ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Plenty of Room at the Bottom?

Micropower Deep Learning for Cognitive Cyberphysical Systems

Luca Benini

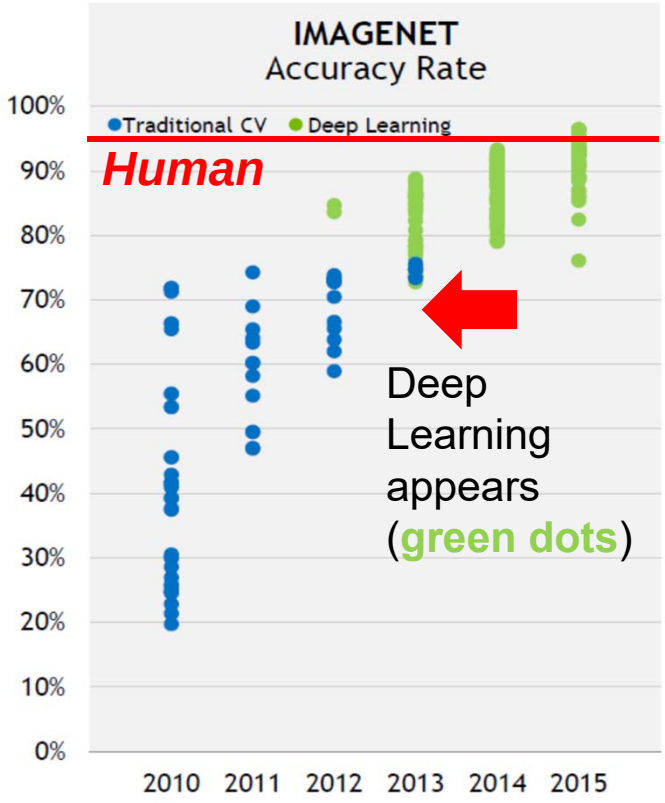
http://www.pulp-platform.org

PULP
Parallel Ultra Low Power

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

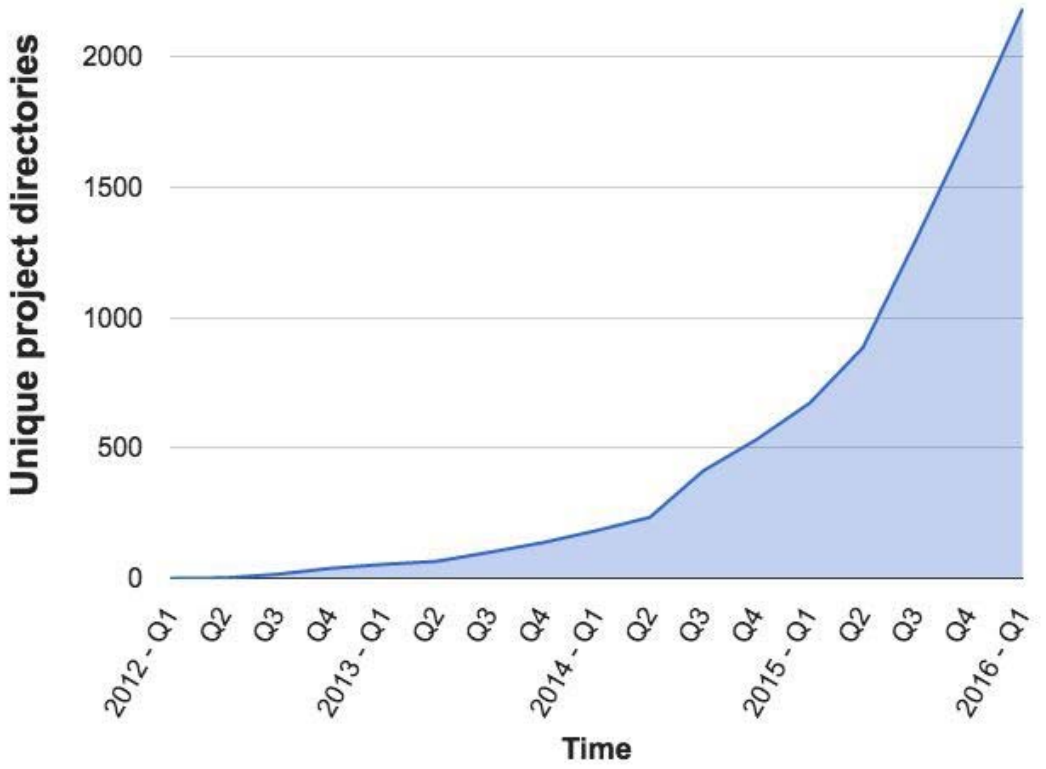
Deep Learning: Why?

First, it was machine vision…

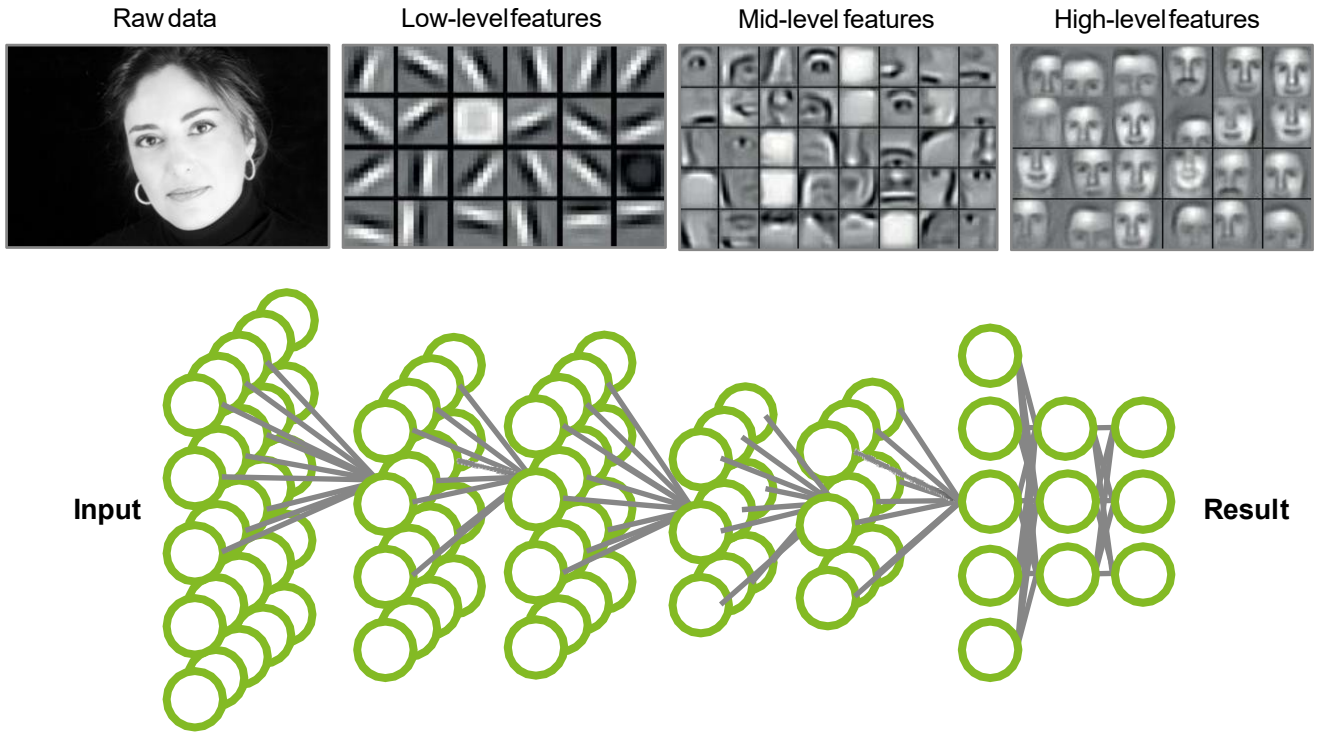


Now it's everywhere!

