PULP
Parallel Ultra Low Power

Extreme Edge AI on Open Hardware

ACCML@HIPEAC Bologna *20.01.2020*

Luca Benini[1,2]

[1]Department of Electrical, Electronic and Information Engineering

ETH zürich

[2]Integrated Systems Laboratory

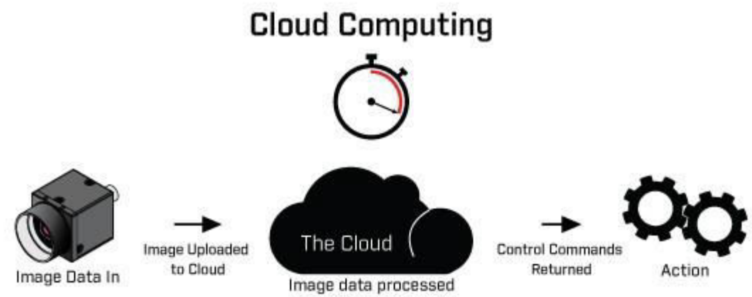
erc
European Research Council

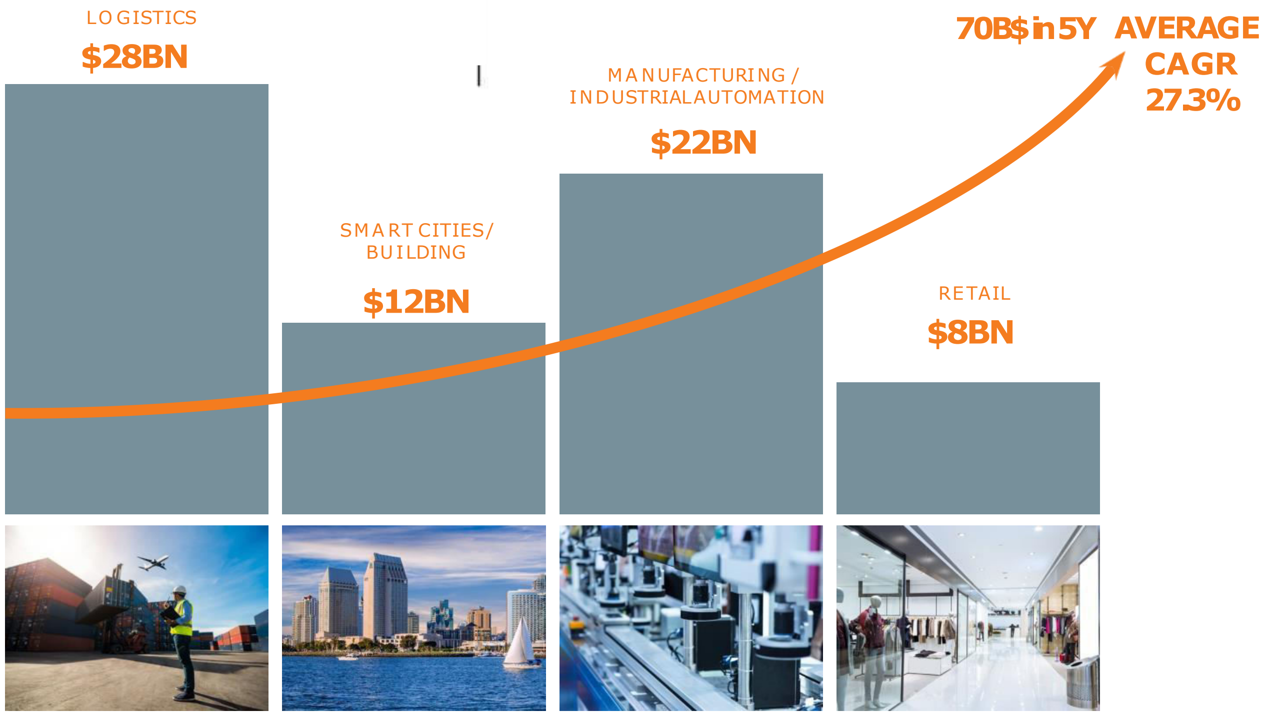
European Commission
Horizon 2020
European Union funding
for Research & Innovation

nano-tera.ch

FNSNF
Fonds National Suisse
Schweizerischer Nationalfonds
Fondo Nazionale Svizzero
Swiss National Science Foundation

Cloud → Edge → Extreme Edge AI aka TinyML





#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?*

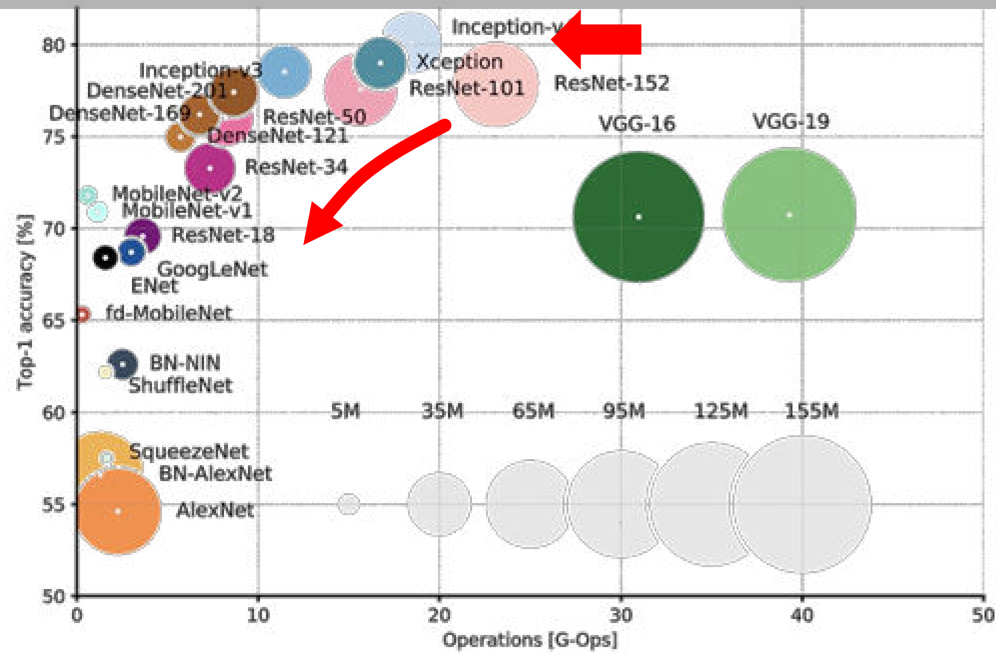
Source: www.amazon.com/ask/questions/asin/B01M3VHG87/

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

THE SILENT INTELLIGENCE E. Gousev, Qcomm research

PULP

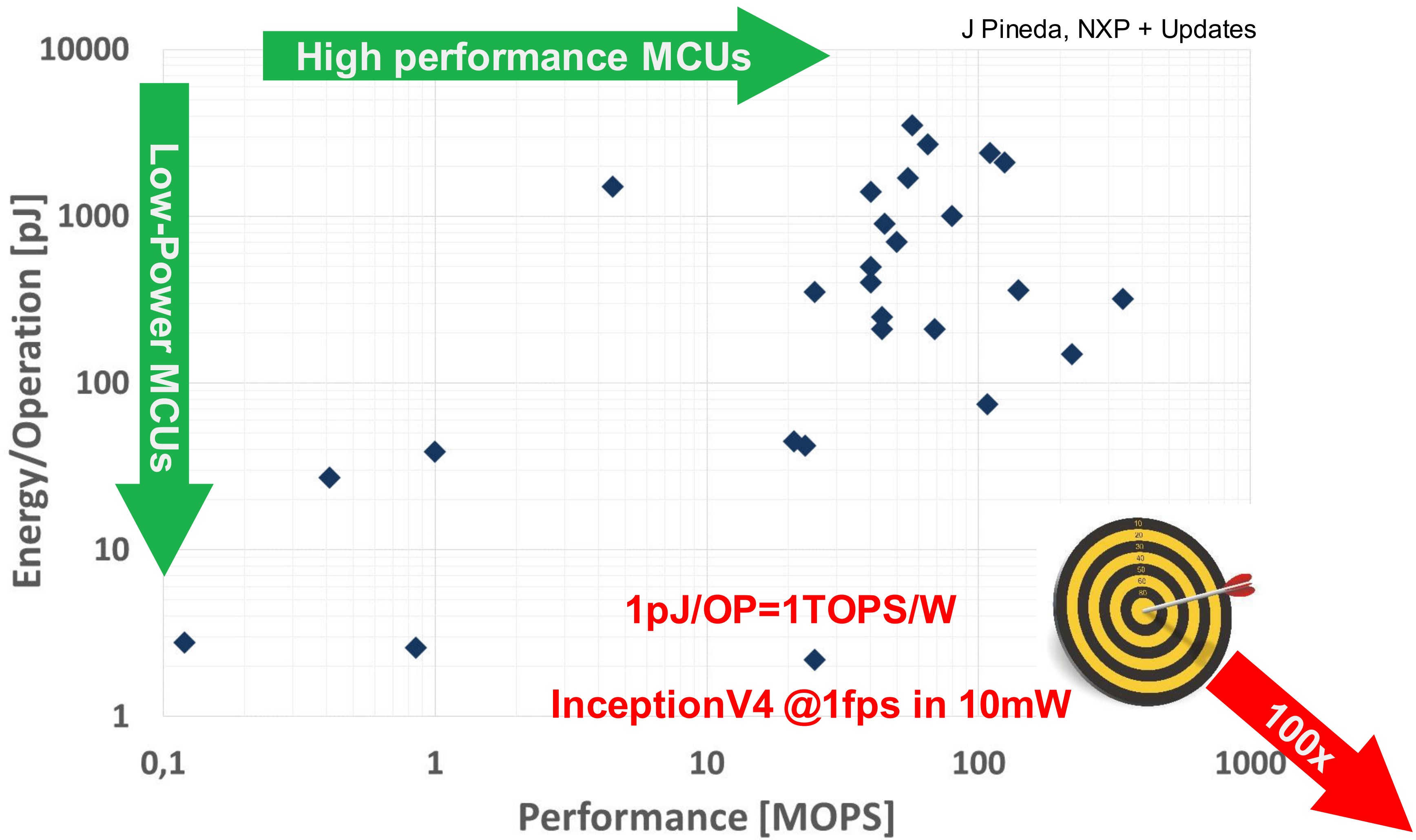
AI Workloads from Cloud to Edge (Extreme?)



GOP+
MB+

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

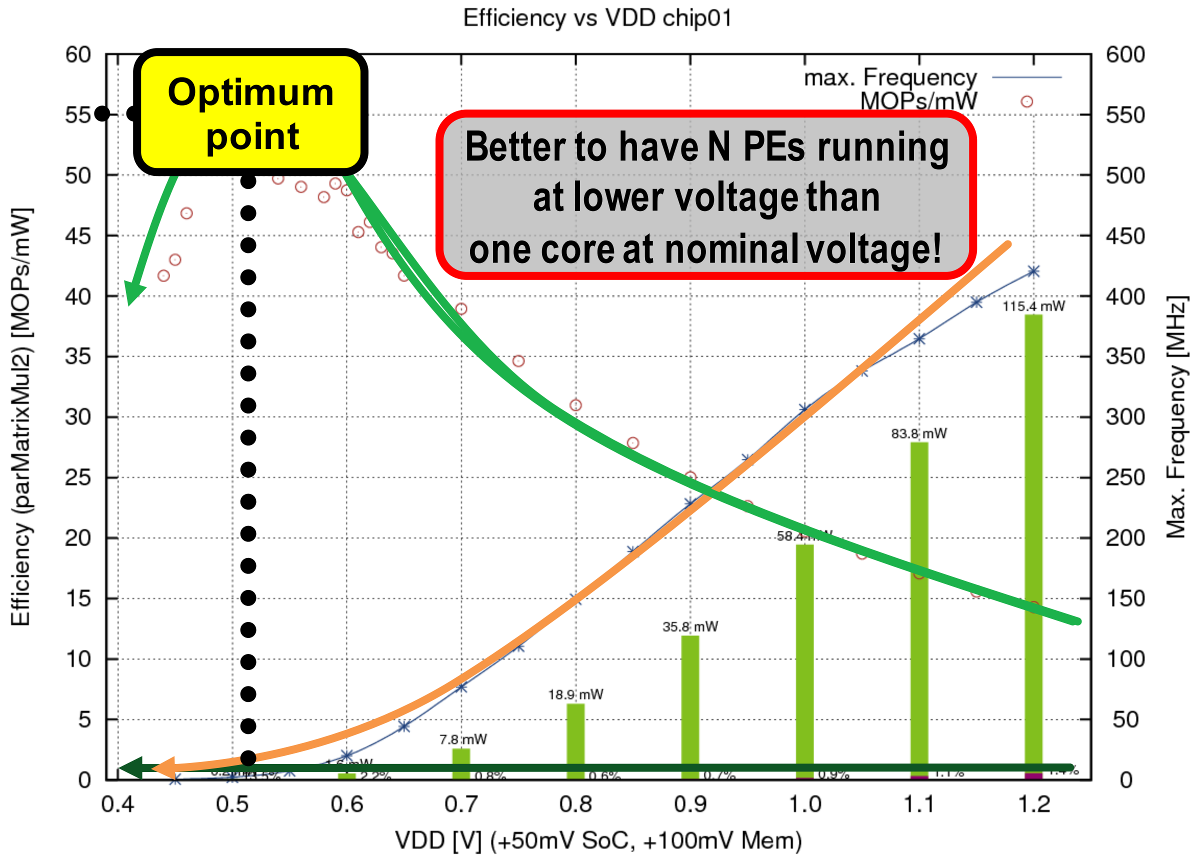
Energy efficiency is THE Challenge



Cool… But, HOW??

ML & Near-threshold: a Marriage Made in Heaven

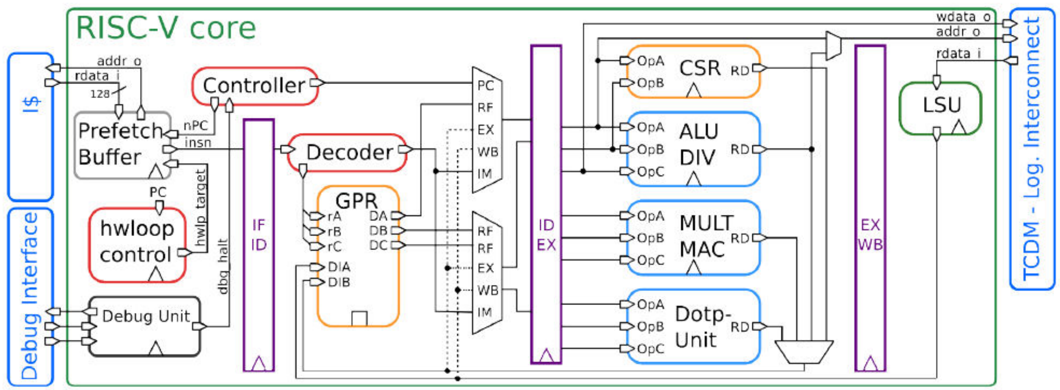
- As VDD decreases, operating speed decreases
- However efficiency increases→ more work done per Joule
- Until leakage effects start to dominate
- Put more units in parallel to get performance up and keep them busy with a parallel workload



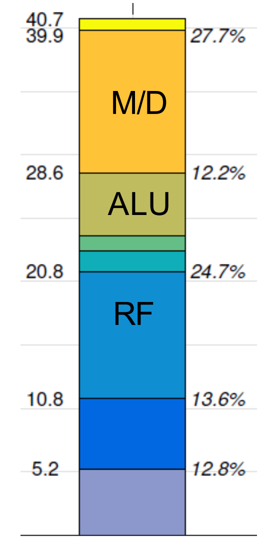
ML is massively parallel and scales well (P/S ↑ with NN size)

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

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



40KG
70% RF+DP

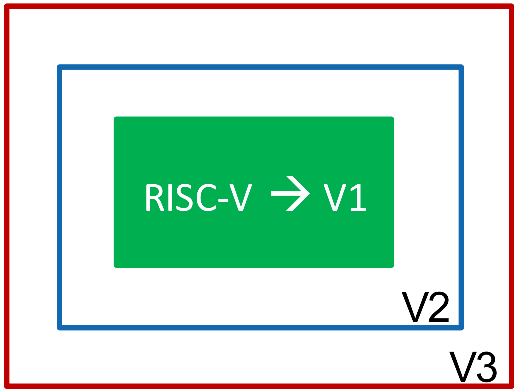


RISC-V® ISA is extensible *by construction* (great!)

