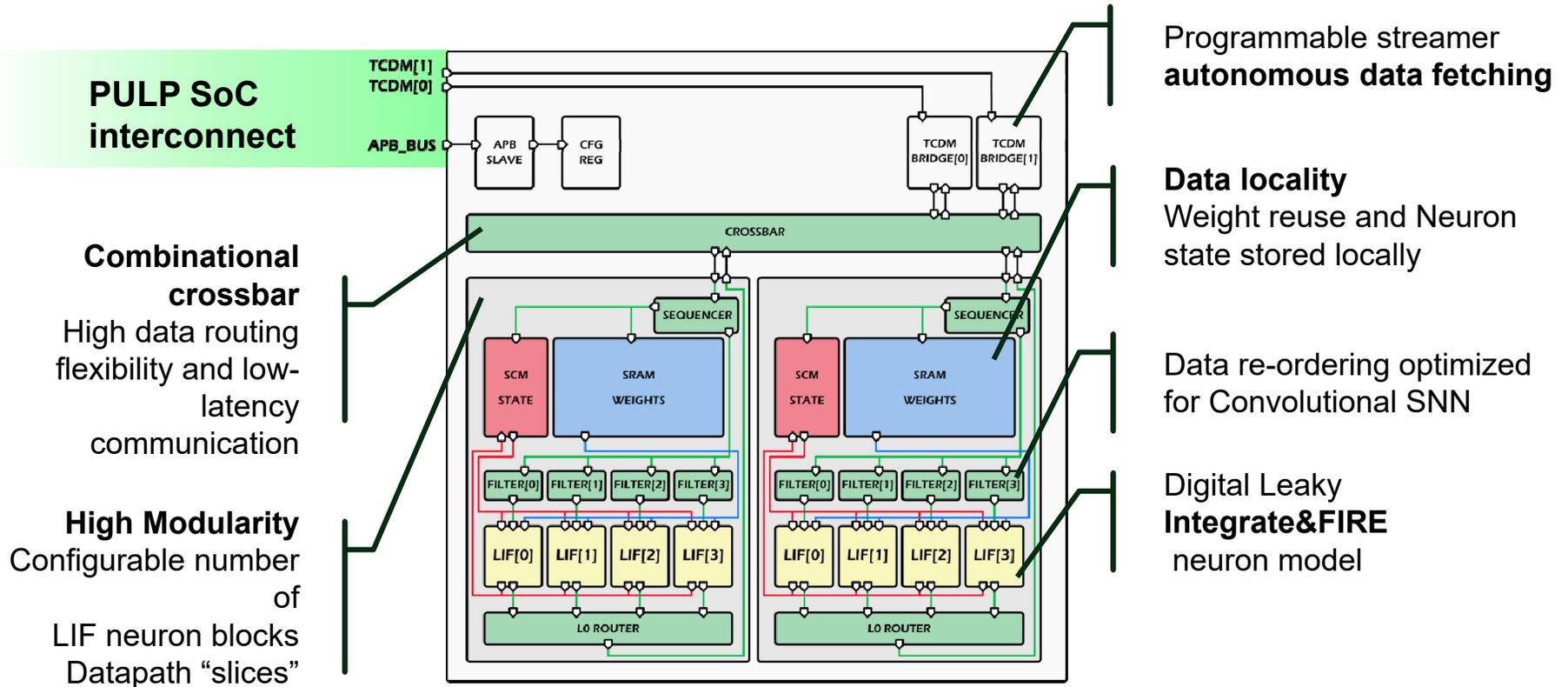
**Membrane potential**  
LIF state variable

**Weighted stream**  
Conv. (3x3 kernel)



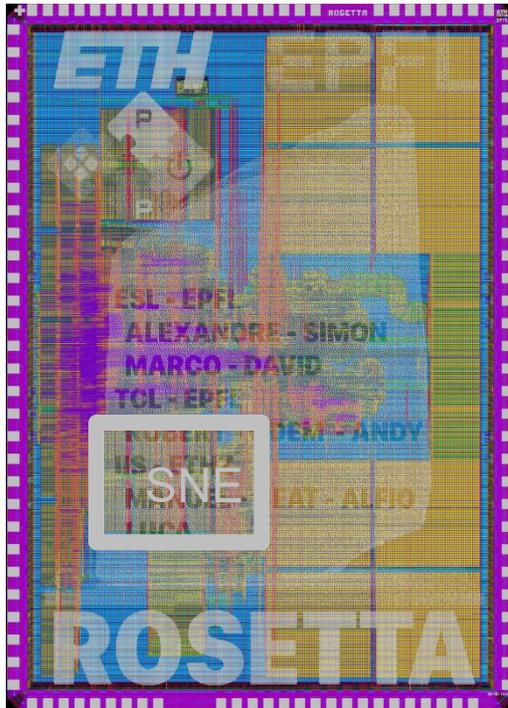
**Output spikes**  
When Membrane potential exceed the threshold