Growing Use of Deep Learning at Google [J. Dean]



Deep neural networks (DNNs)



Application components:

Task objective
- e.g. Identify face

Training data
- 10-100M images

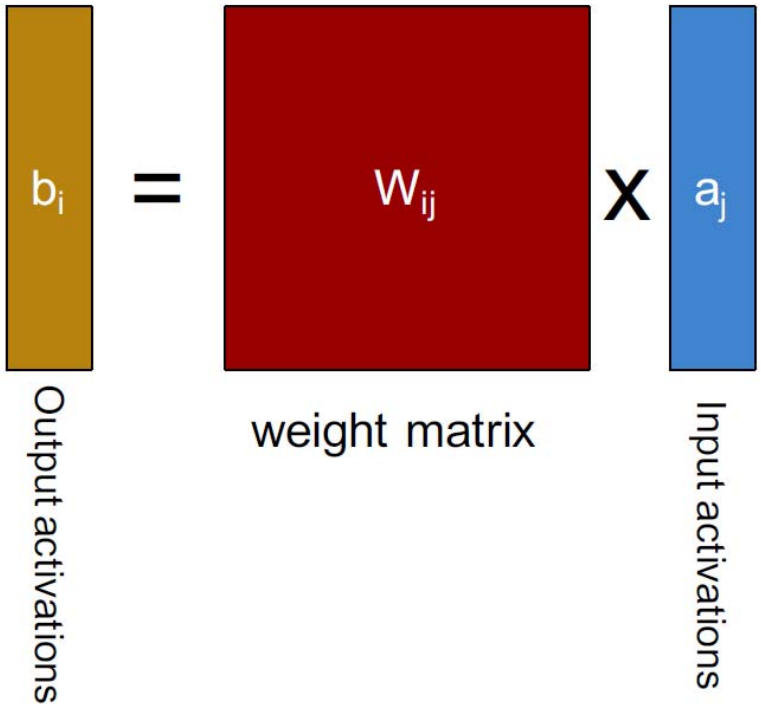
Network architecture
- ~10 layers
- 1B parameters

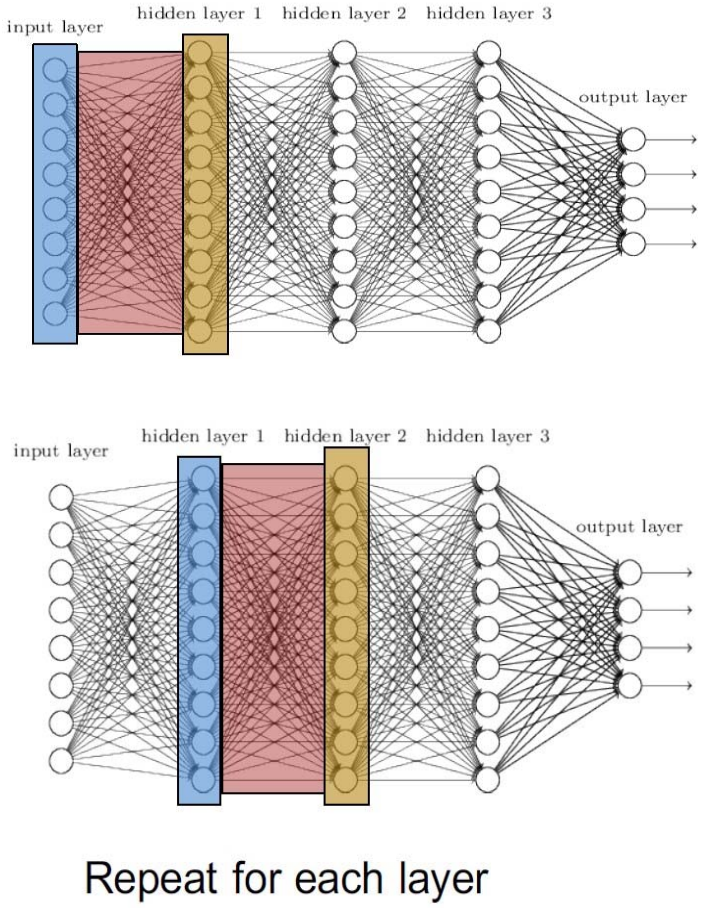
Learning algorithm
- ~30 Exaflops
- ~30 GPU days

NVIDIA

Key operation is dense M x V

Layer-by-layer computation





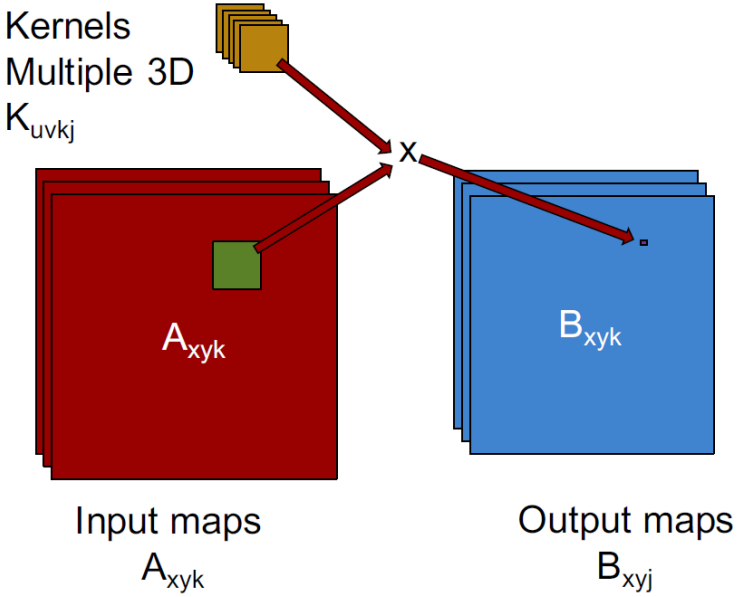
Repeat for each layer

Training by recursive backpropagation of error on fitness function

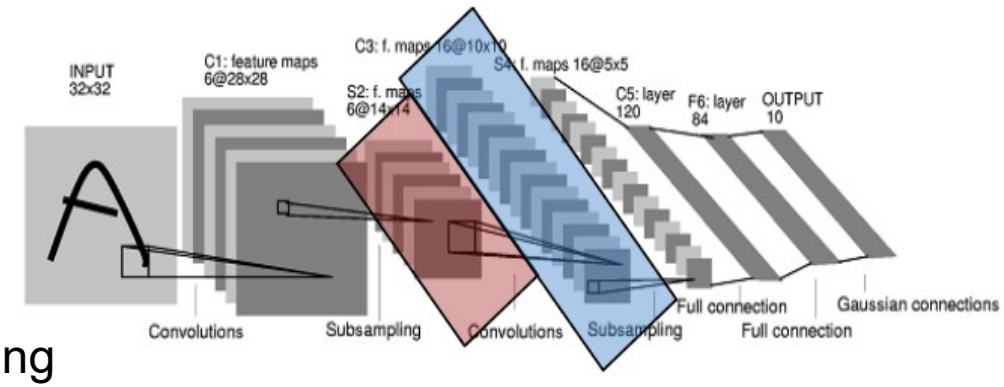
NVIDIA

Deep Convolutional NNs

CNN Computation: main kernel (per layer)



6D Loop
For each output map j
  For each input map k
    For each pixel x,y
      For each kernel element u,v
        B_{xyj} += $A_{(x-u)(y-v)k}$ x K_{uvkj}



Filters conserved through plane

MAC-dominated – even without batching

Can be cast as matrix multiply

NVIDIA

GPUs are Great for Vanilla CNNs

Why?

Because they are good at matrix multiply → 90% utilization is achievable (on lots of "cores")

28pJ/OP*

Pascal GP100
3840 "cores"