V1 Baseline RISC-V RV32IMC

V2 HW loops
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

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
```

N/4

RV32IMCXpulp

```
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
```

HW Loop

LD/ST with post increment

8-bit SIMD sdotp

9x less instructions than RV32IMC

Pooling & ReLu

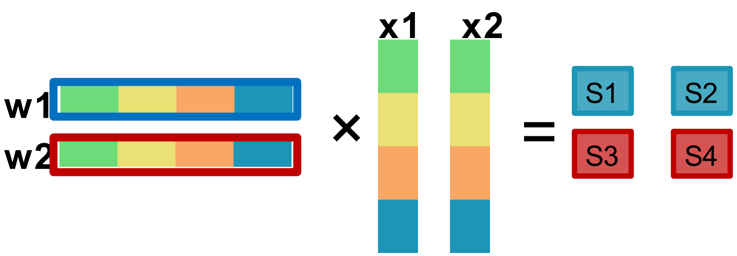
- HW loop
- LD/ST with post-increment
- 8-bit SIMD max, avg INSNS

PULP

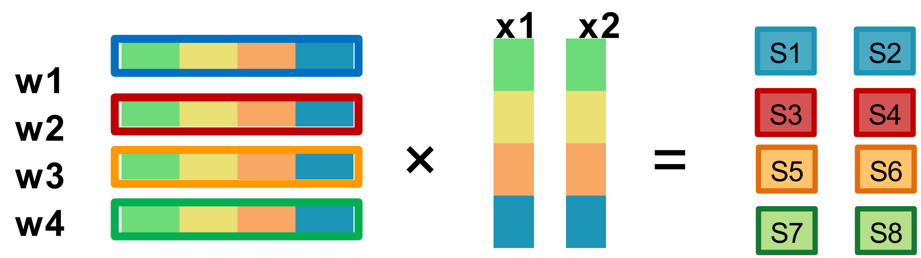
PULP-NN: Data Reuse in the Register File

8-bit Convolution

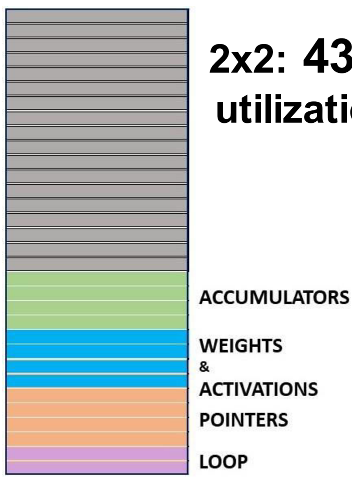
CMSIS-NN based Matrix Multiplication Layout: 2x2



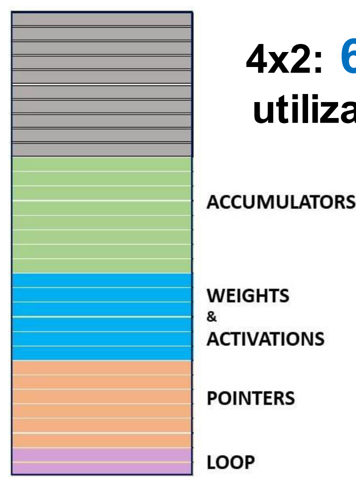
PULP-NN Matrix Multiplication Layout: 4x2



RegisterFile of the RI5CY core: 32 general purpose registers



2x2: 43% utilization



4x2: 69% utilization

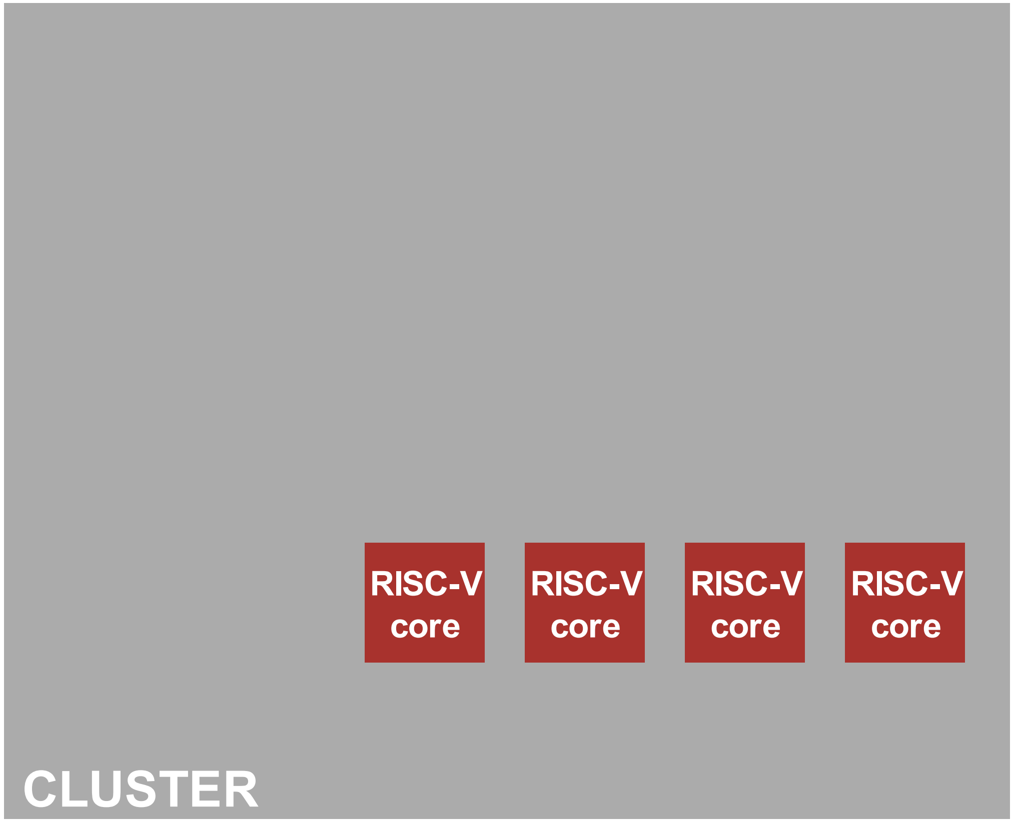
More Data Reuse
&
Higher utilization of the RF

Peak Performance (8 cores)

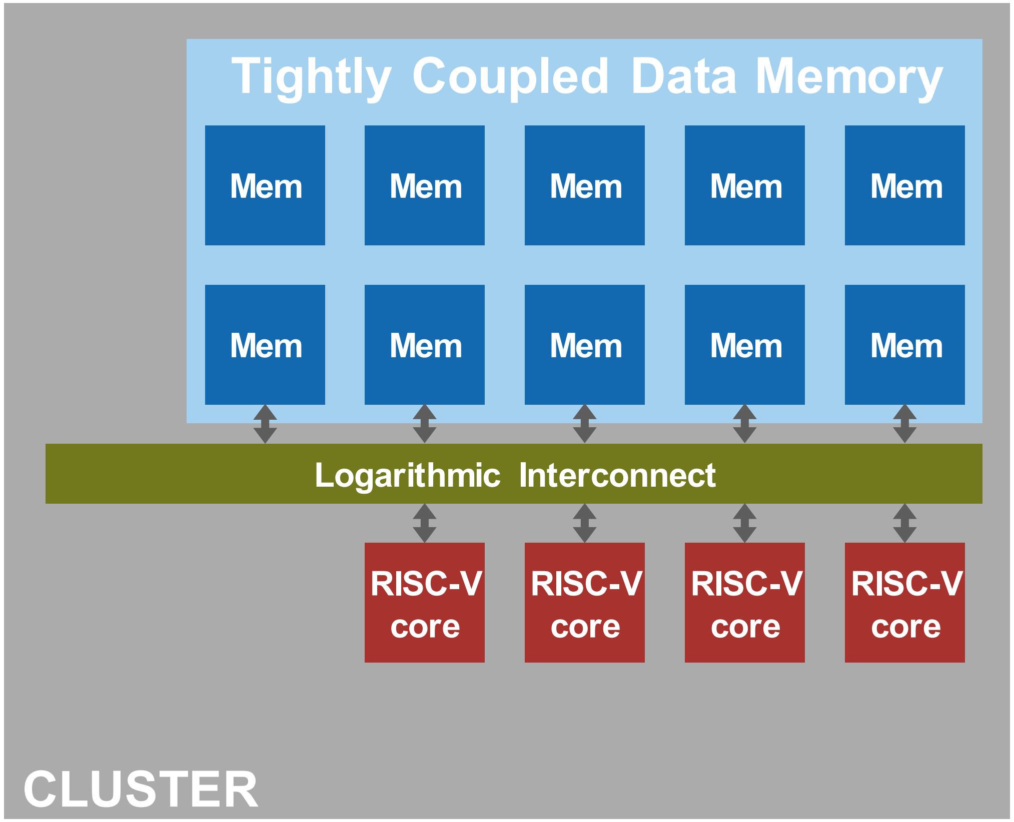
2x2 : 12.8 MAC/cyc

4x2 : **15.5 MAC/cyc**

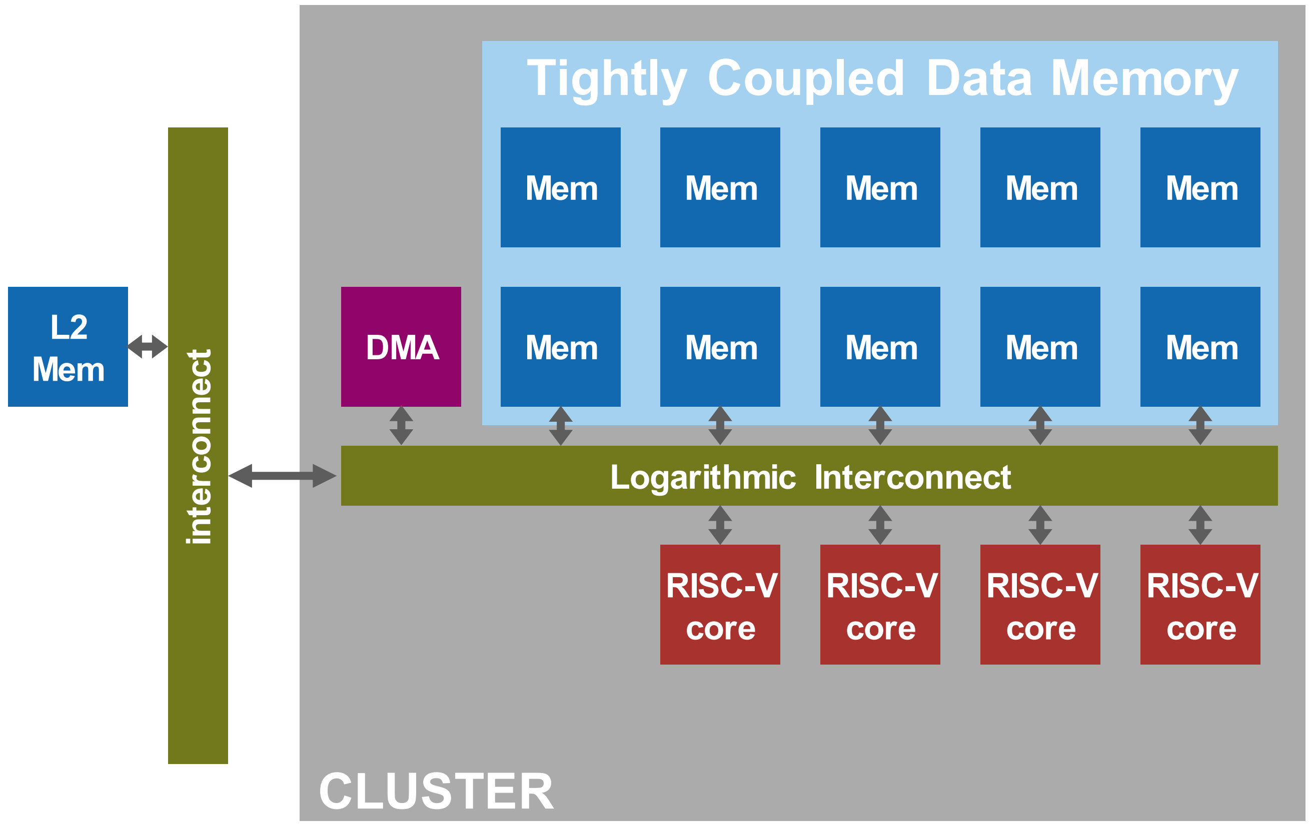
Multiple RI5CY Cores (1-16)



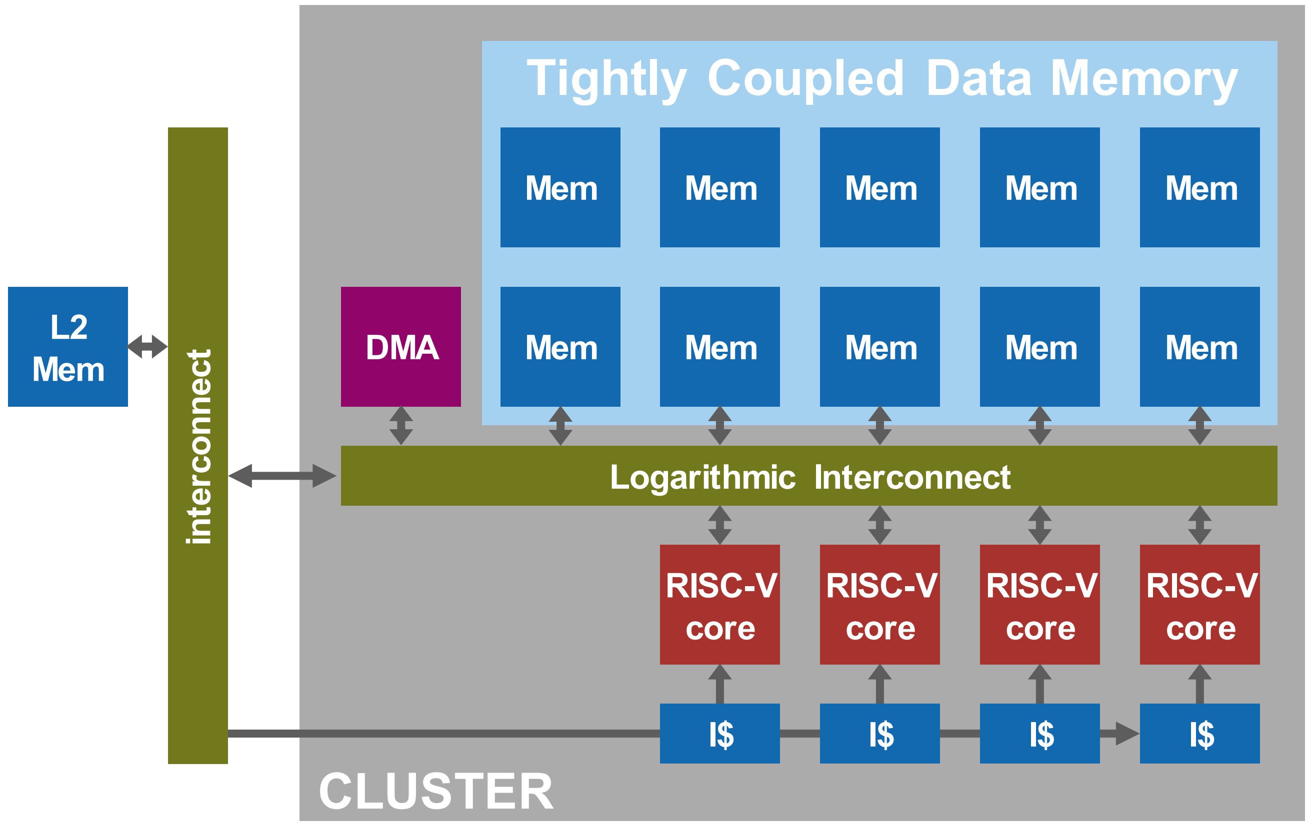
Low-Latency Shared TCDM



DMA for data transfers from/to L2



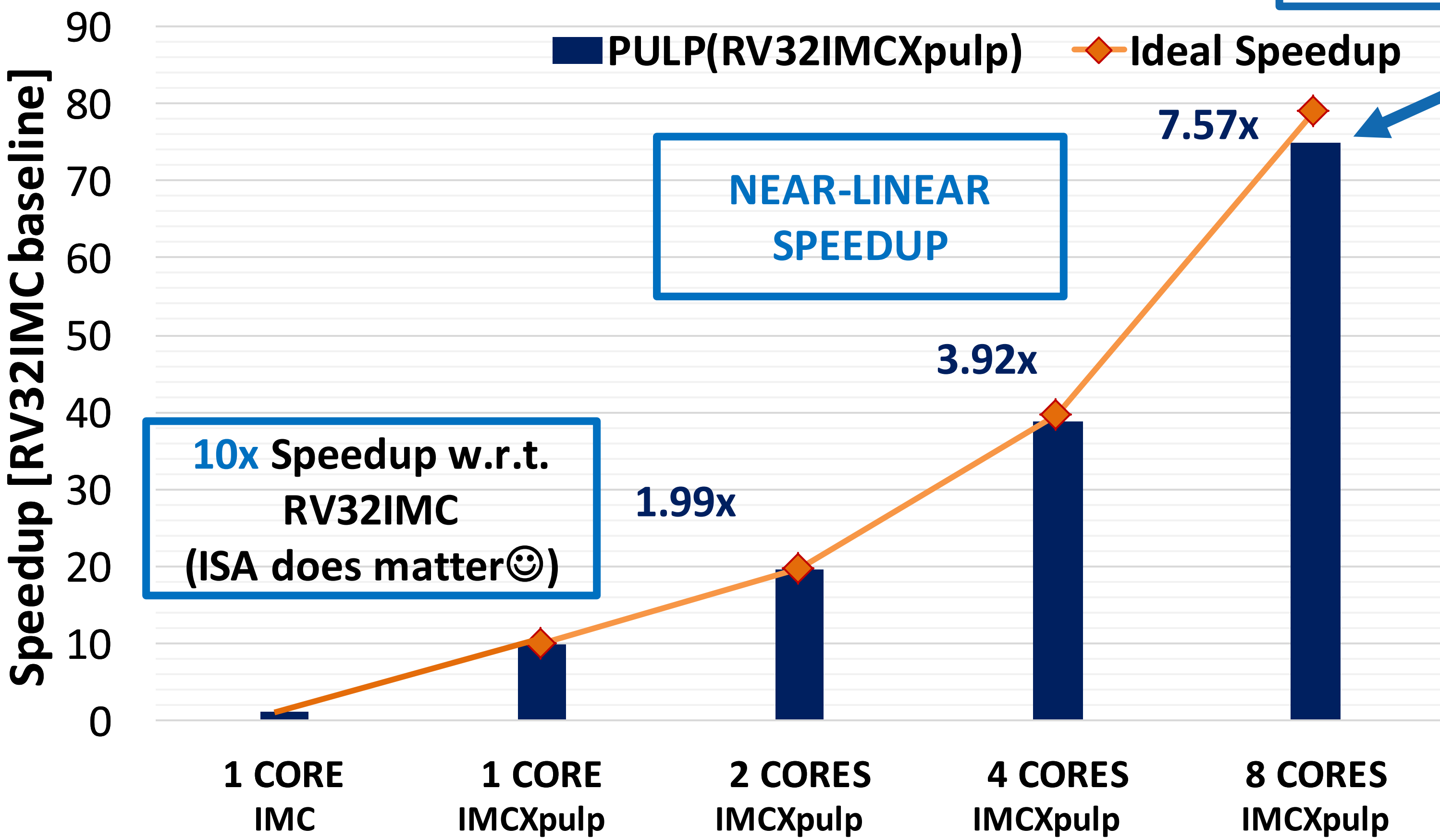
Shared Icached with private “loop buffer”



Results: RV32IMCXpulp vs RV32IMC

8-bit Convolution Results

Overall Speedup of **75x**

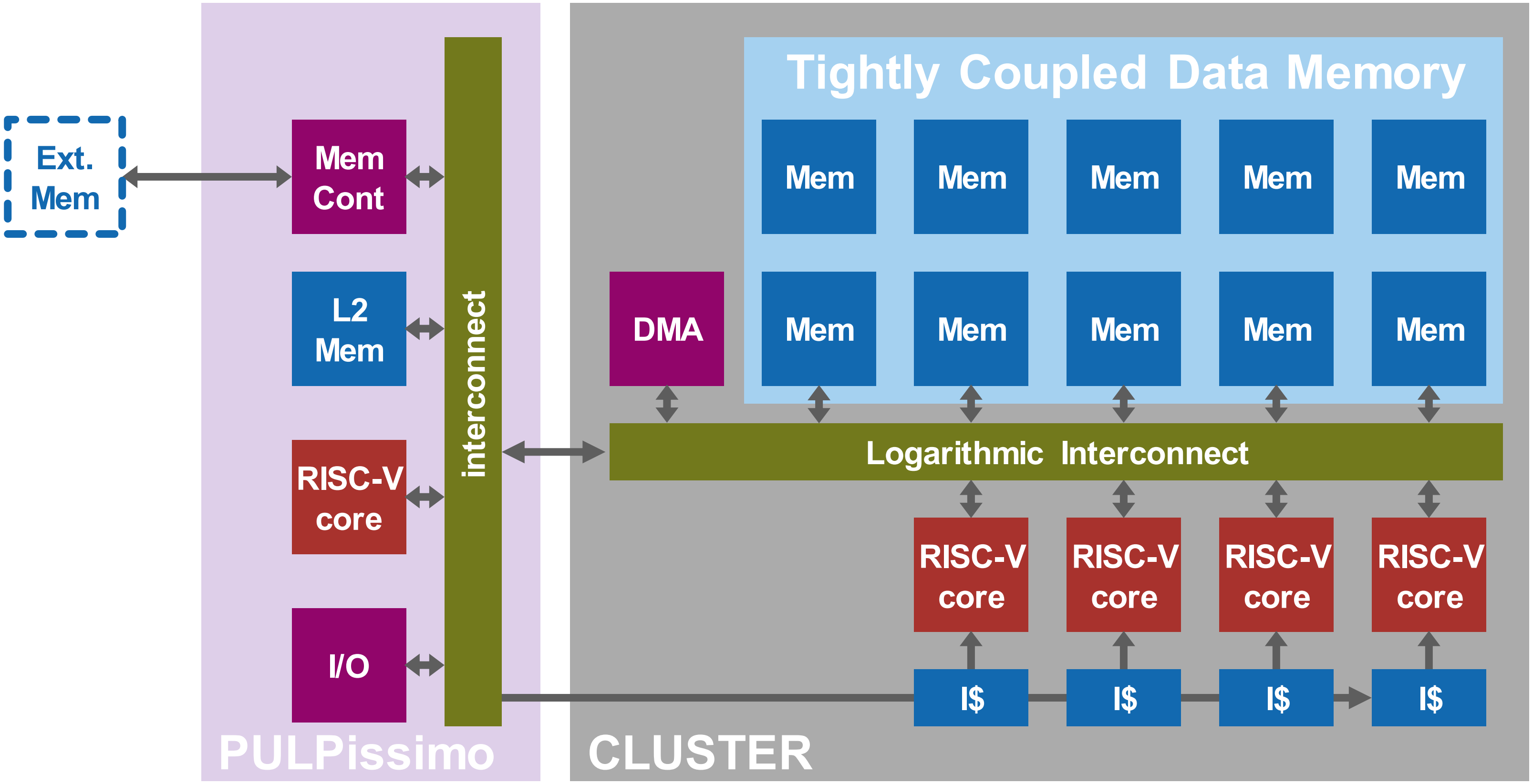


3pJ/OP in fdx22

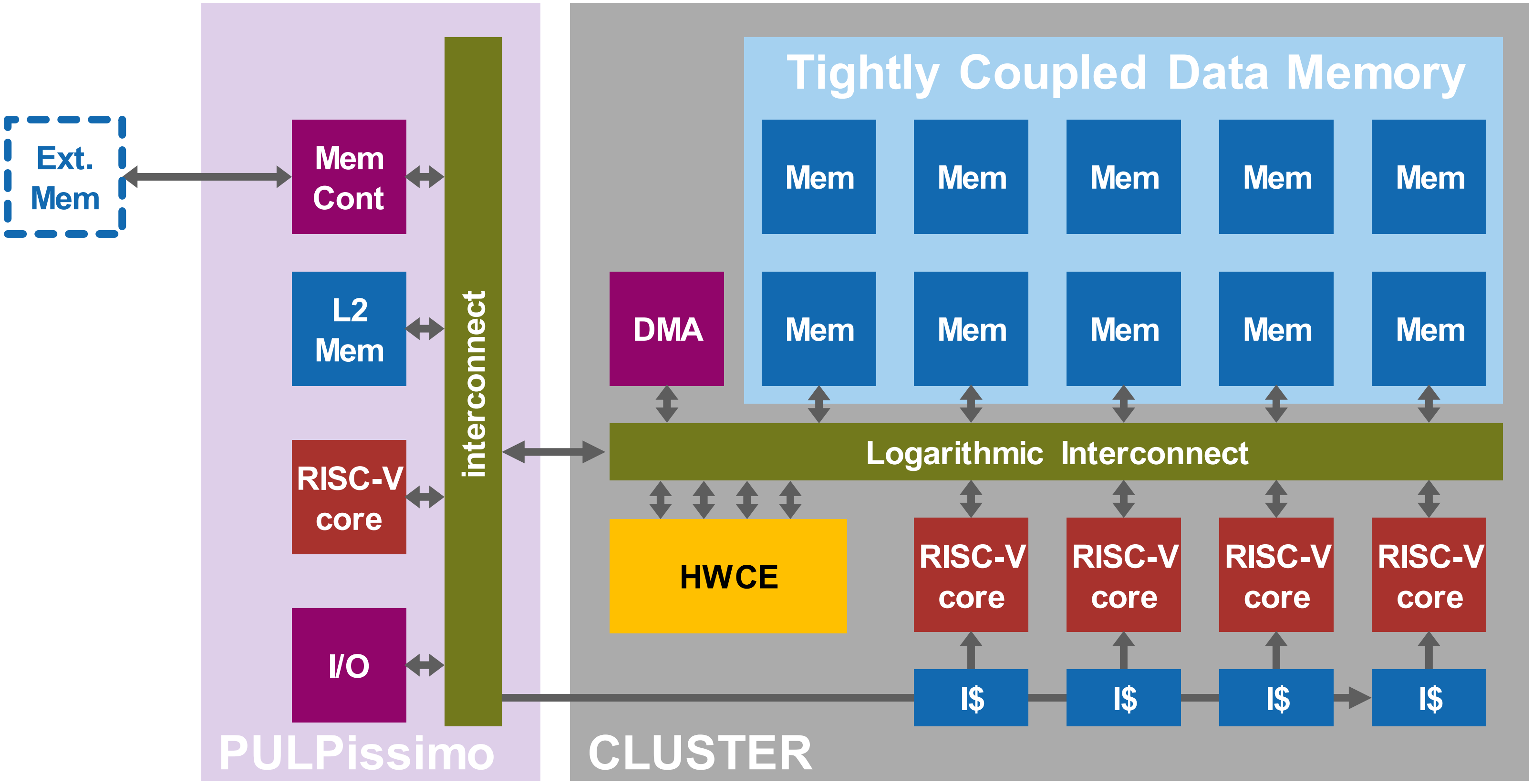
PULP-NN: an open Source library for DNN inference on PULP cores

PULP