# SNE Accelerator Architecture



# First silicon implementation

## ROSETTA SoC



### SoC physical implementation

- TSMC65 nm technology
- Chip area  $4100\mu\text{m} \times 3000\mu\text{m}$
- Gates full SoC 6M

### Accelerator physical implementation

- Area Accelerator  $333\mu\text{m}^2$
- Gates Accelerator  $\sim 260\text{k}$

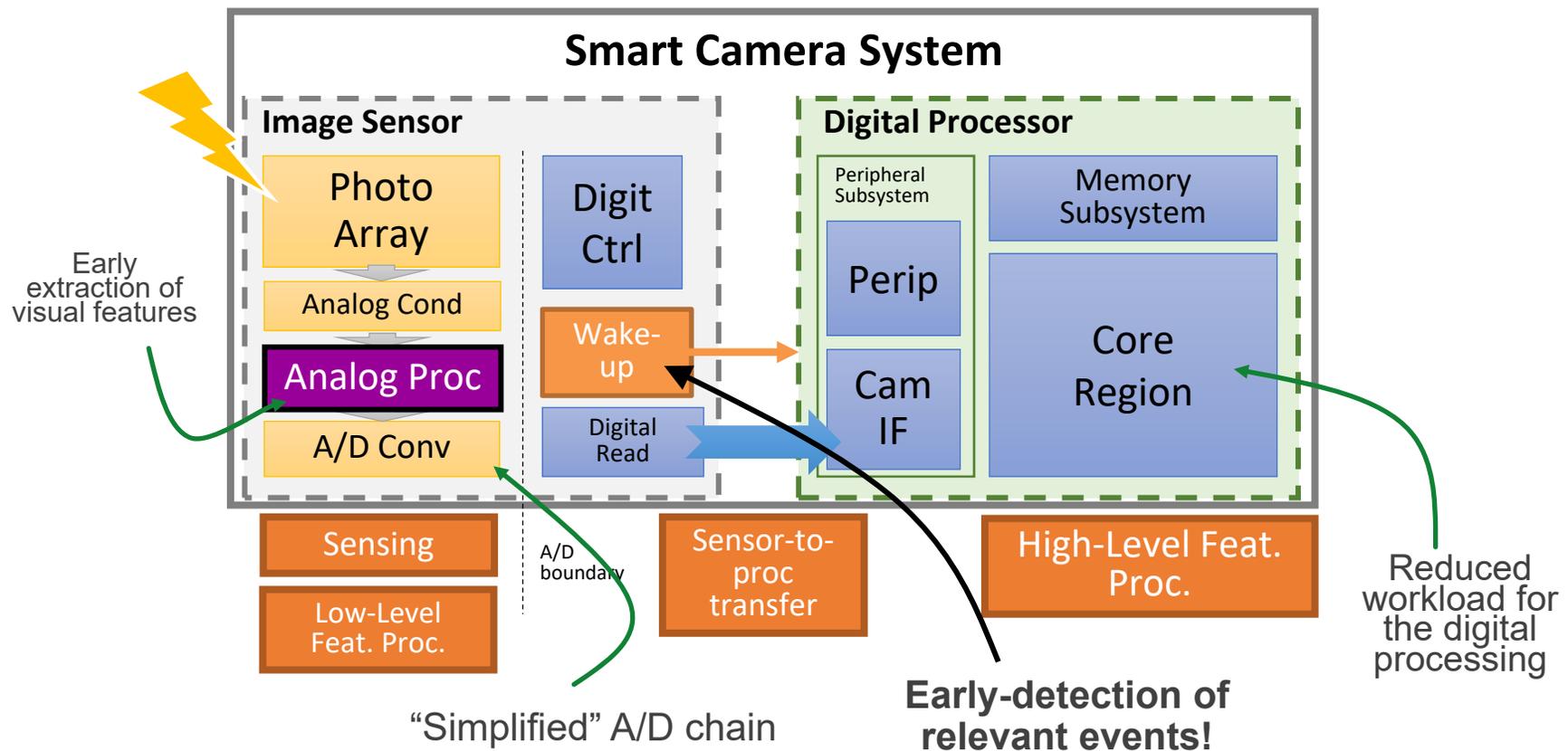
### Rosetta's SNE configuration

- 4x64 time domain multiplexed LIF neurons per slice
- 8kB per data-path slice
- 2 data-path slices

Number of Neurons	Memory (Accelerator)	Target Frequency	Performance (estimation)	Synaptic OP
512	16kB	250MHz	1.2 TOP/s	$\sim 250$ GSOP/s

# More Efficiency (4): Focal Plane Processing

Enable the extraction of low-level features in a parallel and efficient way by **integrating pixel-wise mixed-signal processing circuits** on the sensor die **to reduce the imager energy costs**.