↓

3840 MAC/cycle
@ 1.4GHz

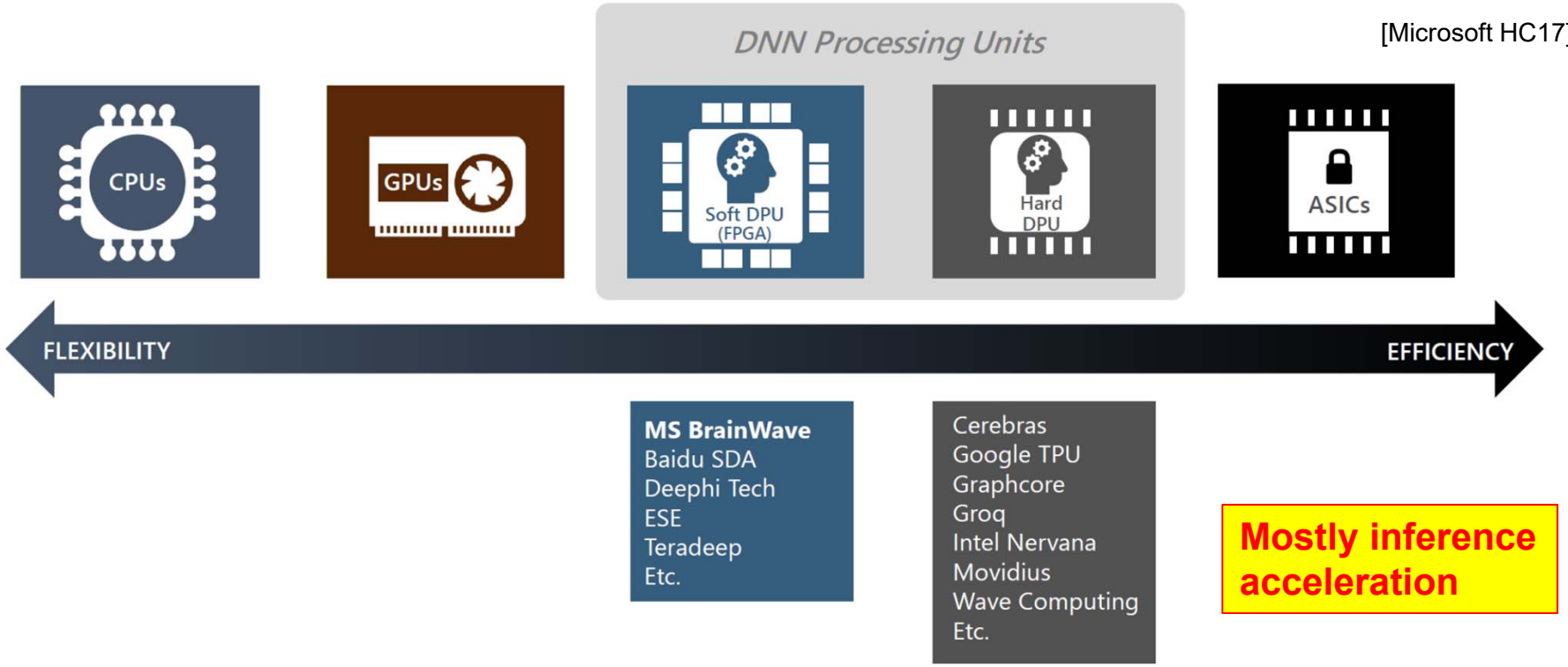
↓

5.3 TMACS (FP)
@300W

*Volta with tensor engine claims 4x better E

HW for deep Networks: Frenzy

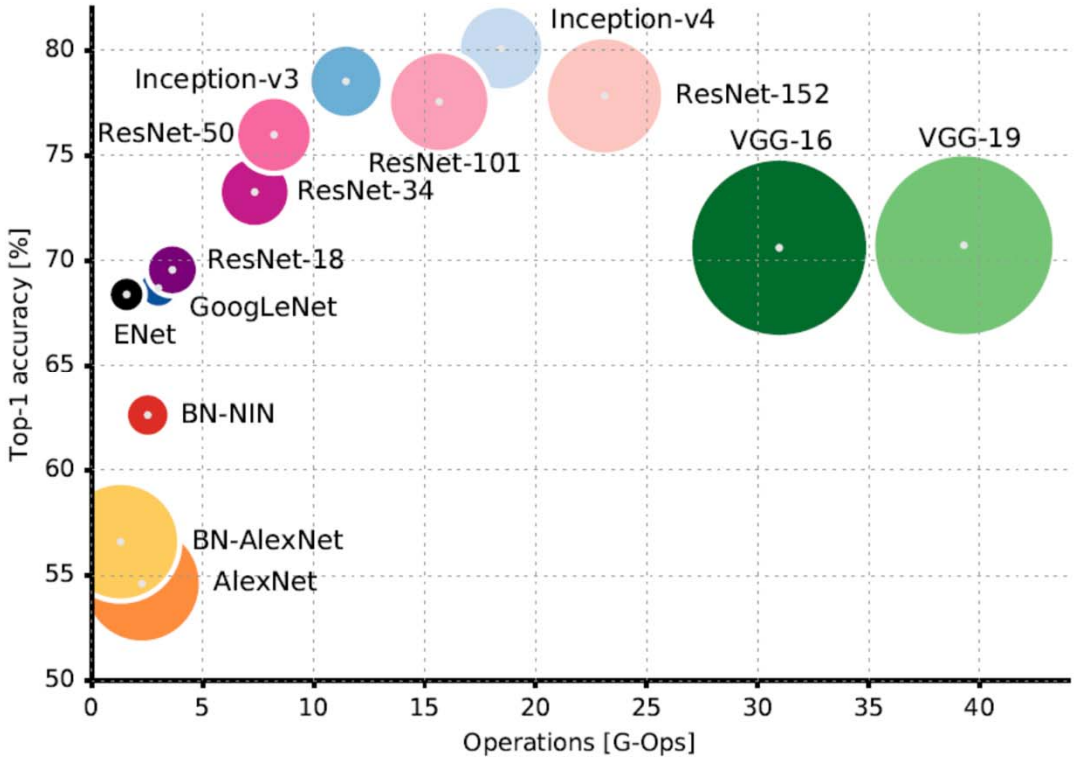
[Microsoft HC17]



Mostly inference acceleration

Datacenter → High-performance embedded →mobile

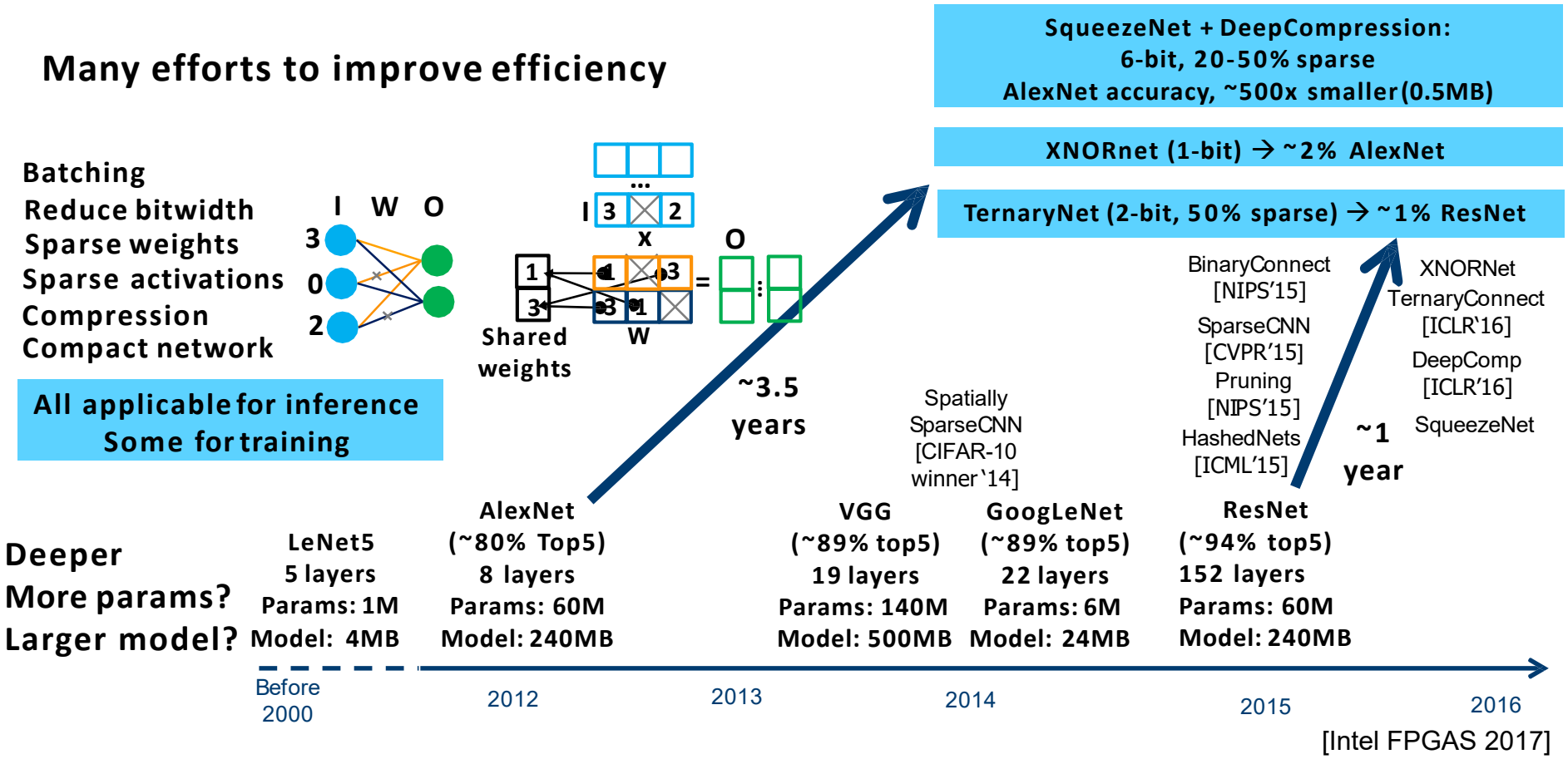
CNN Workloads



[Culurciello16]