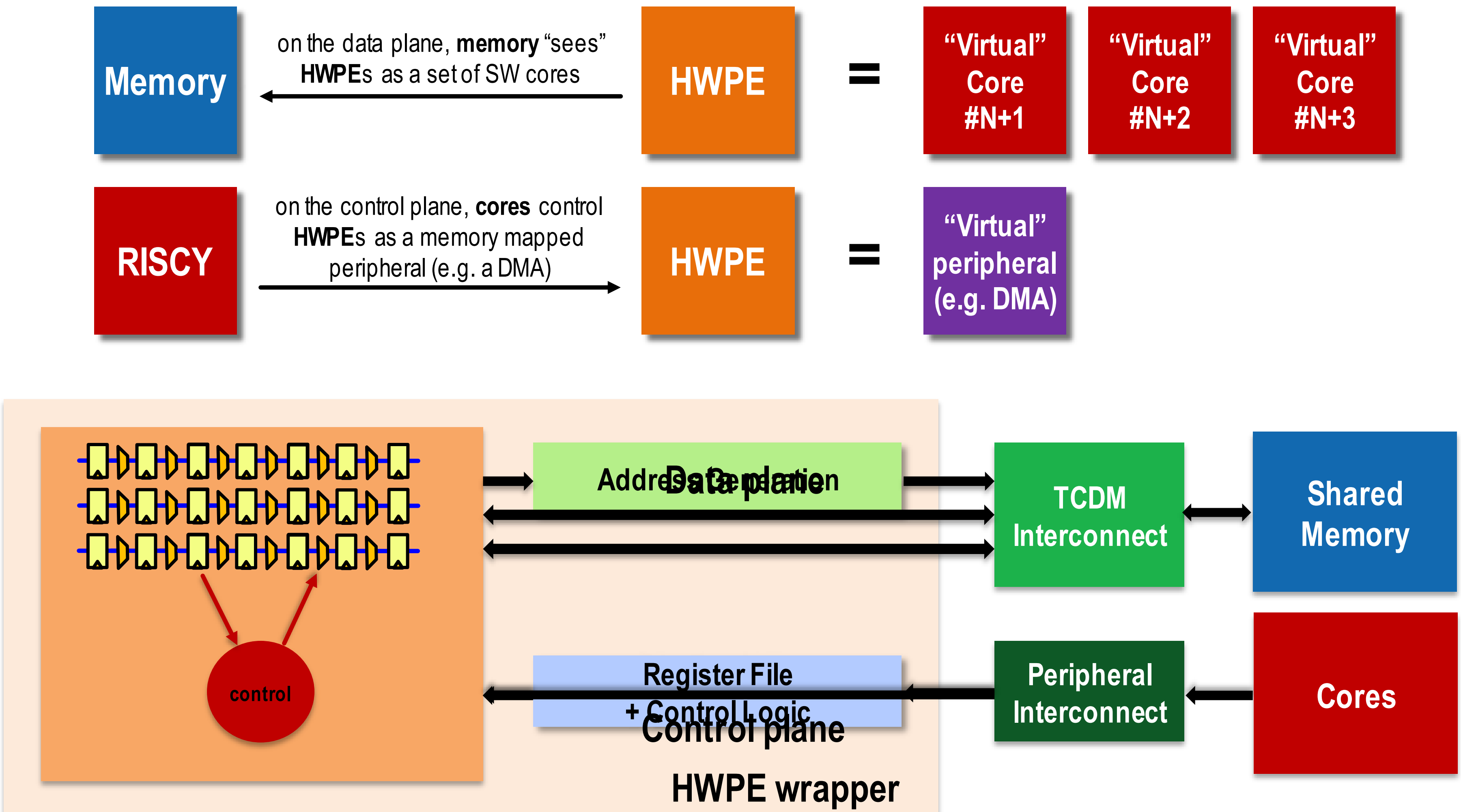
An additional controller is used for I/O



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



Hardware Processing Engines (HWPEs)

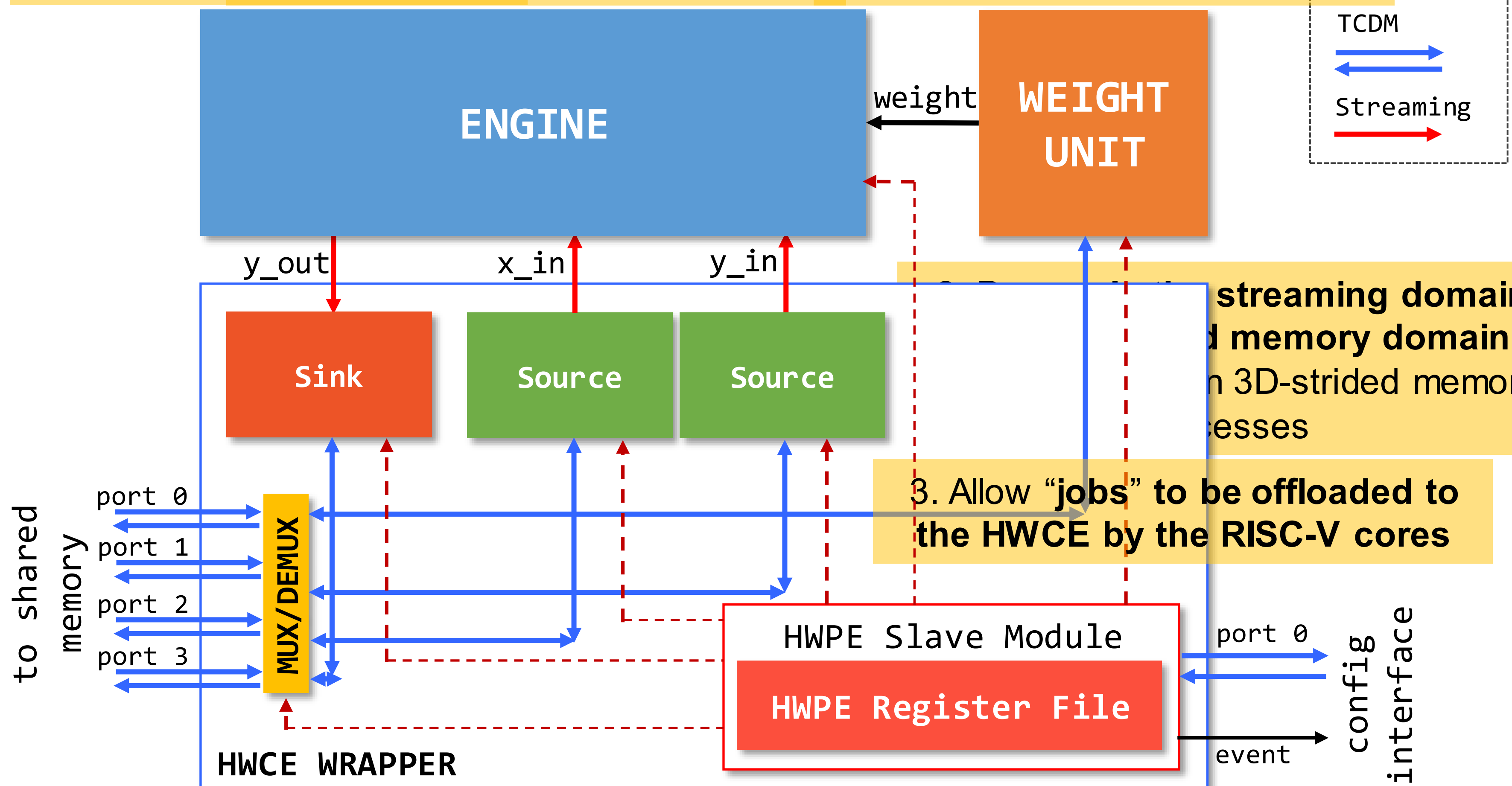


HW Convolution Engine

5. **Fine-grain clock gating** to minimize dynamic power

1. **Perform convolve-accumulate in streaming** fashion

4. **Weights** for each convolution filter are **stored privately**



streaming domain
d memory domain;
n 3D-strided memory
esses

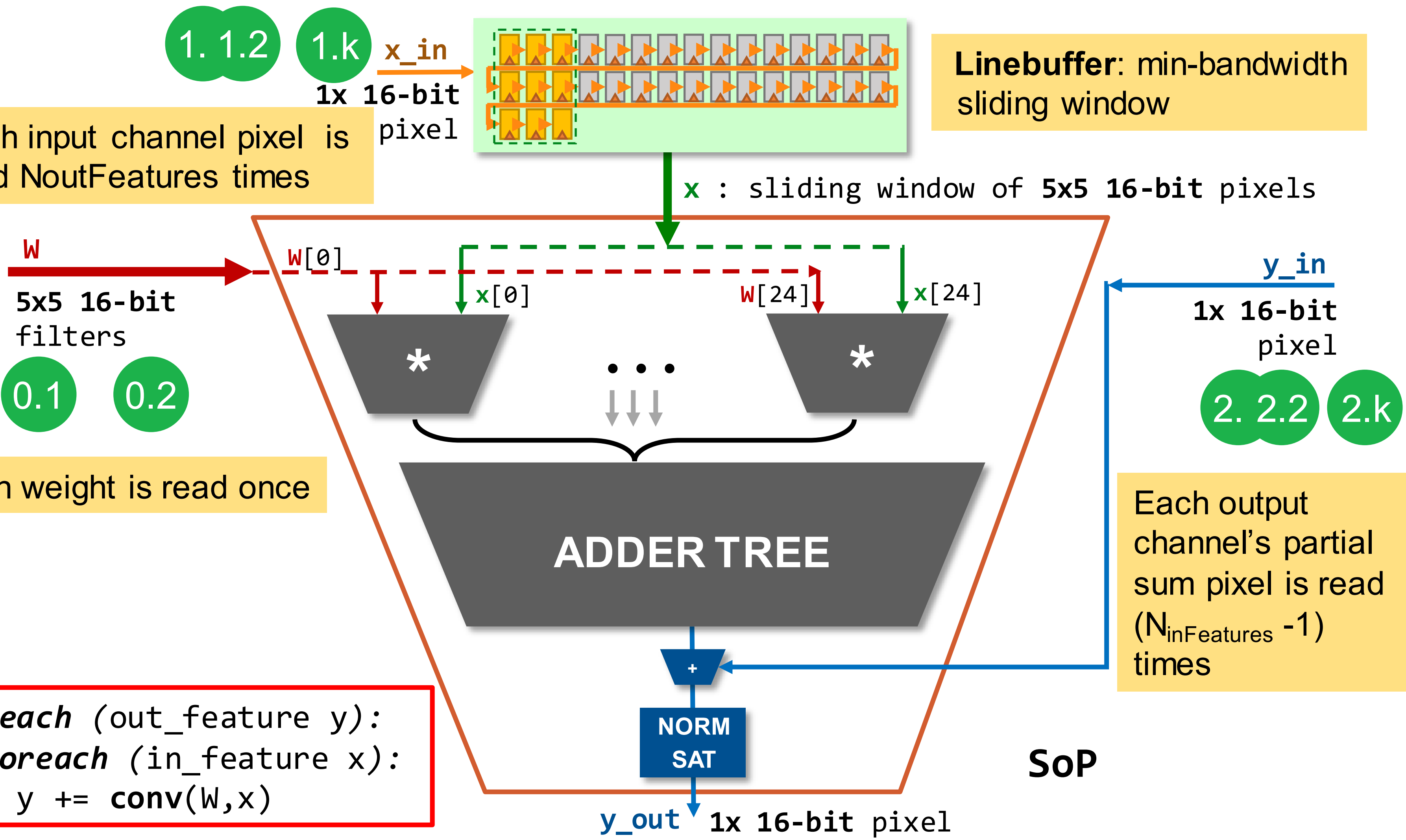
3. Allow "**jobs**" **to be offloaded to the HWCE by the RISC-V cores**

F. Conti and L. Benini, "A ultra-low-energy convolution engine for fast brain-inspired vision in multicore clusters," *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2015, pp. 683-688.

HWCE Sum-of-Products

Each input channel pixel is read NoutFeatures times

Linebuffer: min-bandwidth sliding window



Each weight is read once

Each output channel's partial sum pixel is read ($N_{inFeatures}$ -1) times

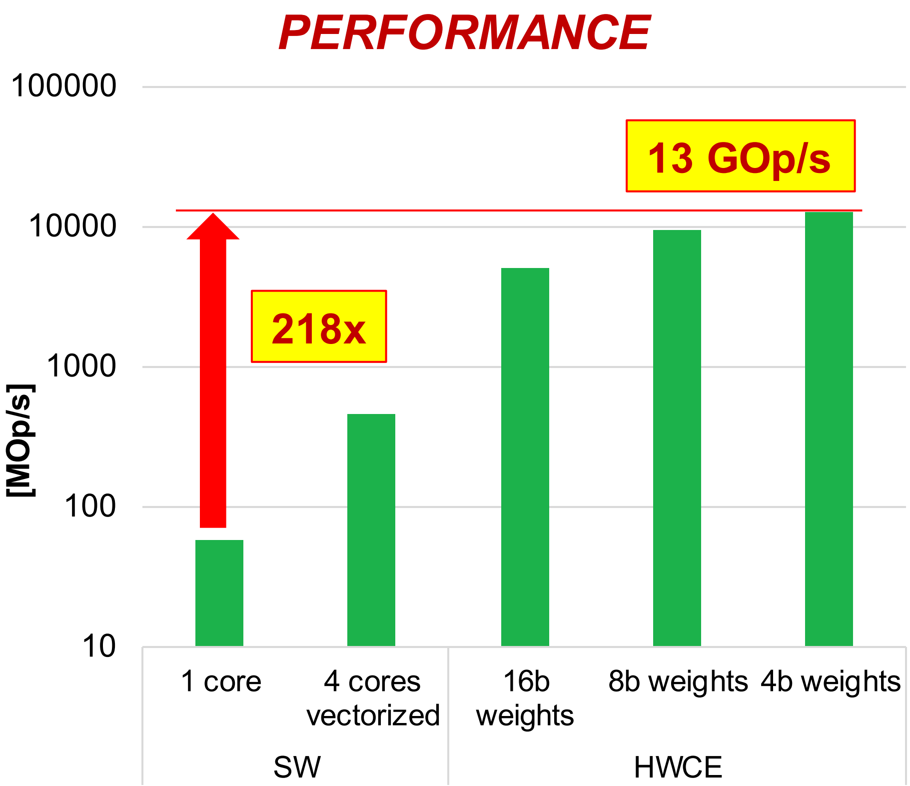
```
foreach (out_feature y):
  foreach (in_feature x):
    y += conv(W,x)
```

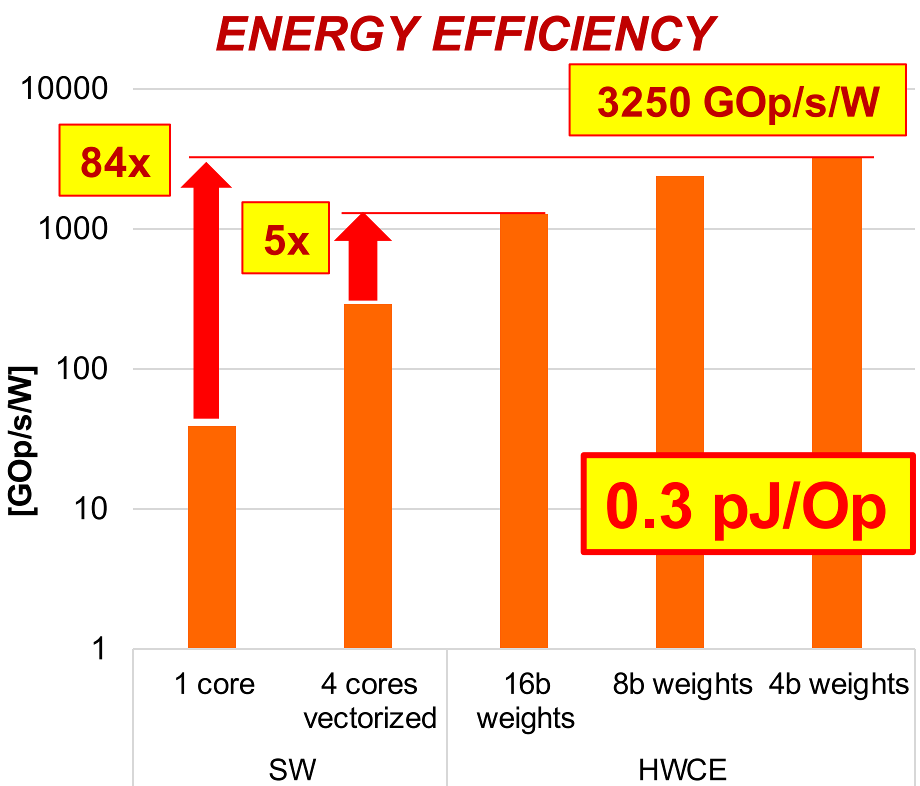
PULP

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

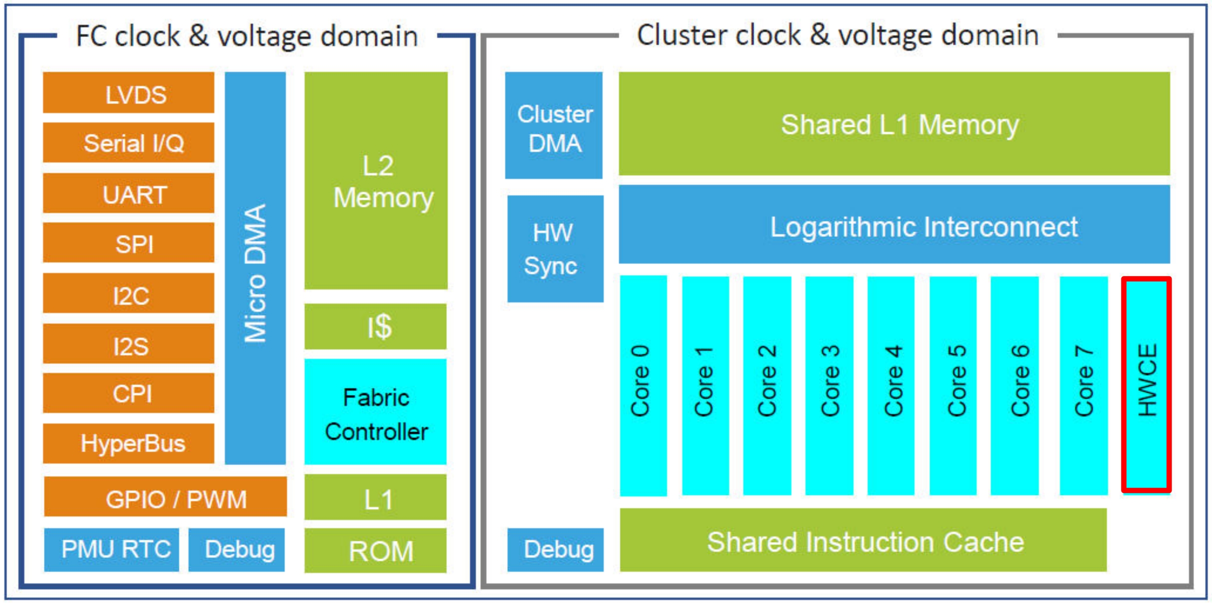




Now coming: HWCEv5.0 – improves scalability & flexibility @ 3TOPS/W

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 co
- Secured execution

Computation engine

- 8 extended RISC
- 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

11 X

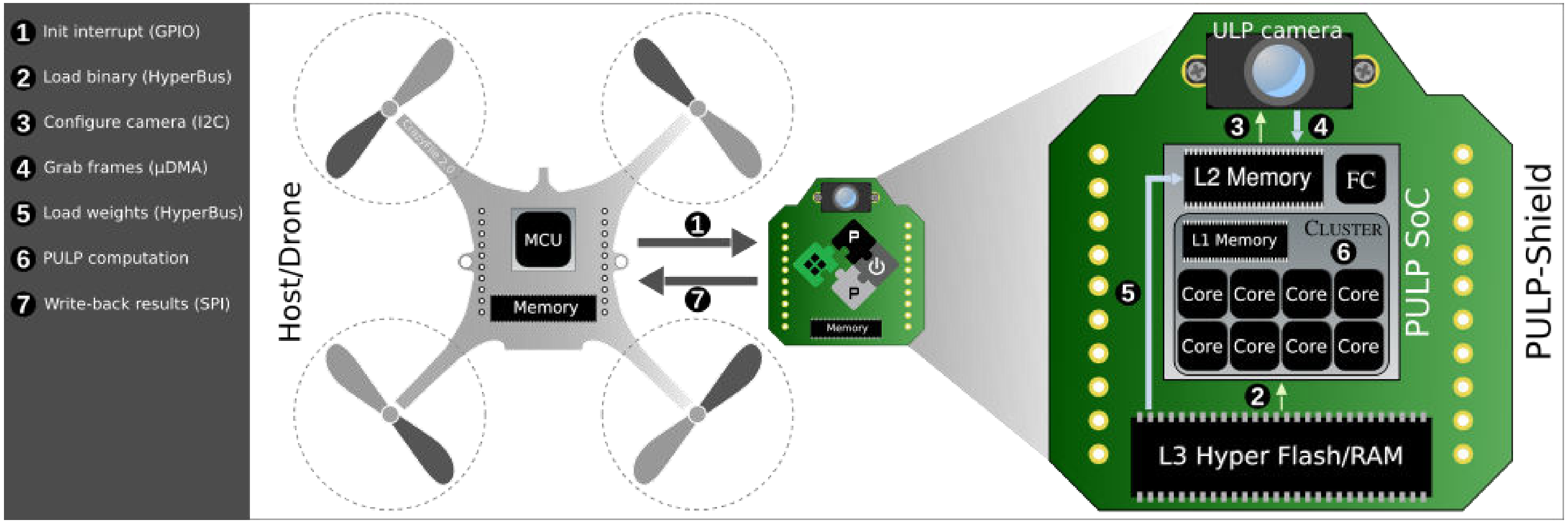
16 X

GREENWAVES TECHNOLOGIES