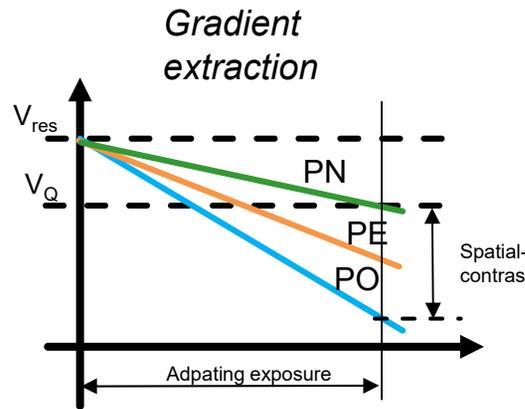
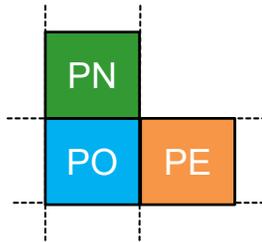
[Fernández-Berni, Jorge, et al. "Image Feature Extraction Acceleration." Image Feature Detectors and Descriptors. Springer International Publishing, 2016. 109-132.]

# Ultra-Low Power Imaging (GrainCam)

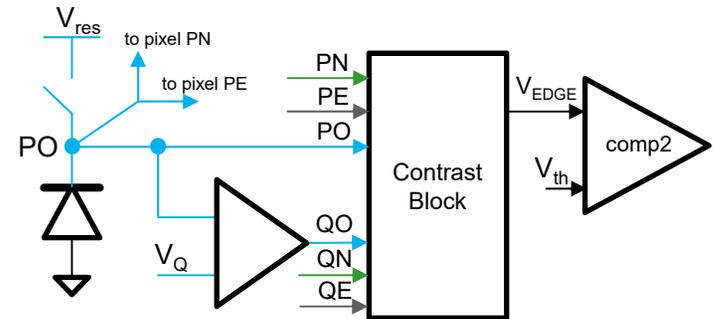
Imager performing **spatial filtering** and **binarization** on the sensor die through **mixed-signal sensing**!