Better networks are not necessarily more complex

DNNs Are Evolving Rapidly



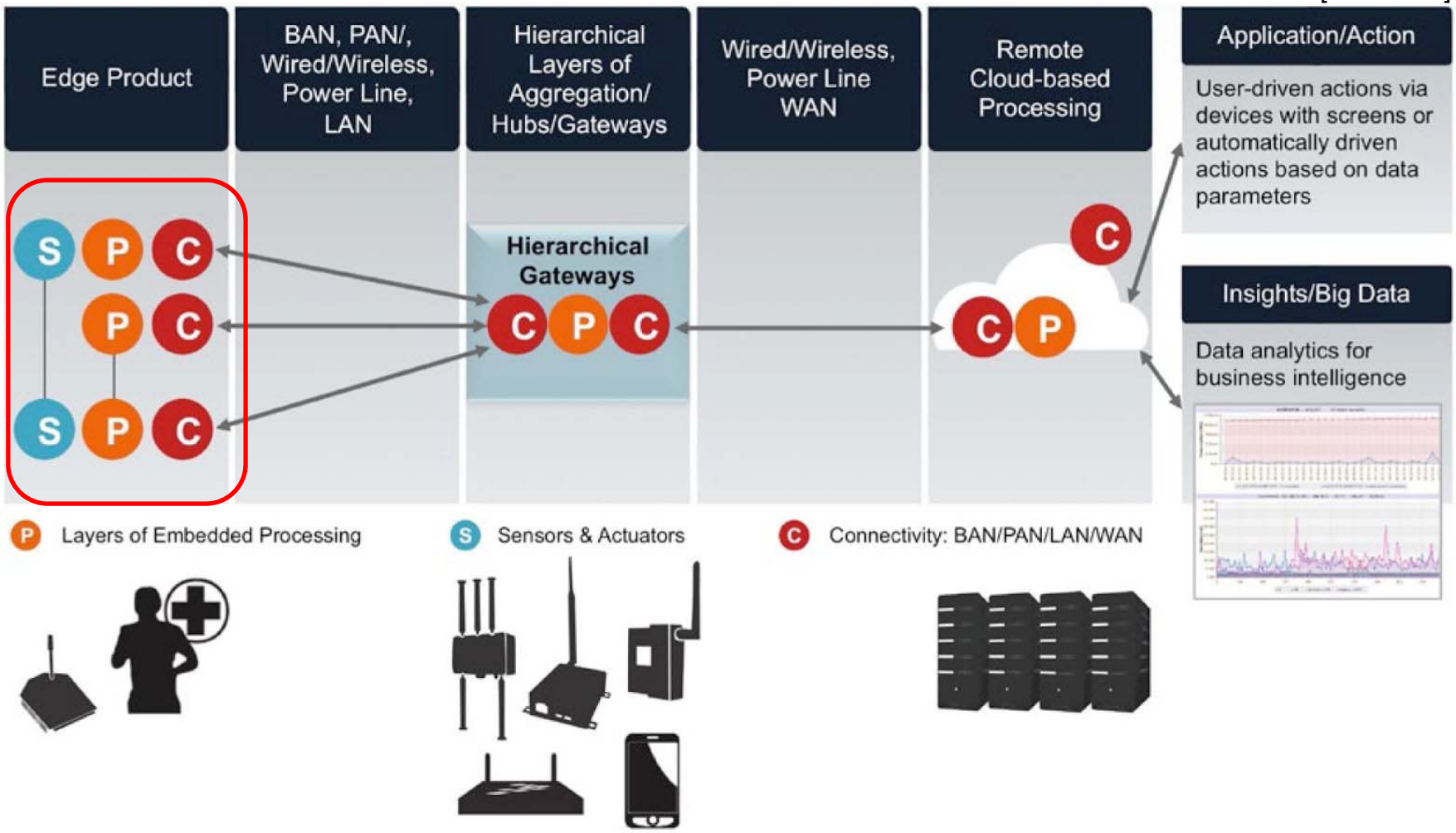
[Intel FPGAS 2017]

Orders of magnitude compute effort an memory reduction with no loss in accuracy

What's Next?

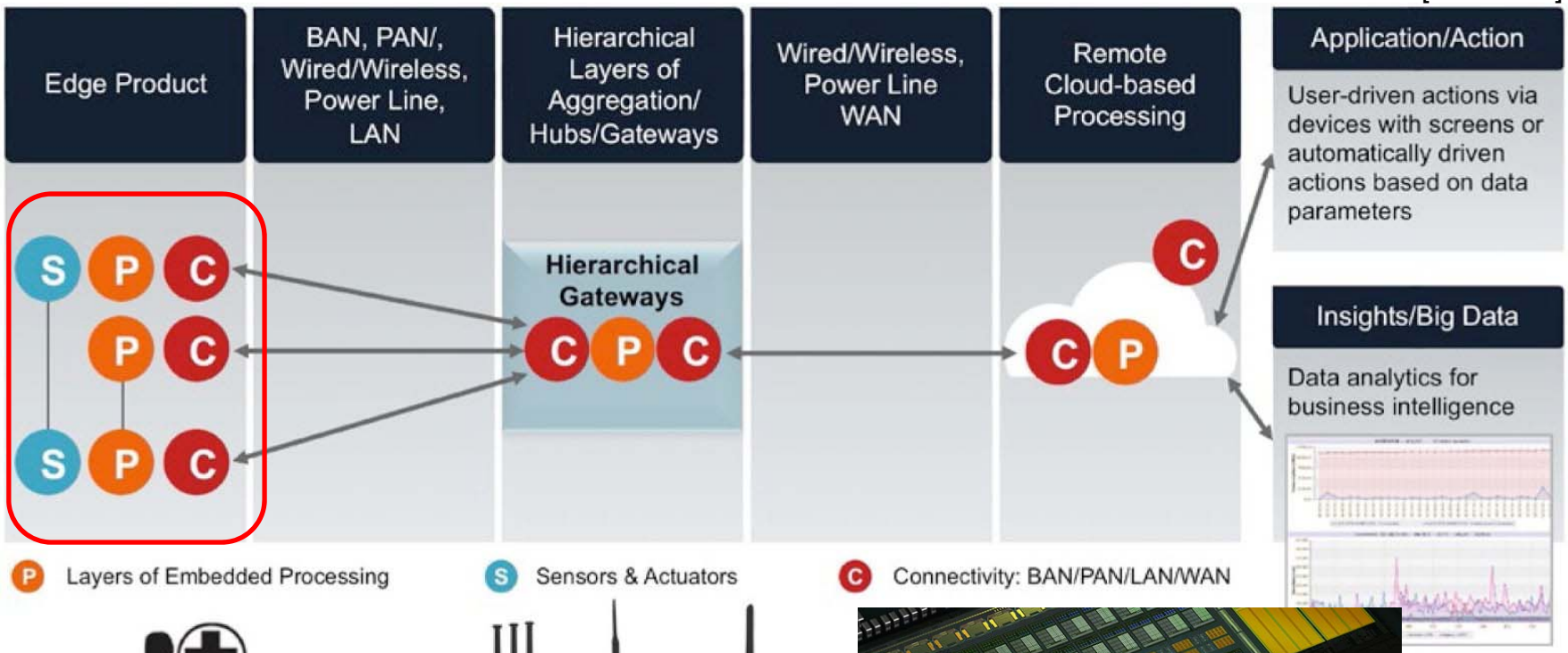
CPS Hierarchical Processing

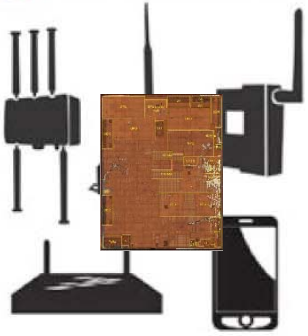
[Freescale]



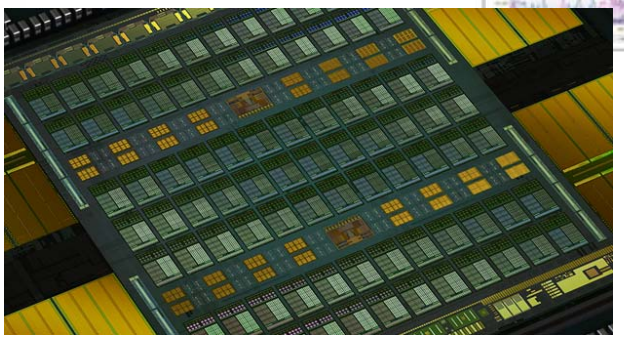
CPS Hierarchical Processing

[Freescale]