4x More efficiency at less than 10% area cost

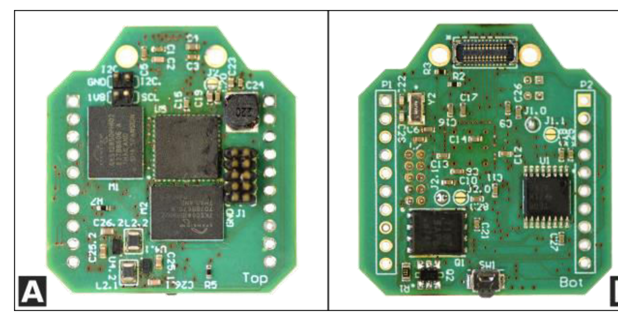
PULP

New Application Frontiers: DroNET on NanoDrone



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

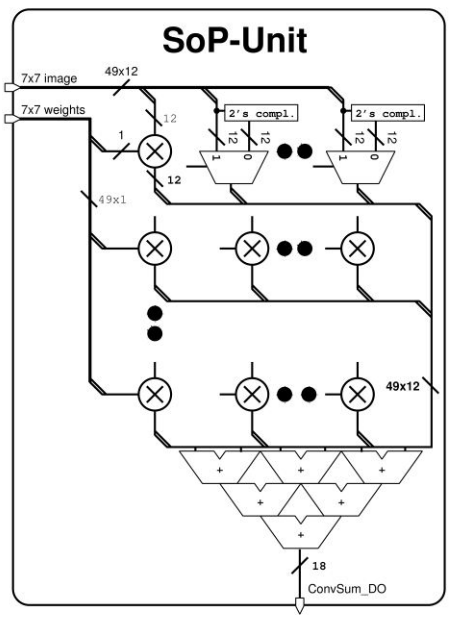
Model	Bit-width	Top-1 error
ResNet-18 ref	32	31.73%
INQ	5	31.02%
INQ	4	31.11%
INQ	3	31.92%
INQ	2 (ternary)	33.98%

SOA INQ retraining

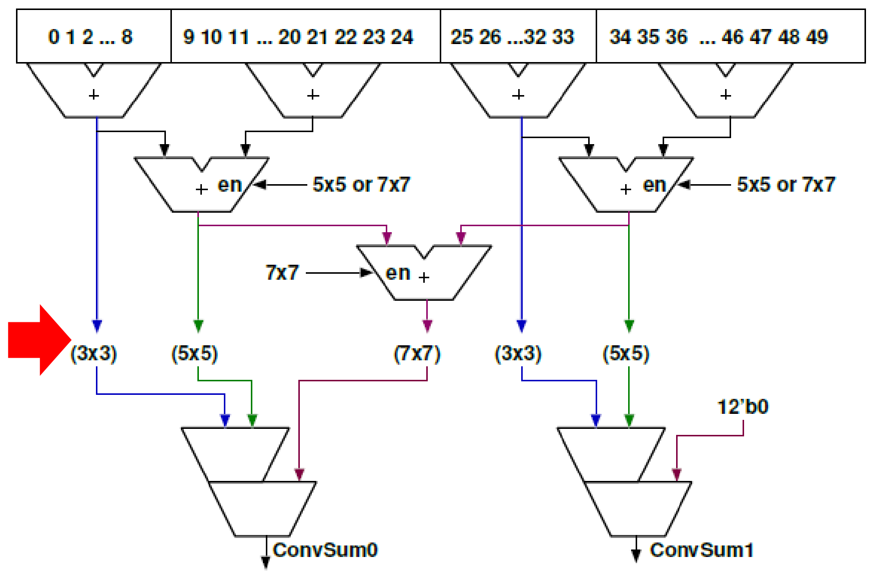
Low(er) precision: 8→4→2

2.2% loss → 0% with 20% larger net

MULT → MUX



Equivalent for 7x7 SoP



ImageBank

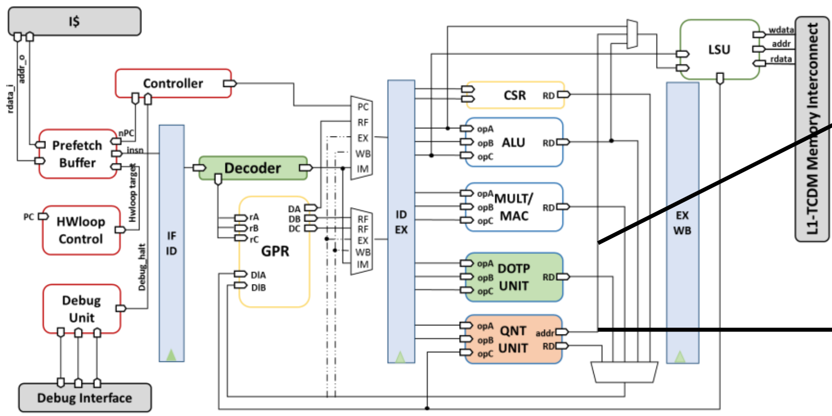
Image Mapping (3x3, 5x5, 7x7)

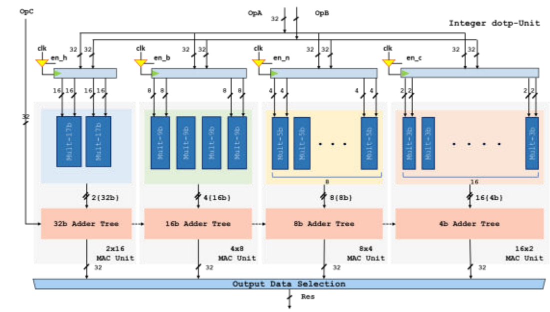
1 MAC Op = 2 Op (1 Op for the "sign-reverse", 1 Op for the add).

PULP

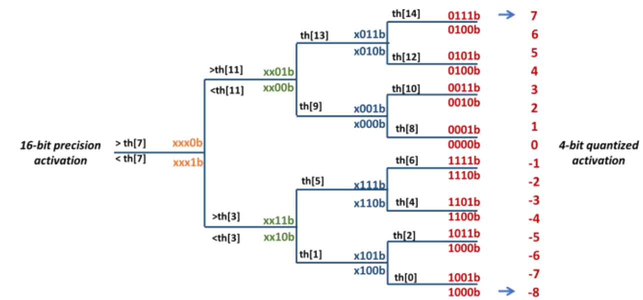
RISC-V ISA Extensions for extreme quantization

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

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

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

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

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

Thresholding

Multi-bit accumulation

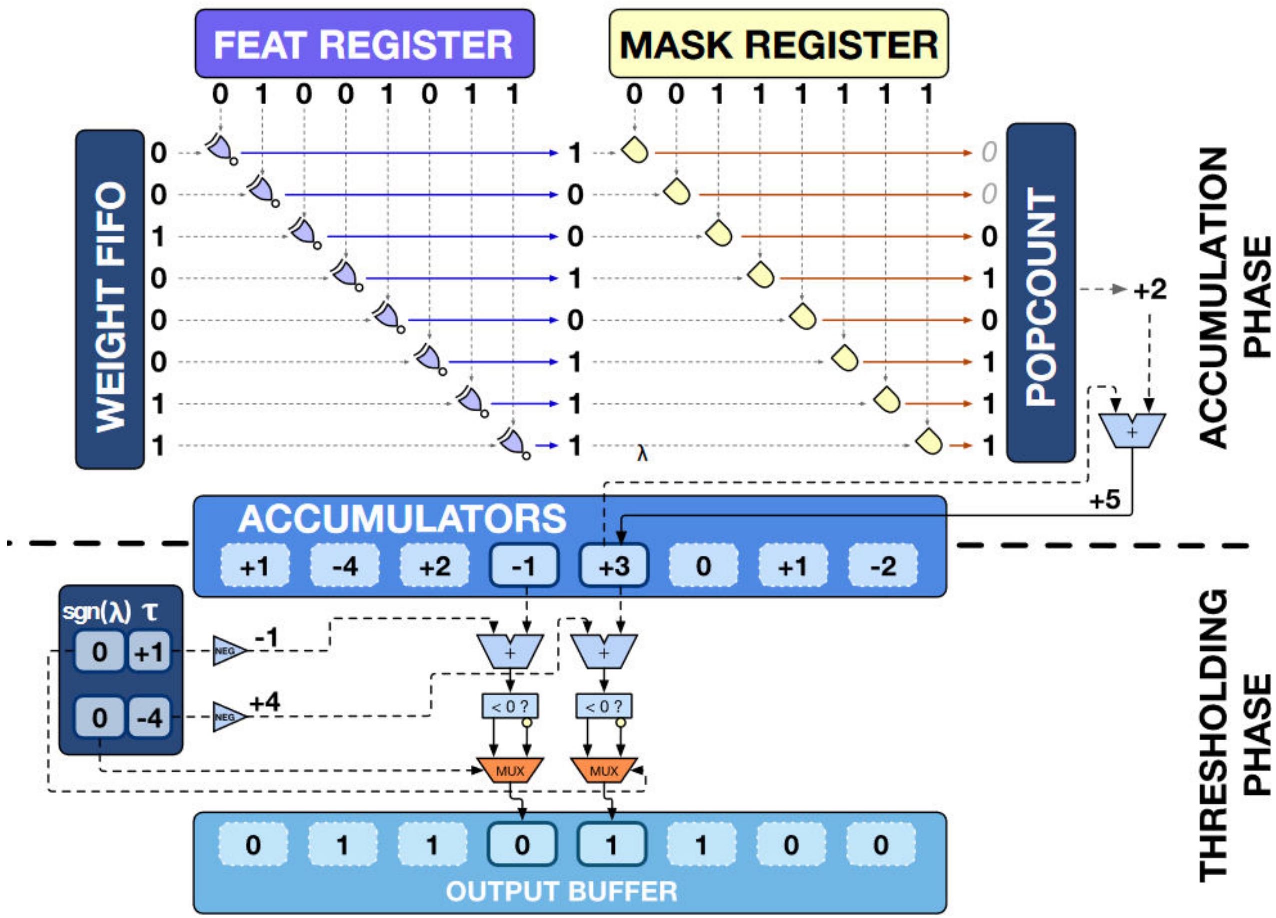
Binary product → XOR

A	B	out
-1	-1	+1
-1	+1	-1
+1	-1	-1
+1	+1	+1

A	B	out
0	0	1
0	1	0
1	0	0
1	1	1

PULP

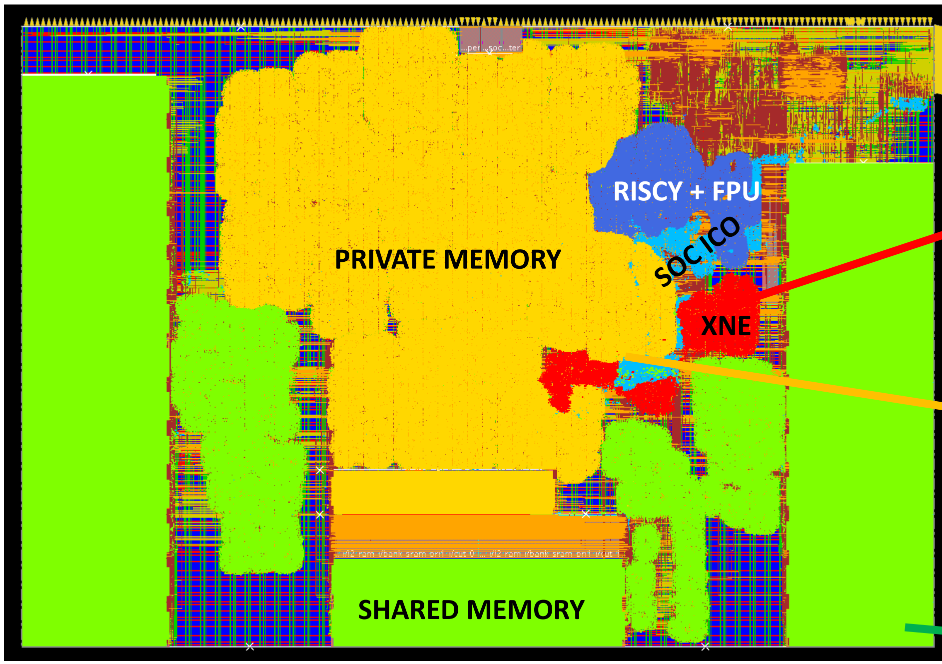
XNE: XNOR Neural Engine



Main unit: binary dot-product and thresholding

Quentin: a XNE-accelerated microcontroller

Quentin in GlobalFoundries 22FDX



XNE area is **~14000 um2** (71 KGE, 72% Riscy+FPU)

Private memory is **448 KB** SRAM + 3r2w **8 KB** SCM