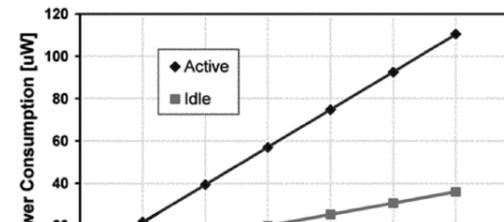
'Moving' pixel window



Per-pixel circuit for filtering and binarization



**Ultra-Low Power Consumption <100uW**

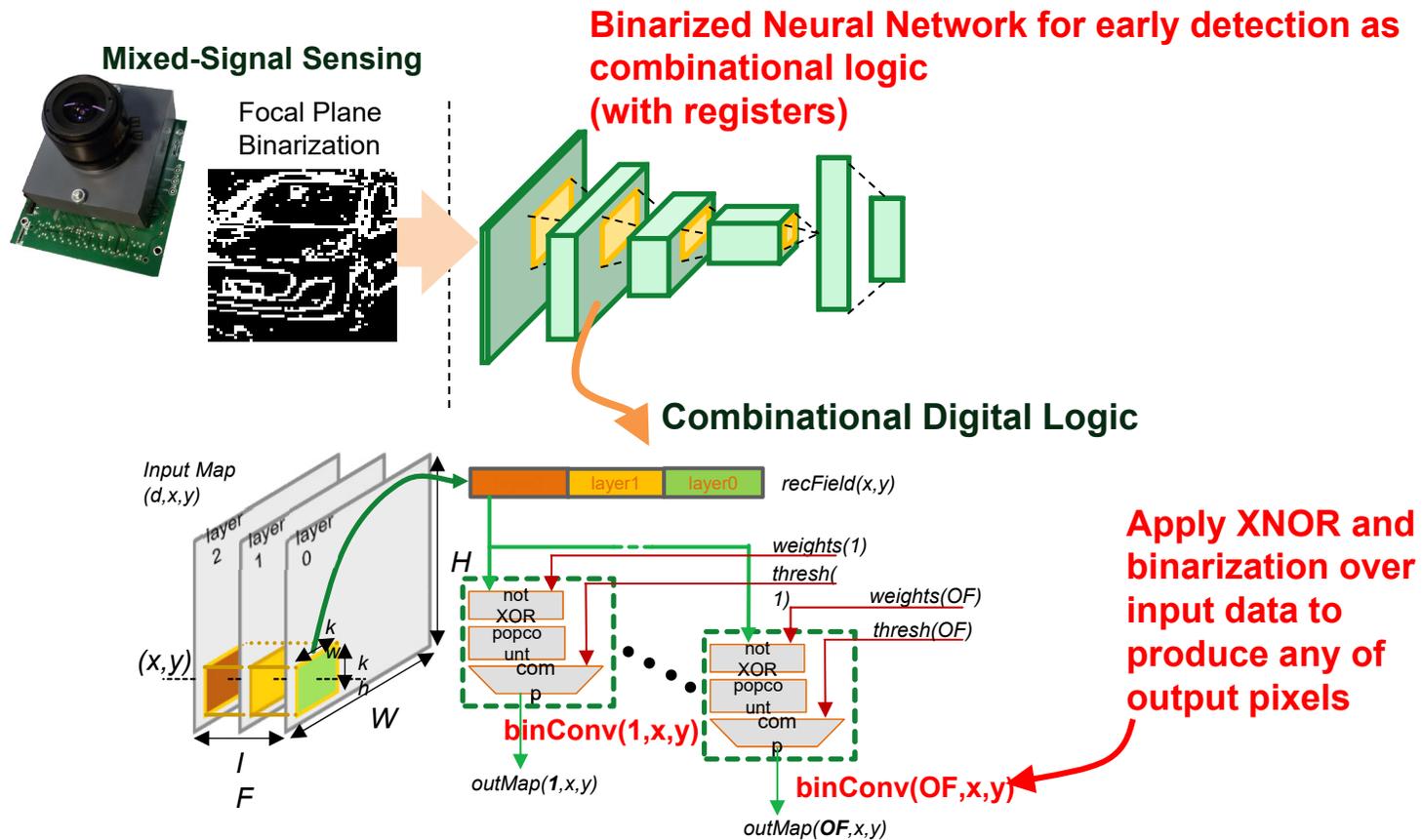


**<10x wrt SoA imagers**

This process naturally reflects the operation of a binarized pixel-wise convolution and can be seen as embedding the first convolutional layer within the image sensor die

M. Gottardi et al, "A 100uw 12864 pixels contrast-based asynchronous binary vision sensor for sensor networks applications," IEEE JSSC, 2009.

# Combinational “Fully Spatial” BNN



# Synthesis Results

Synthesis of both models with hard-wired or reconfigurable weights

GF 22nm SOI with LVT cells (typical corner case 0.65V, 25°C)

TABLE II  
SYNTHESIS AND POWER RESULTS FOR DIFFERENT CONFIGURATIONS

netw.	type	— area —		— time/img —		E/img [nJ]	leak. [μW]	E-eff. [TOP/J]
		[mm <sup>2</sup> ]	[MGE] <sup>†</sup>	[ns]	[FO4] <sup>‡</sup>			
16×16	var.	1.17	5.87	12.82	560	2.40	945	470.8
16×16	fixed	0.46	2.32	12.40	541	1.68	331	672.6
32×32	var.	5.80	29.14	17.27	754	11.14	4810	479.4
32×32	fixed	2.61	13.13	21.02	918	11.67	1830	457.6

<sup>†</sup> Two-input NAND-gate size equivalent: 1 GE = 0.199 μm<sup>2</sup>

<sup>‡</sup> Fanout-4 delay: 1 FO4 = 22.89 ps

Hundreds of TOPS/W!

Massive area reduction when hard-wiring the weights:

- XNOR operations reduce to wires or inverter, which can be also shared among different receptive fields
- popcounts also exploits sharing mechanisms

**Advanced Synthesis Tools become central to exploit weights and intermediate results sharing to reduce the area occupation**

M Rusci, L Cavigelli, L Benini “Design automation for binarized neural networks: A quantum leap Opportunity?”  
2018 IEEE International Symposium on Circuits and Systems (ISCAS), 1-5