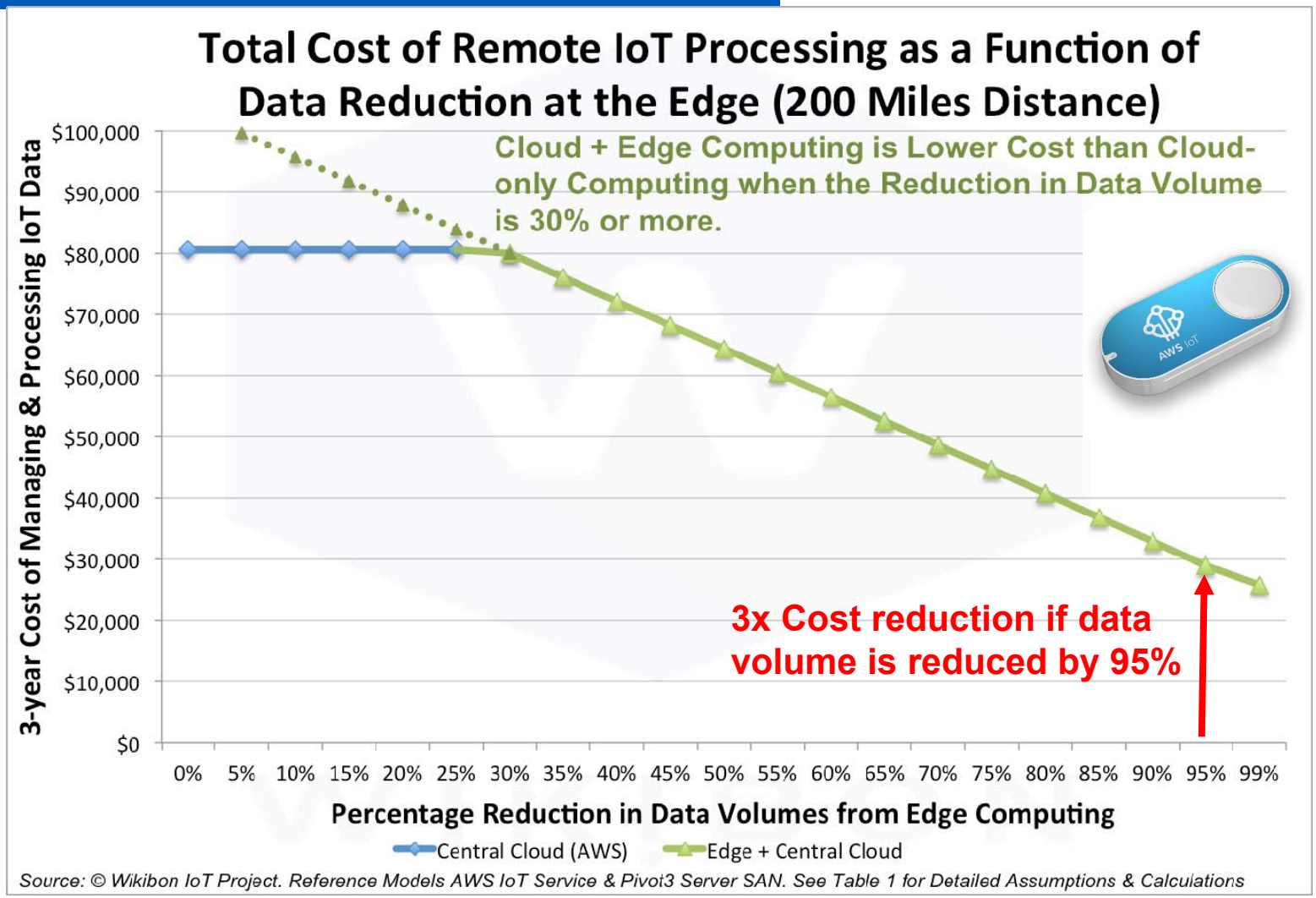


Apple A11 90mm2 TSMC10nm

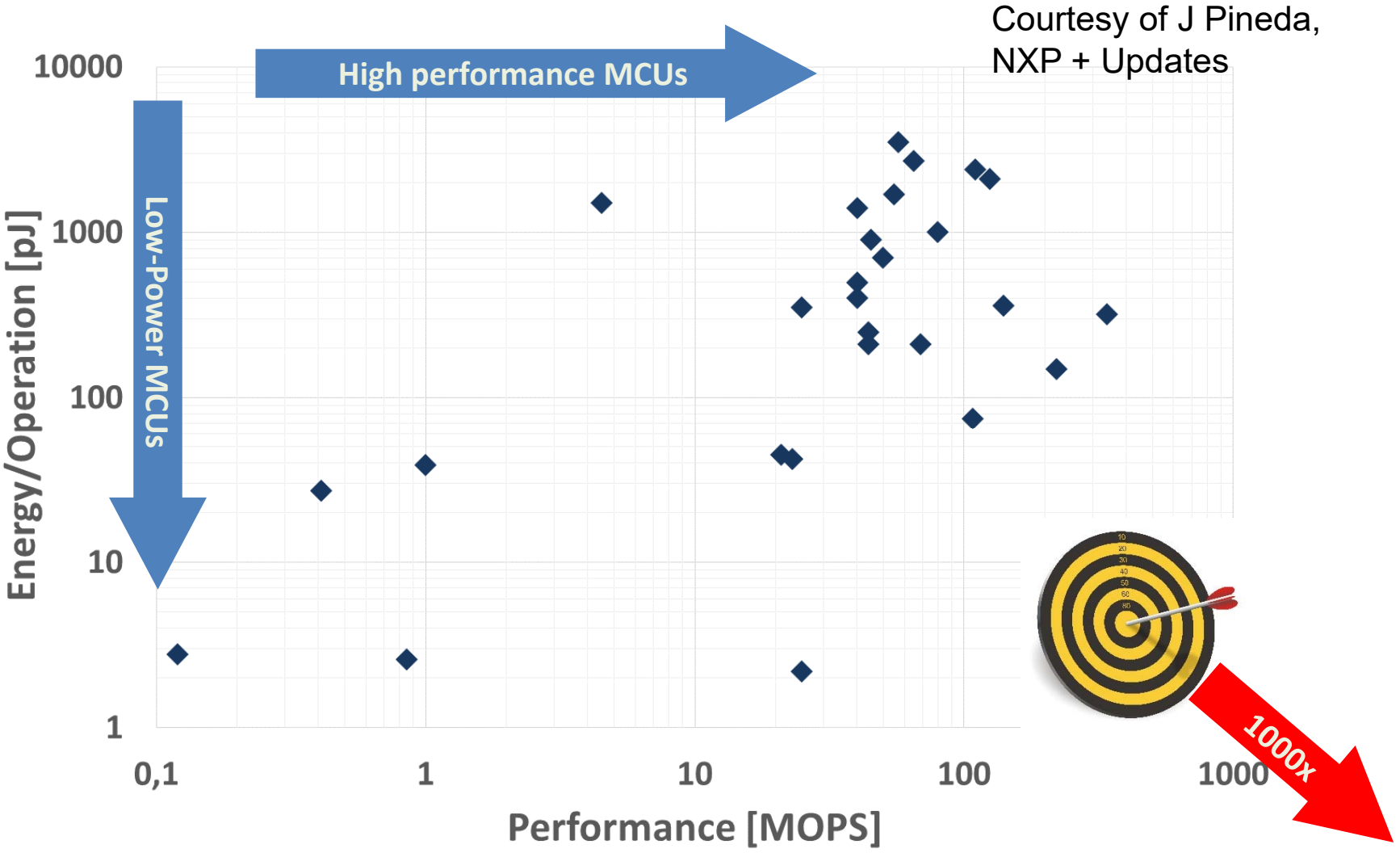


NVIDIA V100 815mm2 TSMC12nm

Does it Matter?



CNN on MCUs?