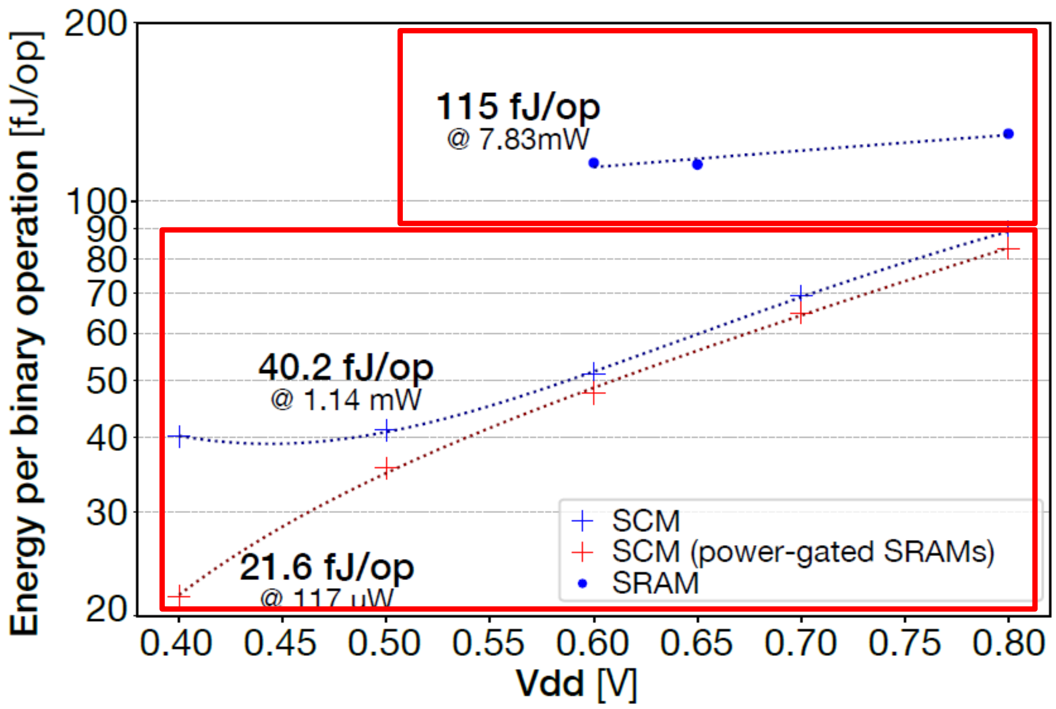
Shared memory is **56 KB** SRAM + **8 KB** SCM

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: $\mathbf{y}(k_{out}) = \mathbf{b}(k_{out}) + \sum_{k_{in}} \left(\mathbf{W}(k_{out}, k_{in}) \otimes \mathbf{x}(k_{in}) \right)$

α_i, β_j: power-of-2
Scale Module

Inspired by ABC-Net:

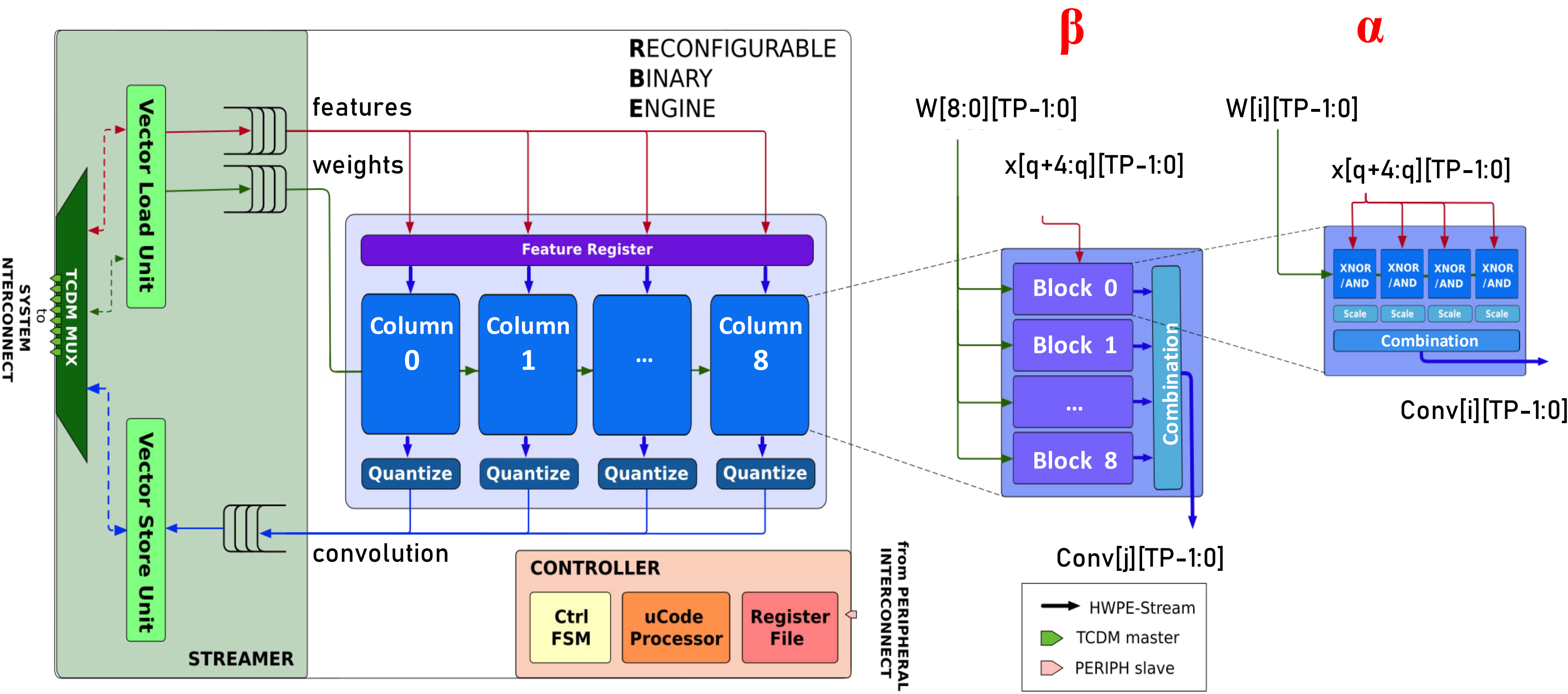
BBQ NN layer: $\mathbf{y}(k_{out}) \approx \mathbf{b}(k_{out}) + \sum_{i=0..Q_W} \sum_{j=0..Q_A} \sum_{k_{in}} \boxed{\alpha_i \beta_j} \boxed{\left(\mathbf{W_{bin}}(k_{out}, k_{in}) \otimes \mathbf{x_{bin}}(k_{in}) \right)}$

Binary NN
BinConv
module

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

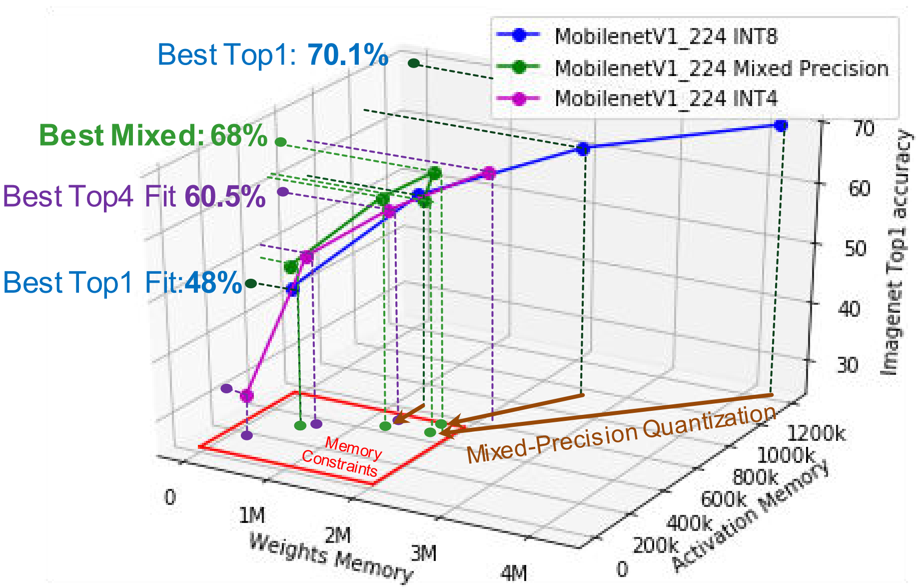
Reconfigurable Binary Engine

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



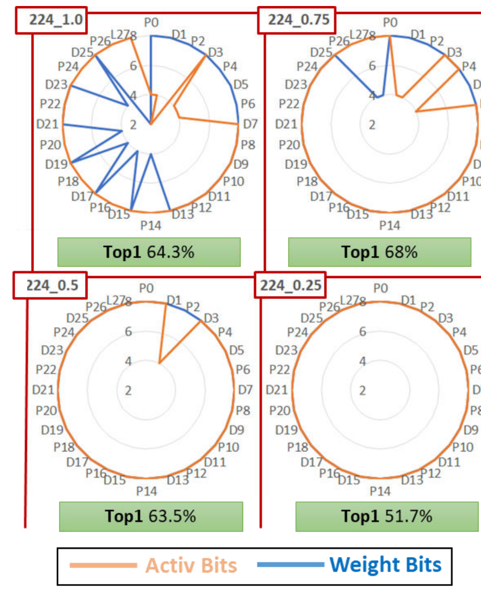
Mixed-Precision Quantized Networks

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

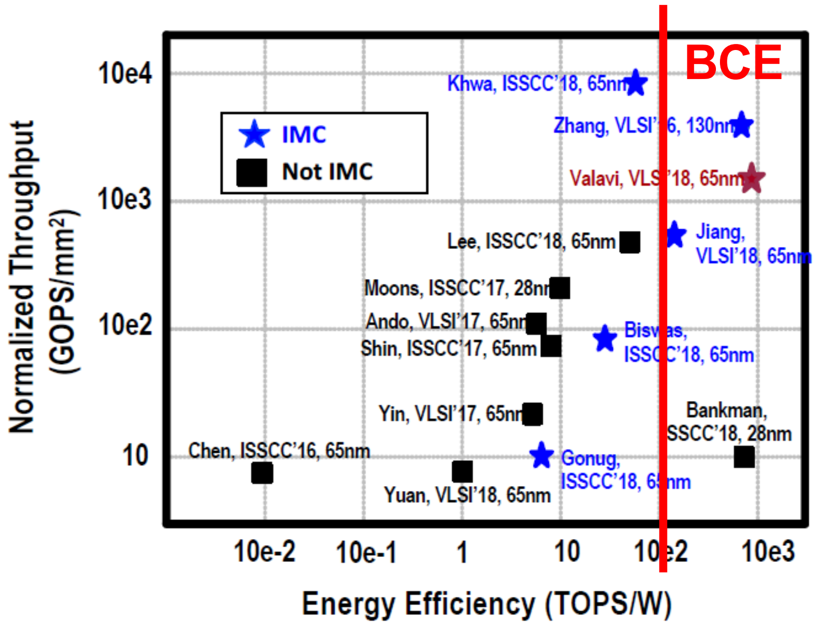


Only **-2%** wrt most accurate INT8 mobilenet (224_1.0) which does not fit on-chip

+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

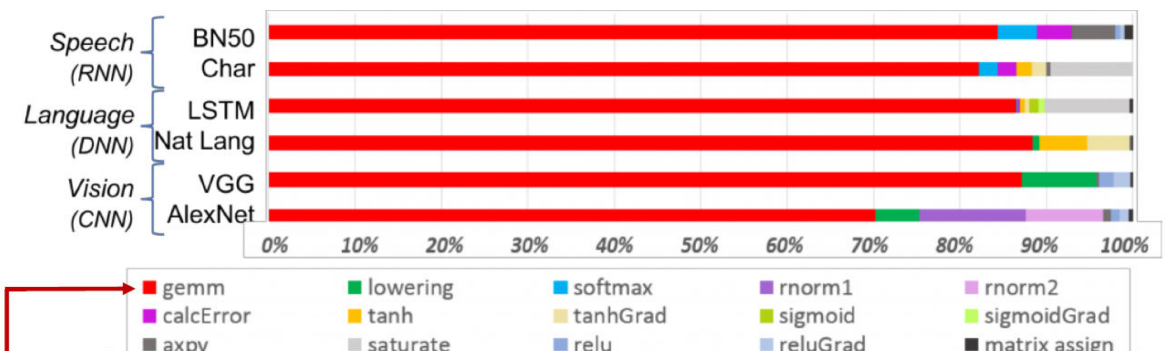
Diversion: why not a fully hardwired engine?



1000TOPS/W…

Flexibility is essential!

However....



[B. Fleischer, *VLSI'18*]

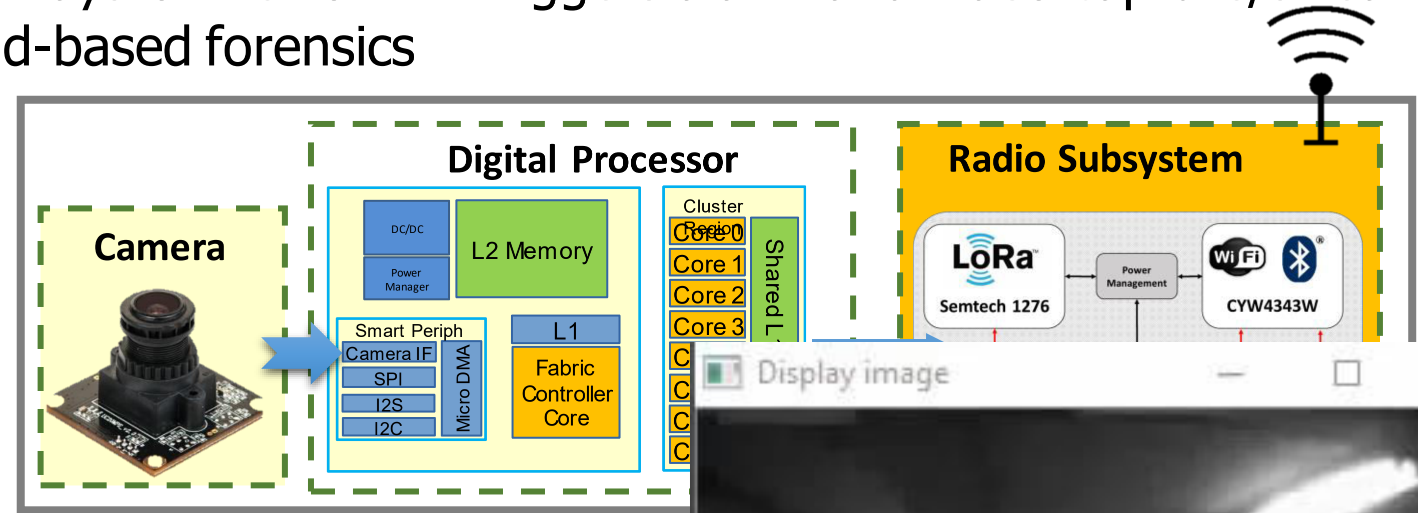
General Matrix Multiply (~256×2300=590k elements)

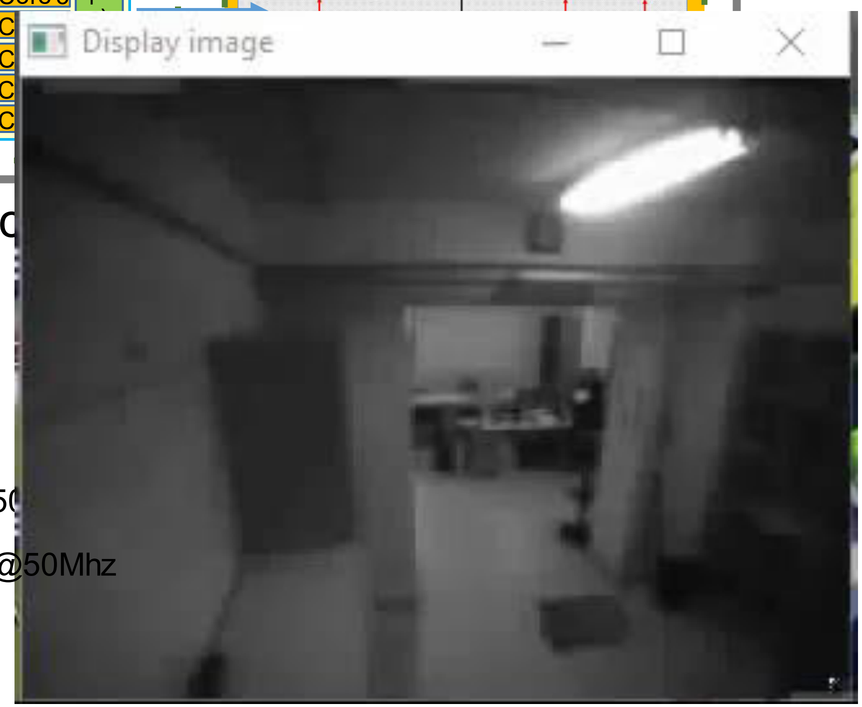
Single/few-word operands (traditional, near-mem. acceleration)

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

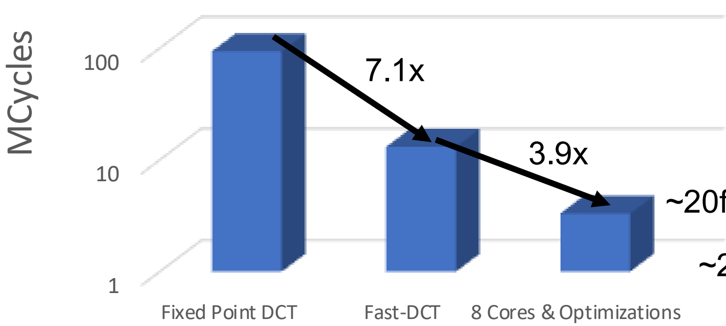