# Conclusion

- Near-sensor processing → Energy efficiency pJ/OP and below
  - Ultra-low power architecture and circuits are needed
  - Memory is THE challenge
- TinyML: Inference can be squeezed into mW envelope
  - Non-von-Neumann acceleration → remove lmem bottleneck
  - Very robust at low precision → memory footprint reduction
  - fJ/OP is in sight! (100+TFLOPs/W) → **mW inference engines!**
- Pushing on the memory+IO bottleneck
  - TCDM + SCM memory hierarchy optimization and logic+physical optimization
  - In-place, in memory, staged, event-based, non-DNN inference
  - *Better memories and memory interfaces (NVM, HBM...)*
  - *Multi-chip Systolic (2.5D chiplets)*
  - *Fature map compression*

} Not covered

**Open-Source Innovation ecosystem!**





*The fun is  
just beginning*

 <http://pulp-platform.org>

# Closing the Bin/Ternarization accuracy Gap

Table 1: Experimental Results on ImageNet

Model	Method*		Levels <sup>†</sup>	Accuracy [%] (top-1/top-5)
ResNet-18	baseline	torchvision v0.4.0	full-prec.	69.76/89.08
ResNet-18	QN	(Yang et al., 2019)	5: $\{\alpha_i\}_i$	69.90/89.30
ResNet-18	ADMM	(Leng et al., 2018)	5: $\{0\} \cup \{\pm 2^i\}_i$	67.50/87.90
ResNet-18	LQ-Nets	(Zhang et al., 2018)	4: $\{\pm \alpha_i\}_i$	68.00/88.00
ResNet-18	QN	(Yang et al., 2019)	3: $\{\alpha_1, \alpha_2, \alpha_3\}$	69.10/88.90
ResNet-18+ <sup>‡</sup>	TTQ	(Zhu et al., 2017)	3: $\{\alpha_1, 0, \alpha_2\}$	66.60/87.20
ResNet-18	ADMM	(Leng et al., 2018)	3: $\{-1, 0, 1\}$	67.00/88.00
ResNet-18	INQ	(Zhou et al., 2017)	3: $\{-1, 0, 1\}$	66.00/88.00
ResNet-18+ <sup>‡</sup>	TWN	(Li et al., 2016)	3: $\{-1, 0, 1\}$	65.30/86.20
ResNet-18	TWN	(Li et al., 2016)	3: $\{-1, 0, 1\}$	61.80/84.20
ResNet-18	<b>RPR (ours)</b>		3: $\{-1, 0, 1\}$	<b>66.31/87.84</b>
ResNet-18	ADMM	(Leng et al., 2018)	2: $\{-1, 1\}$	64.80/86.20
ResNet-18	XNOR-net BWN	(Rastegari et al., 2016)	2: $\{-1, 1\}$	60.80/83.00
ResNet-18	<b>RPR (ours)</b>		2: $\{-1, 1\}$	<b>64.62/86.01</b>
ResNet-50	baseline	torchvision v0.4.0	full-prec.	76.15/92.87
ResNet-50	ADMM	(Leng et al., 2018)	3: $\{-1, 0, 1\}$	72.50/90.70
ResNet-50	TWN	(Li et al., 2016)	3: $\{-1, 0, 1\}$	65.60/86.50
ResNet-50	<b>RPR (ours)</b>		3: $\{-1, 0, 1\}$	<b>71.83/90.28</b>
ResNet-50	ADMM	(Leng et al., 2018)	2: $\{-1, 1\}$	68.70/88.60
ResNet-50	XNOR-net BWN	(Rastegari et al., 2016)	2: $\{-1, 1\}$	63.90/85.10
ResNet-50	<b>RPR (ours)</b>		2: $\{-1, 1\}$	<b>65.14/86.31</b>
GoogLeNet	baseline	torchvision v0.4.0	full-prec.	69.78/89.53
GoogLeNet	ADMM	(Leng et al., 2018)	3: $\{-1, 0, 1\}$	63.10/85.40
GoogLeNet	TWN	(Li et al., 2016)	3: $\{-1, 0, 1\}$	61.20/84.10
GoogLeNet	<b>RPR (ours)</b>		3: $\{-1, 0, 1\}$	<b>64.88/86.05</b>
GoogLeNet	ADMM	(Leng et al., 2018)	2: $\{-1, 1\}$	60.30/83.20
GoogLeNet	XNOR-net BWN	(Rastegari et al., 2016)	2: $\{-1, 1\}$	59.00/82.40
GoogLeNet	<b>RPR (ours)</b>		2: $\{-1, 1\}$	<b>62.01/84.83</b>

Soa results (Sept19)

1. First and last layer FP
2. ResNets have type-B bypasses (with 1x1 conv. In the non residual paths)
3. Modified network 2.25x more weights