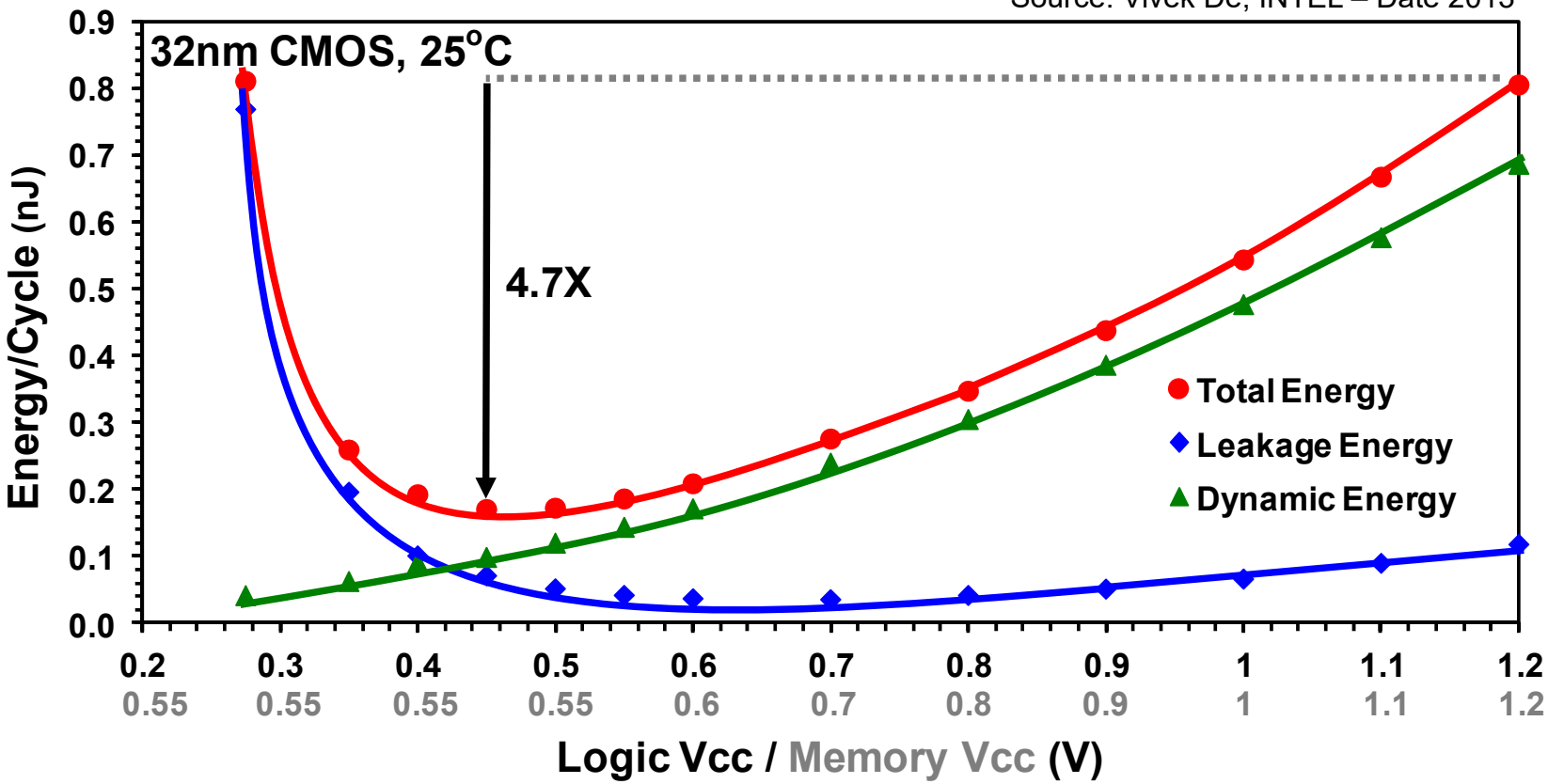

Courtesy of J Pineda,
NXP + Updates
High performance MCUs
Low-Power MCUs
Energy/Operation [pJ]
10000
1000
100
10
1
0,1
1
10
100
1000
Performance [MOPS]
1000x

Outline

- **Near Threshold Multiprocessing**
- Non-Von Neumann Accelerators
- Aggressive Approximation
- From Frame-based to Event-based Processing
- Outlook and Conclusion

Minimum energy operation

Source: Vivek De, INTEL – Date 2013

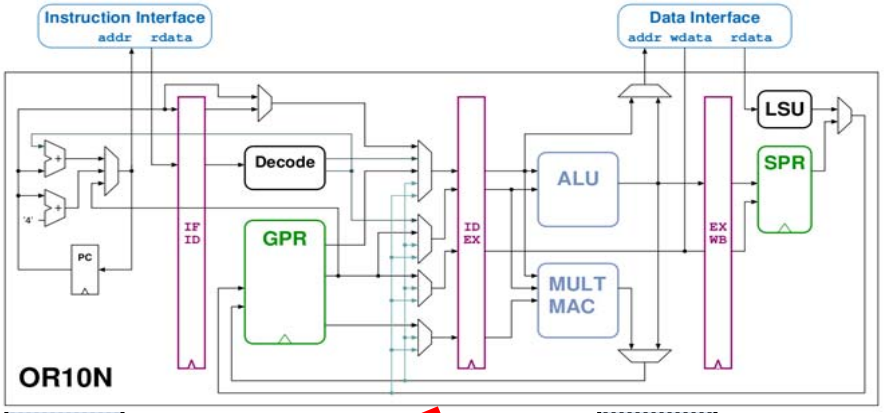


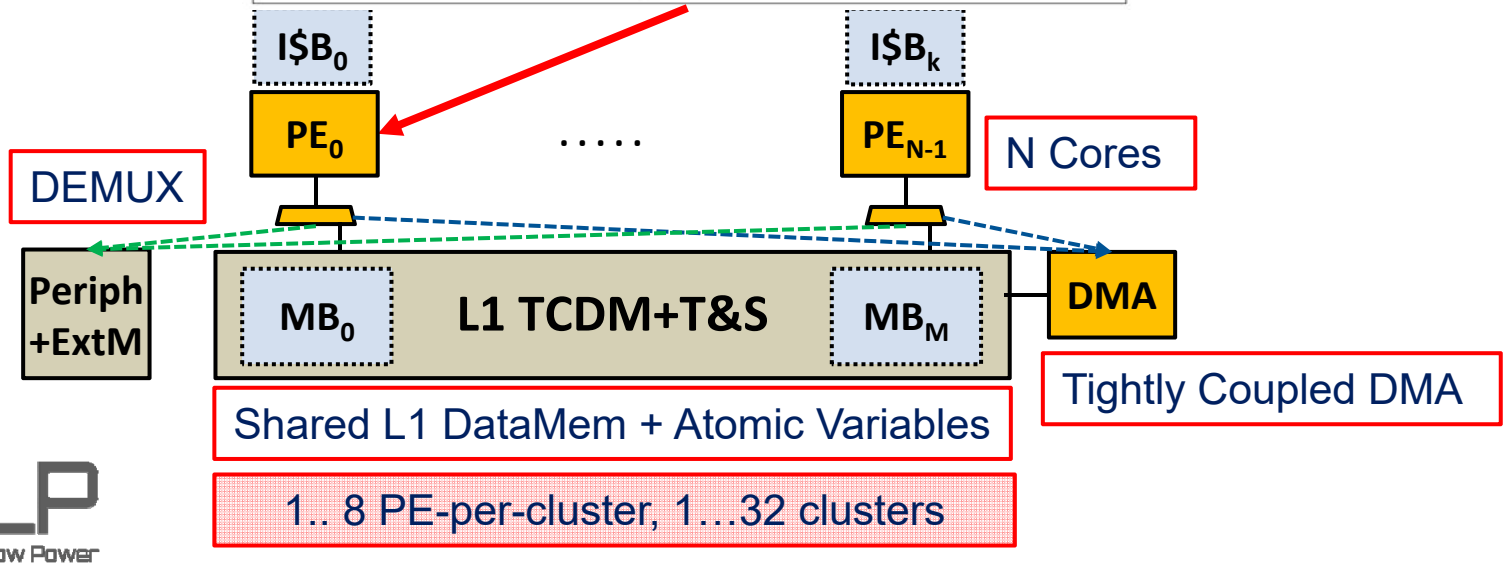
Near-Threshold Computing (NTC):

1. **Don't waste energy pushing devices in strong inversion**
2. **Recover performance with parallel execution**