What about uW «sleep»?

Small always-on nework → triggers alarm and video capture/streaming for cloud-based forensics





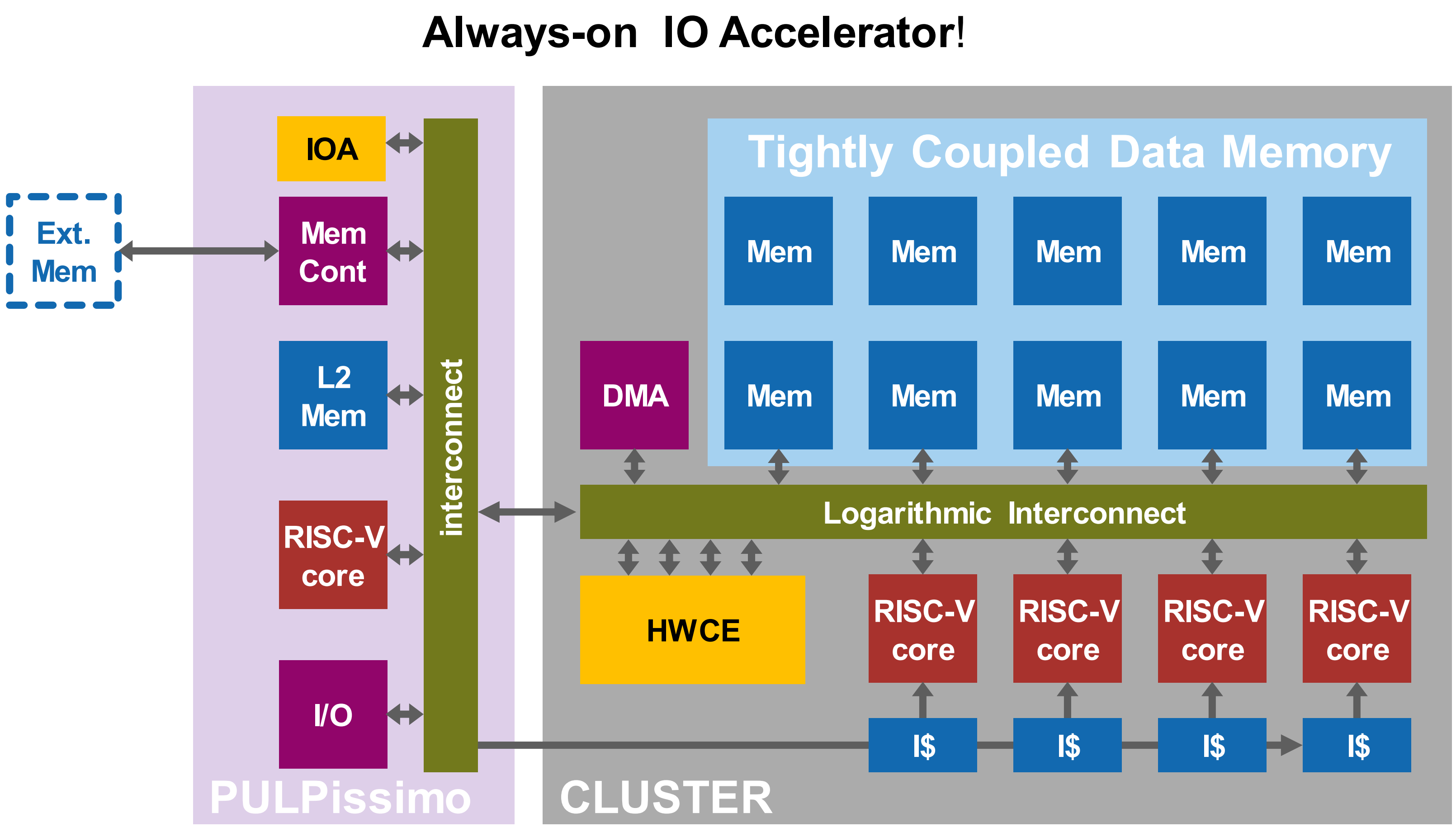
JPEG encoding on c



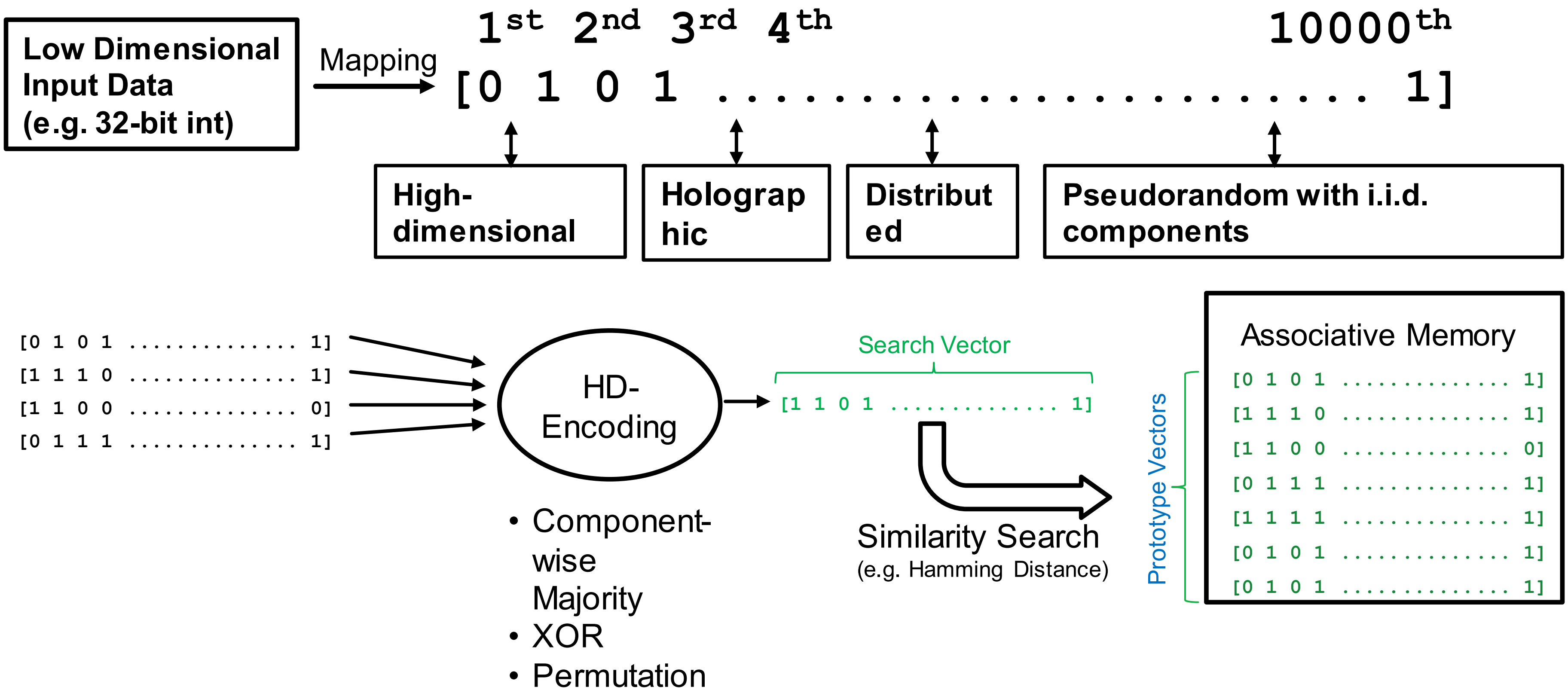
~20fps@50Mhz

Target 30 fps ...

Need uW always-on Intelligence



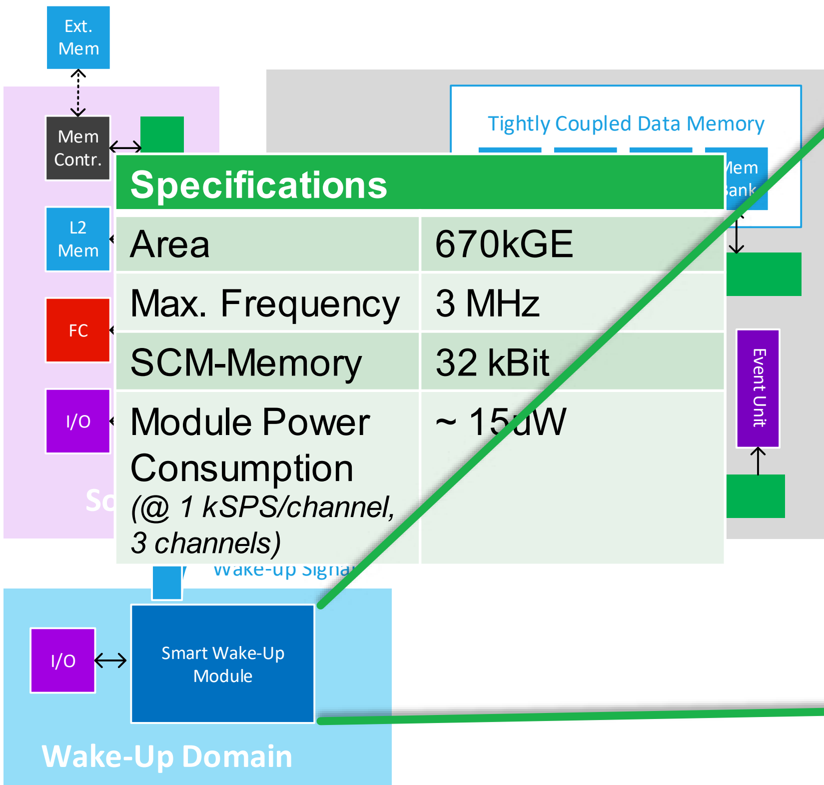
Not Only CNNs: Hyper-Dimensional Computing



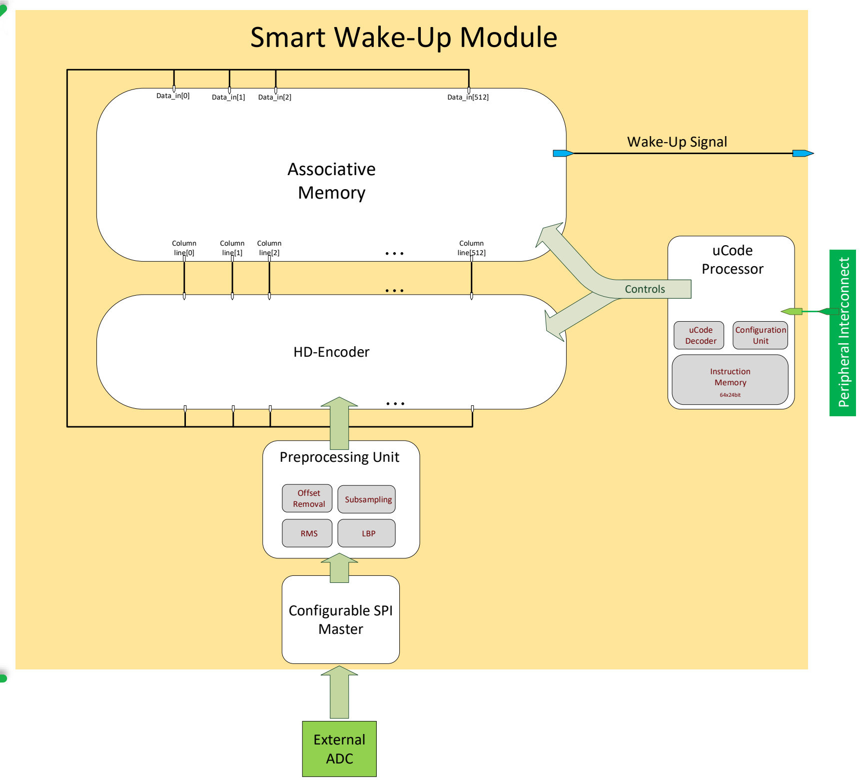
Highly parallel, fault-tolerand binary operators, assoc-min-distance search

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

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



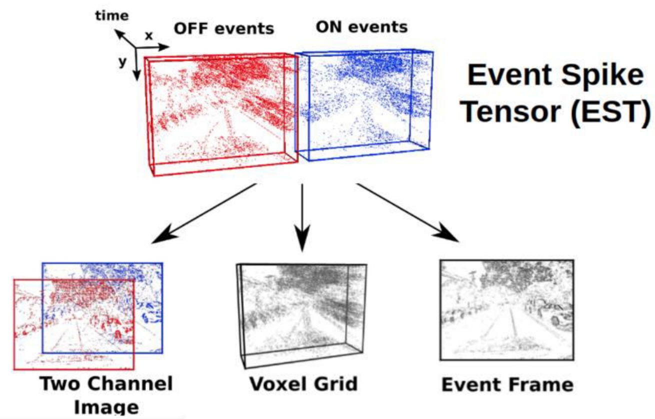
Specifications	
Area	670kGE
Max. Frequency	3 MHz
SCM-Memory	32 kBit
Module Power Consumption *(@ 1 kSPS/channel, 3 channels)*	~ 15uW

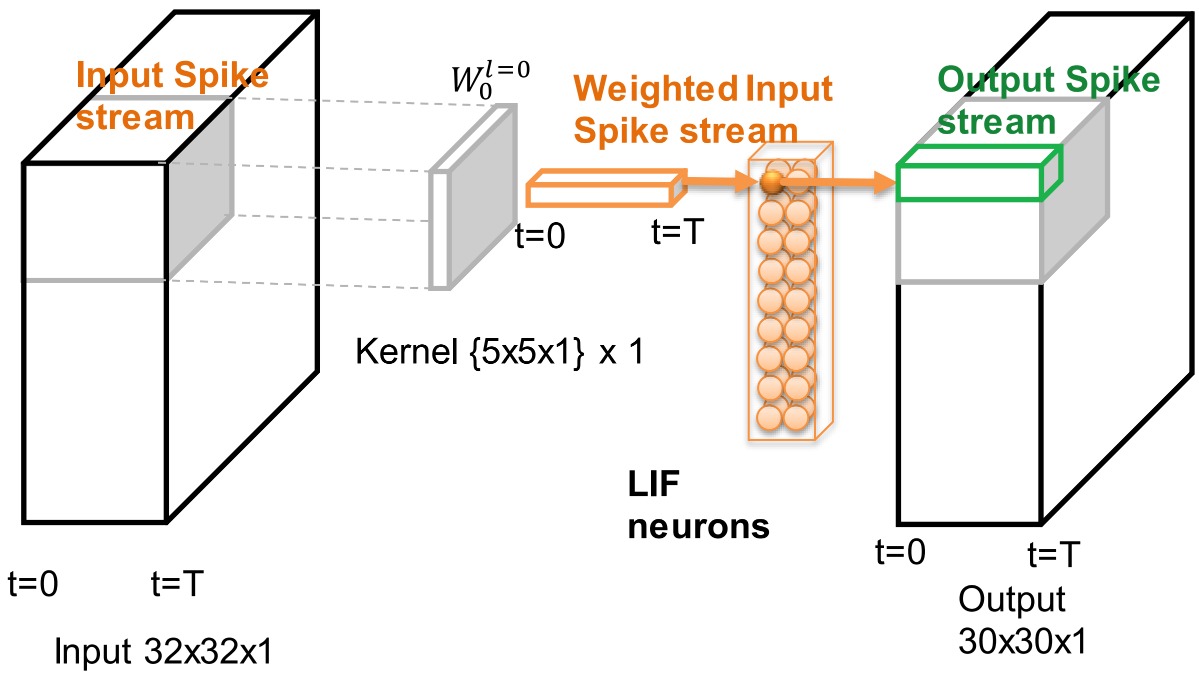


Taped out in 22fdx

Spiking Convolutional Neural Network

DVS camera



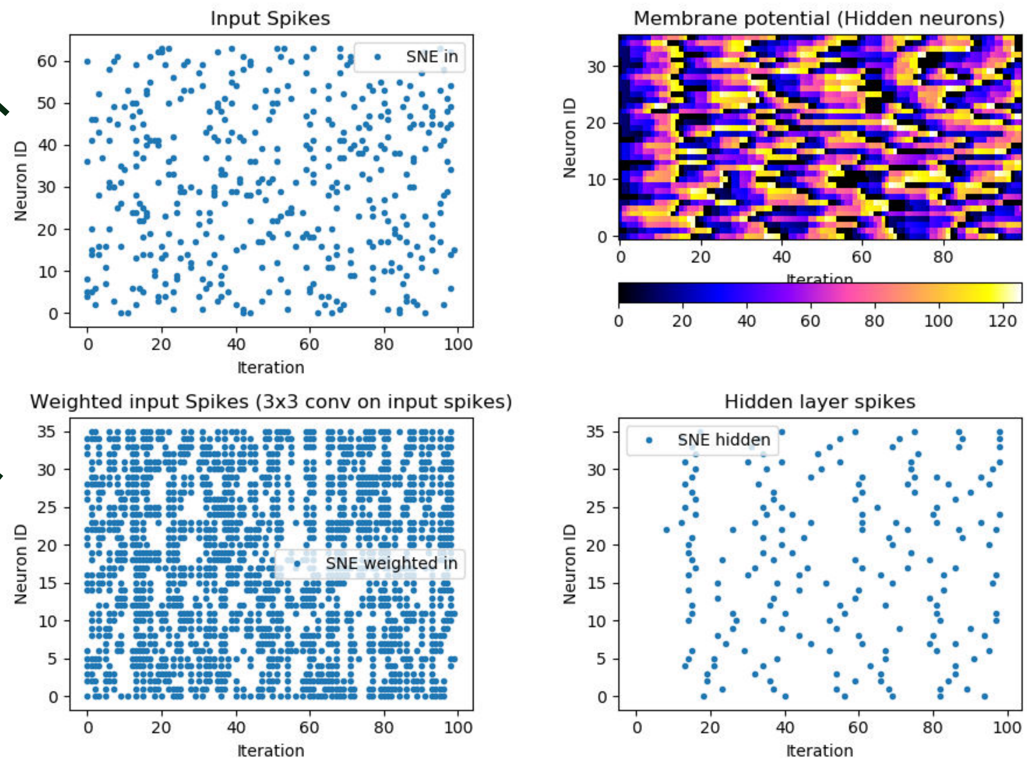


- Activity-driven computation
- Event-like (sparse) feature representation
- No processing in absence of input events
- Lightweight pre-processing required on emerging Event-based sensor data (e.g. DVS cameras)

Leaky Integrate&Fire CSNN layer

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

Weighted stream
Conv. (3x3 kernel)

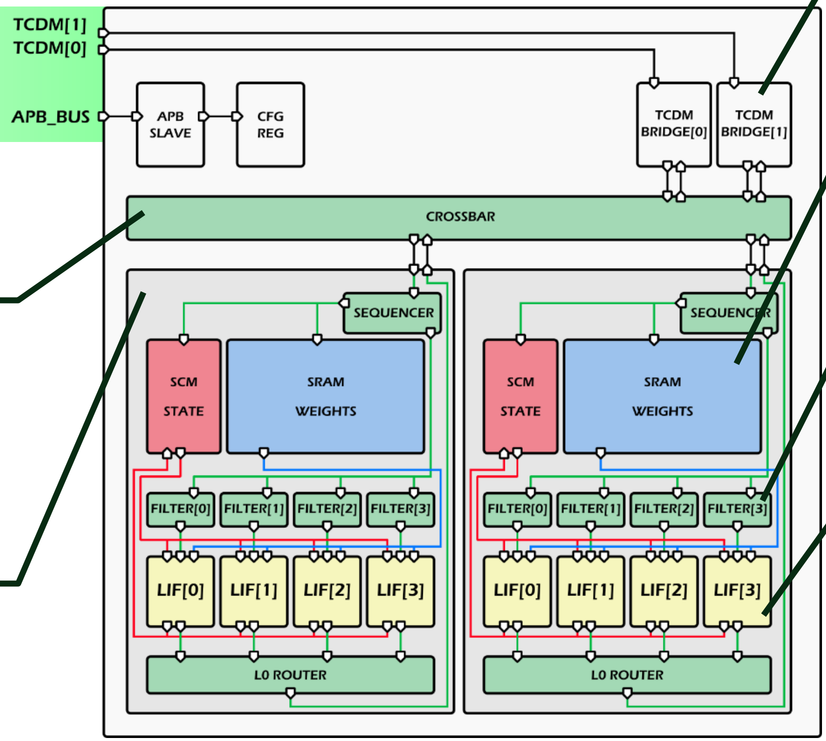


Membrane potential
LIF state variable

Output spikes
When Membrane potential exceed the threshold

SNE Accelerator Architecture

PULP SoC interconnect



Combinational crossbar
High data routing flexibility and low-latency communication

High Modularity