Near-Threshold Multiprocessing

4-stage OpenRISC & RISC-V RV32IMC





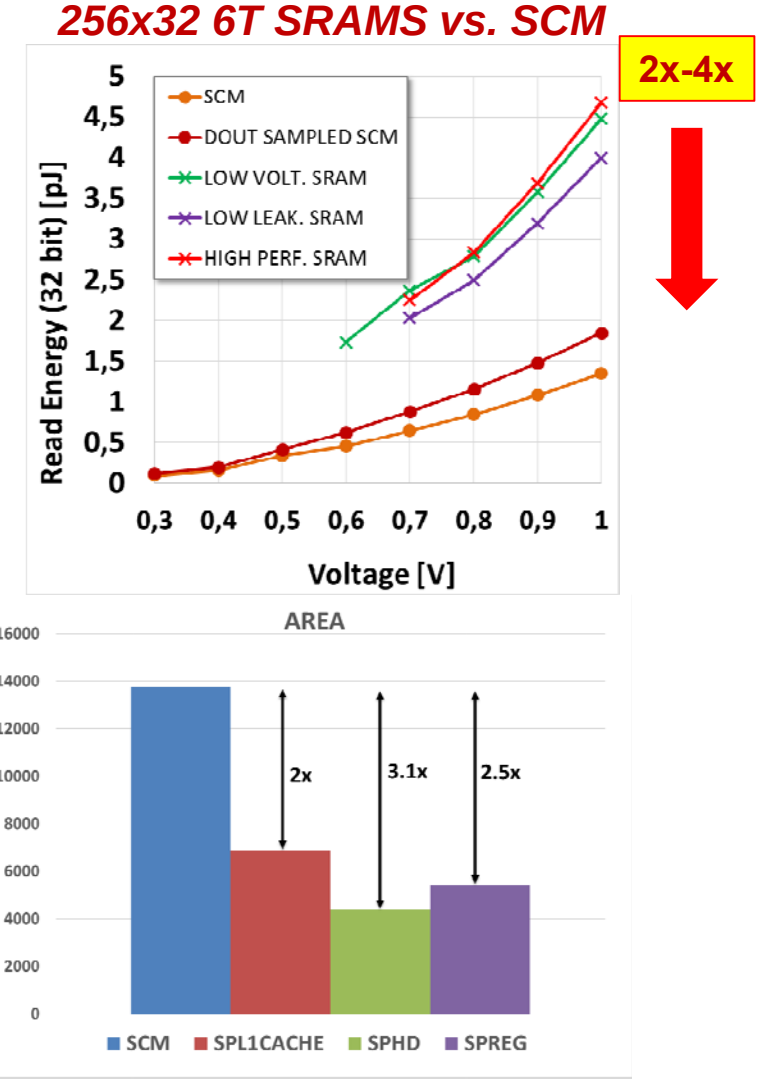
PULP Parallel Ultra Low Power

NT but parallel → Max. Energy efficiency when Active

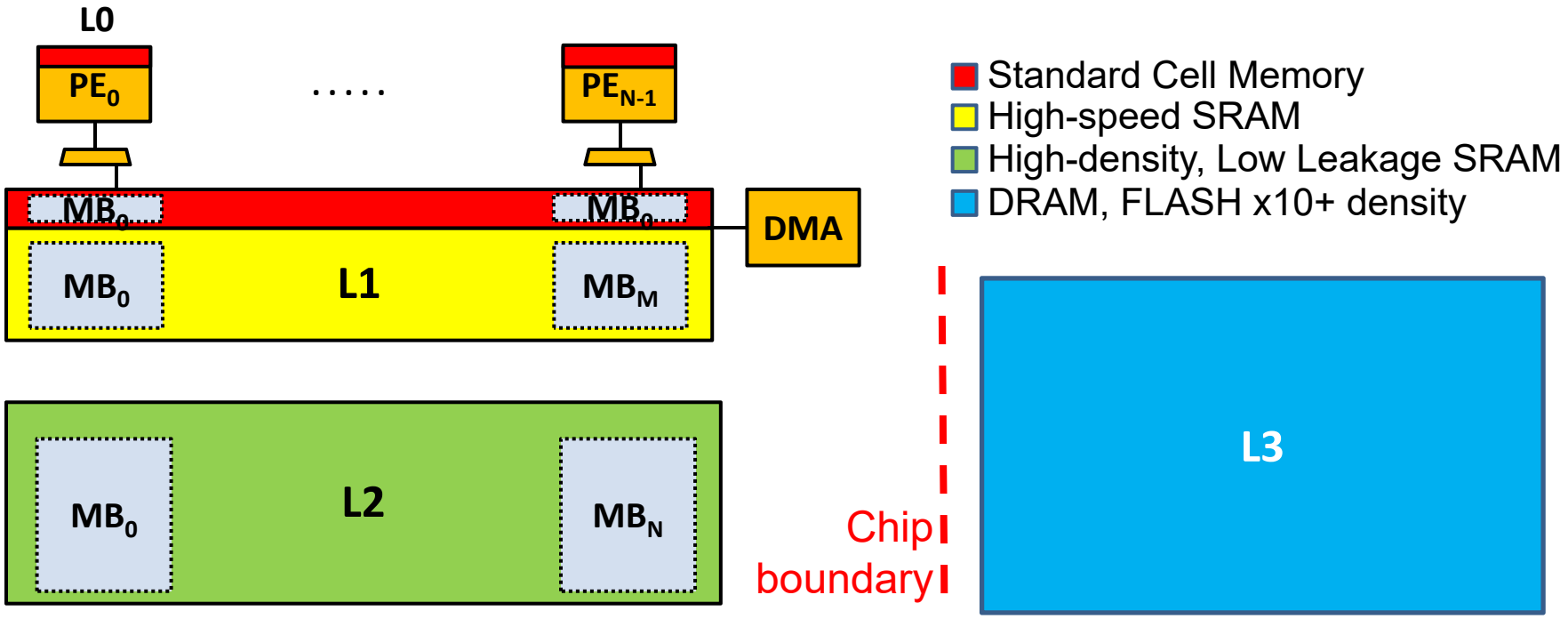
+ strong PM for (partial) idleness

ULP (NT) Bottleneck: Memory

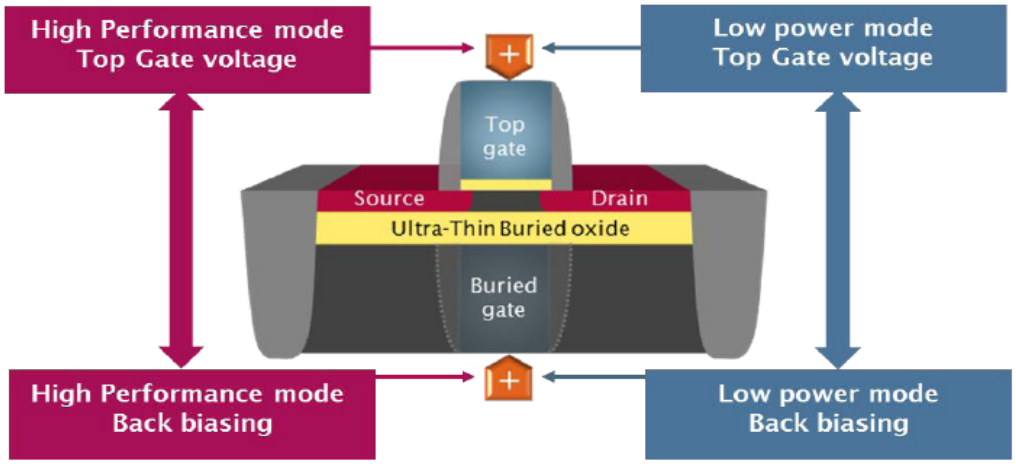
- "Standard" 6T SRAMs:
  - High VDDMIN
  - Bottleneck for energy efficiency
- Near-Threshold SRAMs (8T…)
  - Lower VDDMIN
  - Area/timing overhead (25%-50%)
  - High active energy
  - Low technology portability
- Standard Cell Memories (SCMs):
  - Wide supply voltage range
  - Lower read/write energy (2x - 4x)
  - Easy technology portability
  - Major area overhead (4x)