Configurable number of LIF neuron blocks
Datapath "slices"

Programmable streamer
autonomous data fetching

Data locality
Weight reuse and Neuron state stored locally

Data re-ordering optimized for Convolutional SNN

Digital Leaky
Integrate&FIRE
neuron model

PULP

First silicon implementation

ROSETTA SoC

SoC physical implementation

- TSMC65 nm technology
- Chip area 4100µm x 3000µm
- Gates full SoC 6M

Accelerator physical implementation

- Area Accelerator 333μm^2
- Gates Accelerator ~260k

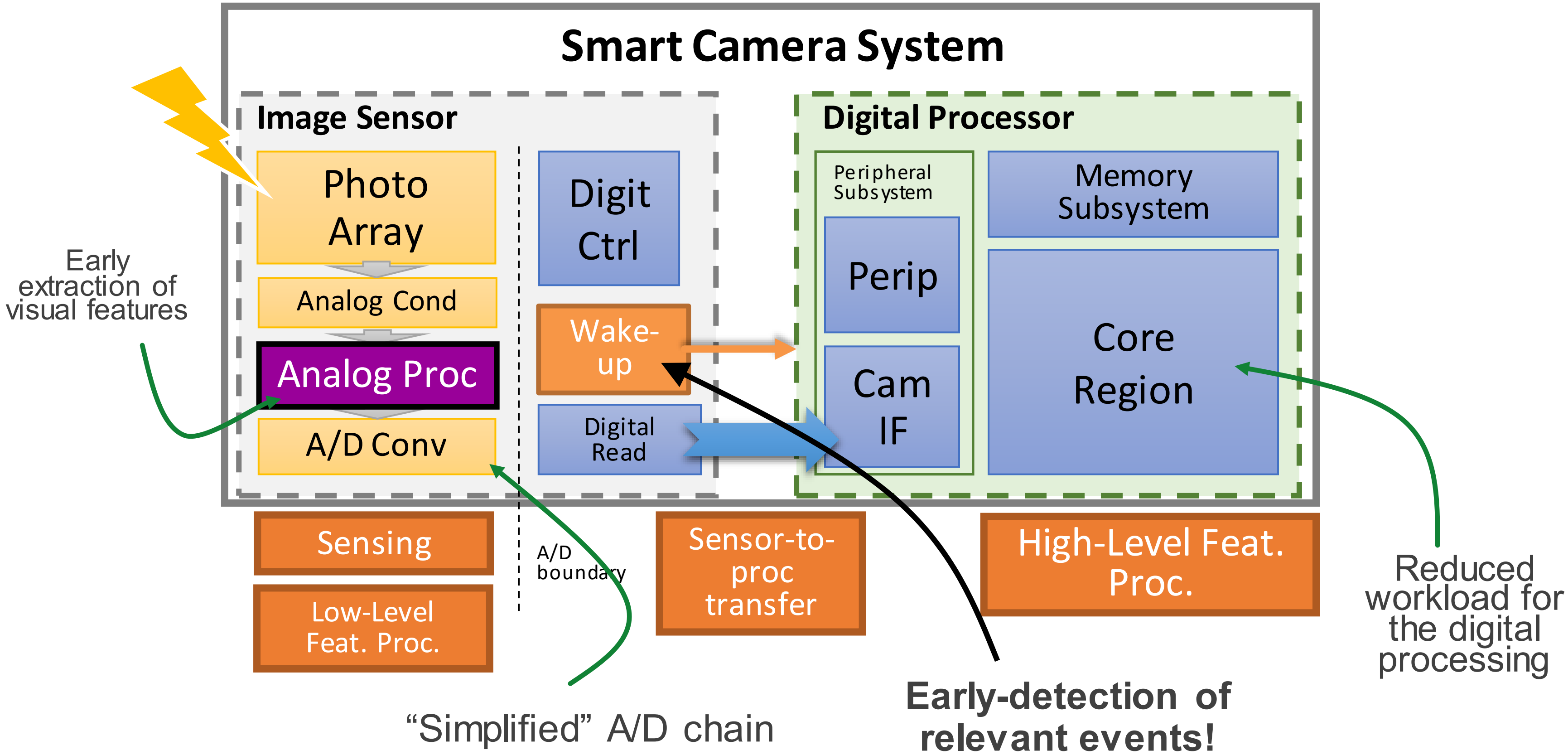
Rosetta's SNE configuration

- 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	~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**.

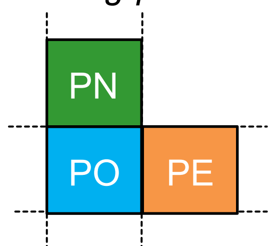


[Fernández-Berni, Jorge, et al. "Image Feature Extraction Acceleration." Image Feature Detectors and Descriptors. Springer International Publishing, 2016. 109-132.

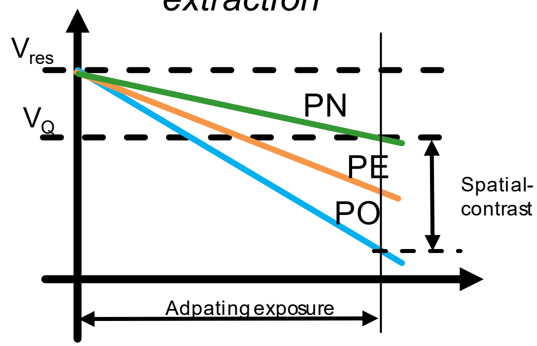
Ultra-Low Power Imaging (GrainCam)

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

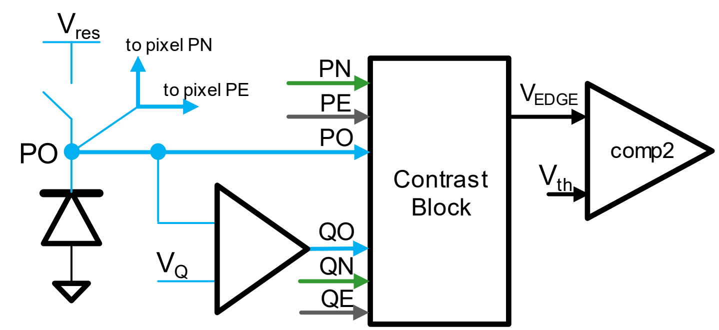
'Moving' pixel window



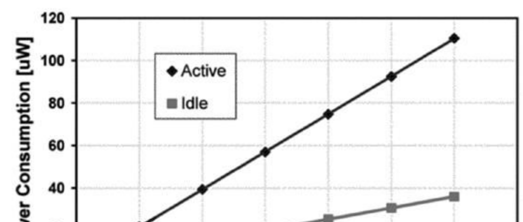
Gradient extraction



Per-pixel circitut for filtering and binarization



Ultra-Low Power Consumption <100uW



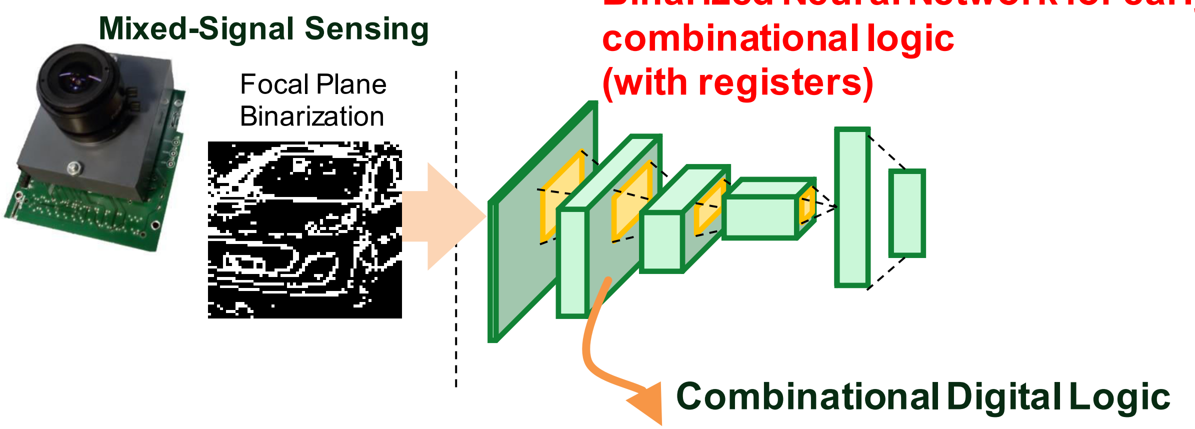
<10x wrt SoA imagers

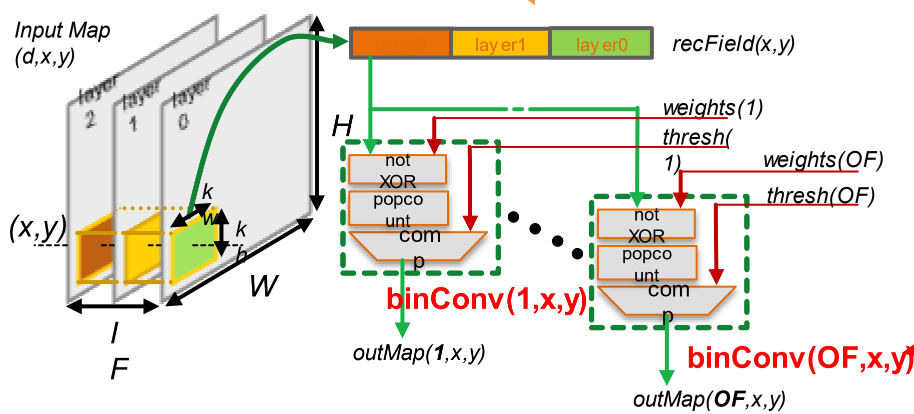
This process naturally reflects the operation of a binarized pixel-wise convolution and can be seen as embedding the first convolutional layer within the image sensor die

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

Binarized Neural Network for early detection as combinational logic (with registers)





Apply XNOR and binarization over input data to produce any of output pixels

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 [mm^2]	area [MGE]†	time/img [ns]	time/img [FO4]‡	E/img [nJ]	leak. [µW]	E-eff. [TOp/J]
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

† Two-input NAND-gate size equivalent: 1 GE = 0.199 μm^2
‡ 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 Imem 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 hiearchy optimizazion and logic+physical optimizazion
  - 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!

PULP


The fun is
just beginning
http://pulp-platform.org

Closing the Bin/Ternarization accuracy Gap

Table 1: Experimental Results on ImageNet

Model	Method*		Levels†	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+‡	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+‡	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	**RPR (ours)**		3: $\{-1, 0, 1\}$	**66.31/87.84**
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	**RPR (ours)**		2: $\{-1, 1\}$	**64.62/86.01**
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	**RPR (ours)**		3: $\{-1, 0, 1\}$	**71.83/90.28**
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	**RPR (ours)**		2: $\{-1, 1\}$	**65.14/86.31**
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	**RPR (ours)**		3: $\{-1, 0, 1\}$	**64.88/86.05**
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	**RPR (ours)**		2: $\{-1, 1\}$	**62.01/84.83**

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