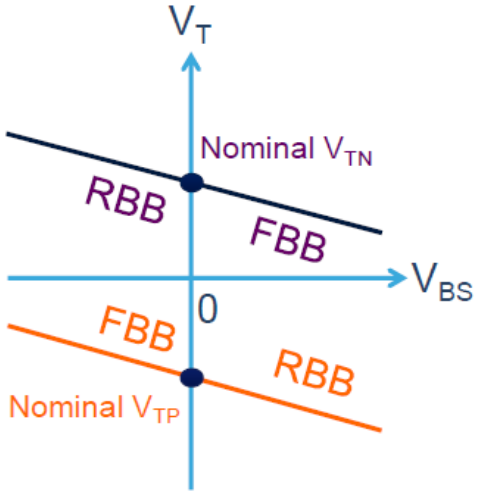


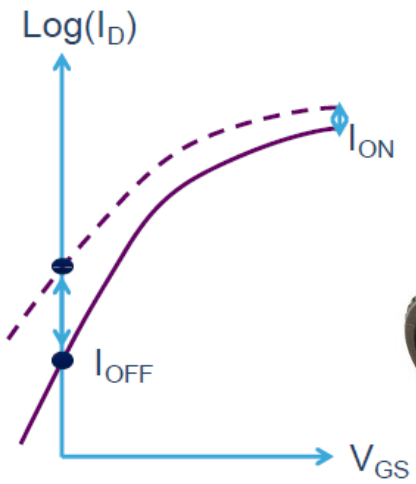
ULP Memory Hierarchy

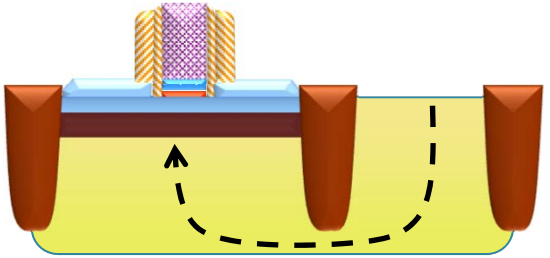


Near threshold FDSOI technology





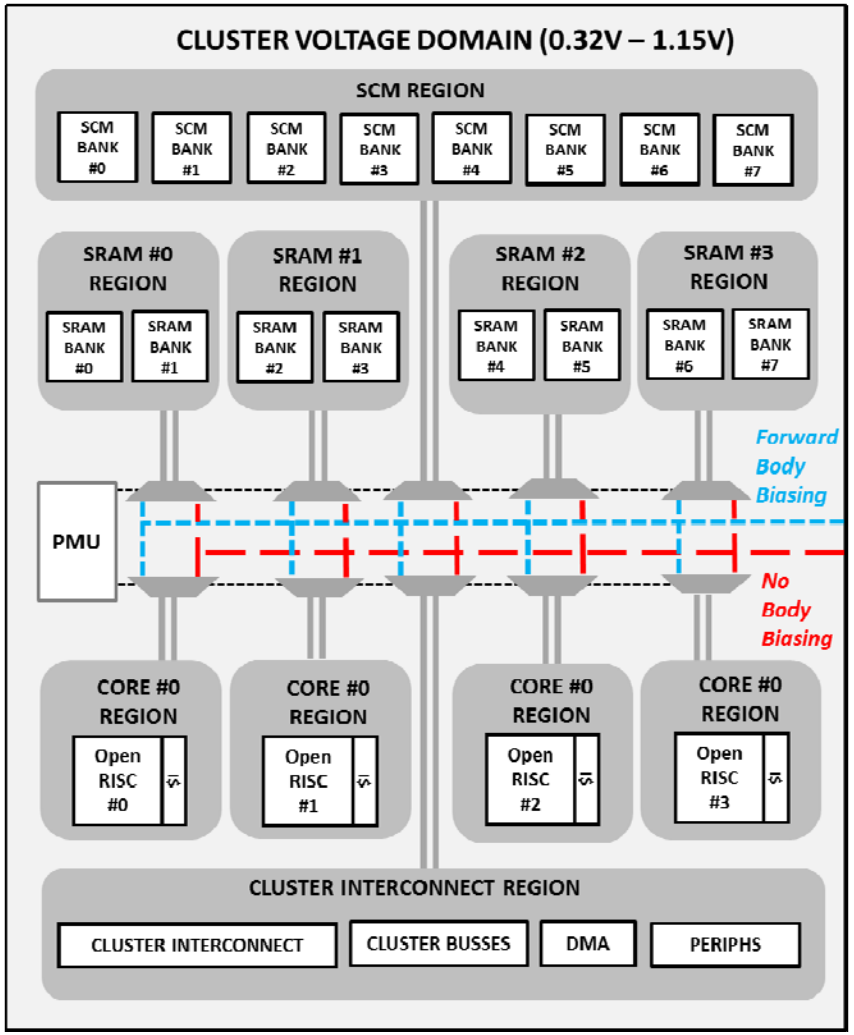




Body bias: Highly effective knob for power & variability management!

Selective, fine-grained BB

- The cluster is partitioned in separate **clock gating and body bias regions**
- **Body bias multiplexers (BBMUXes)** control the well voltages of each region
- A **Power Management Unit (PMU)** automatically manages transitions between the operating modes
- Power modes of each region:
  - **Boost mode:** active + FBB
  - **Normal mode:** active + NO BB
  - **Idle mode:** clock gated + NO BB (in LVT) RBB (in RVT)



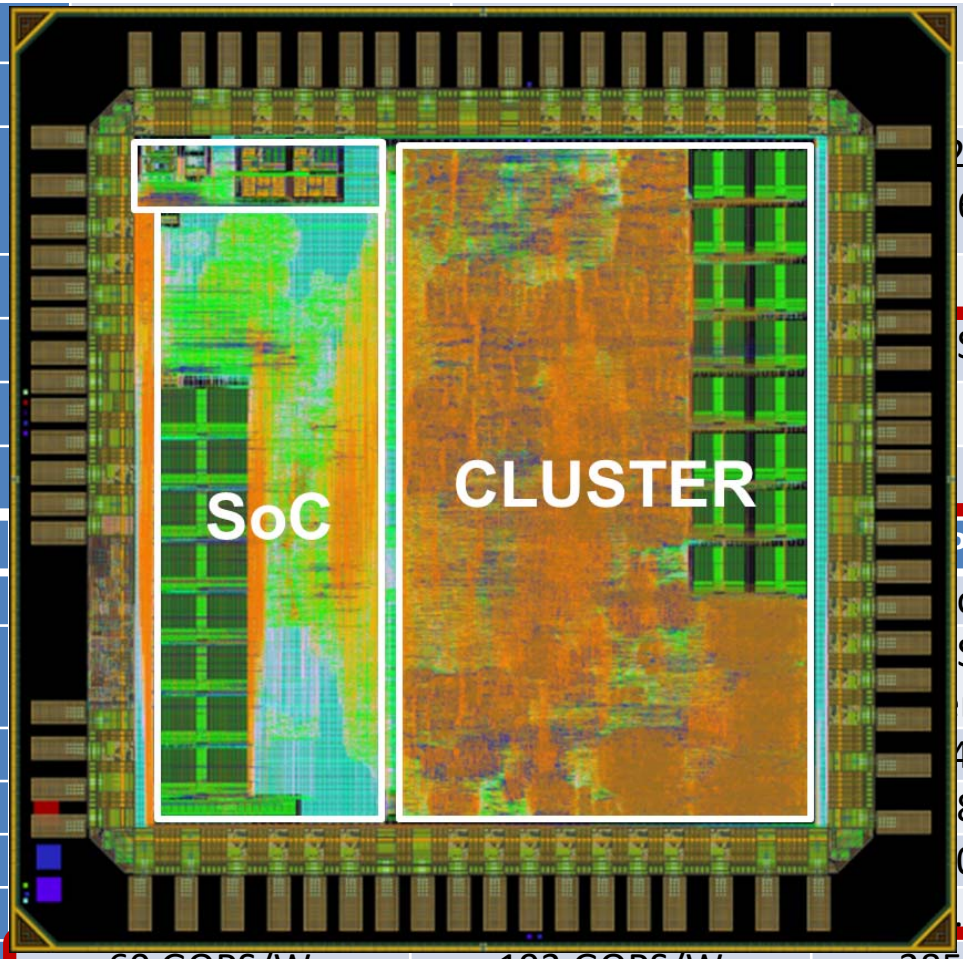
The Evolution of the 'Species'

	PULPv1	PULPv2	PULPv3
# of cores	4	4	4
L2 memory	16 kB	64 kB	64 kB
TCDM	16kB SRAM	32kB SRAM 8kB SCM	32kB SRAM 16kB SCM
DVFS	no	yes	yes
I$	4kB SRAM private	4kB SCM private	4kB SCM shared
DSP Extensions	no	no	yes
HW Synchronizer	no	no	yes

	PULPv1	PULPv2	PULPv3
Status	silicon proven	silicon proven	silicon proven
Technology	FD-SOI 28nm conventional-well	FD-SOI 28nm flip-well	FD-SOI 28nm conventional-well
Voltage range	0.45V - 1.2V	0.32V - 1.2V	0.4V - 0.7V
BB range	-1.8V - 0.9V	0.0V - 1.8V	-1.8V - 0.5V
Max freq.	475 MHz	850 MHz	200 MHz
Max perf.	1.9 GOPS	3.3 GOPS	1.8 GOPS
Peak en. eff.	60 GOPS/W	193 GOPS/W	385 GOPS/W

The Evolution of the 'Species'

	PULPv1	PULPv2	PULPv3
# of cores			4
L2 memory			64 kB
TCDM			2kB SRAM 6kB SCM
DVFS			yes
I$			SCM shared
DSP Extensions			yes
HW Synchronizer			yes
			PULPv3
Status			on proven
Technology			SOI 28nm ntional-well
Voltage range			4V - 0.7V
BB range			8V - 0.5
Max freq.			00 MHz
Max perf.			8 GOPS
Peak en. eff.	60 GOPS/W	193 GOPS/W	385 GOPS/W

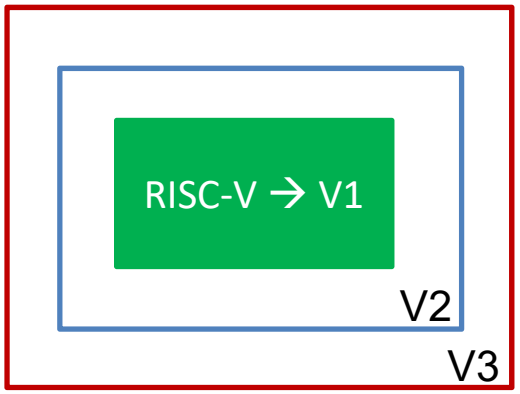


2.6pj/OP

Extending RISC-V for CNNs

<32-bit precision → SIMD2/4 opportunity

1. HW loops and Post modified LD/ST
2. Bit manipulations
3. Packed-SIMD ALU operations with dot product
4. Rounding and Normalizazion
5. Shuffle operations for vectors



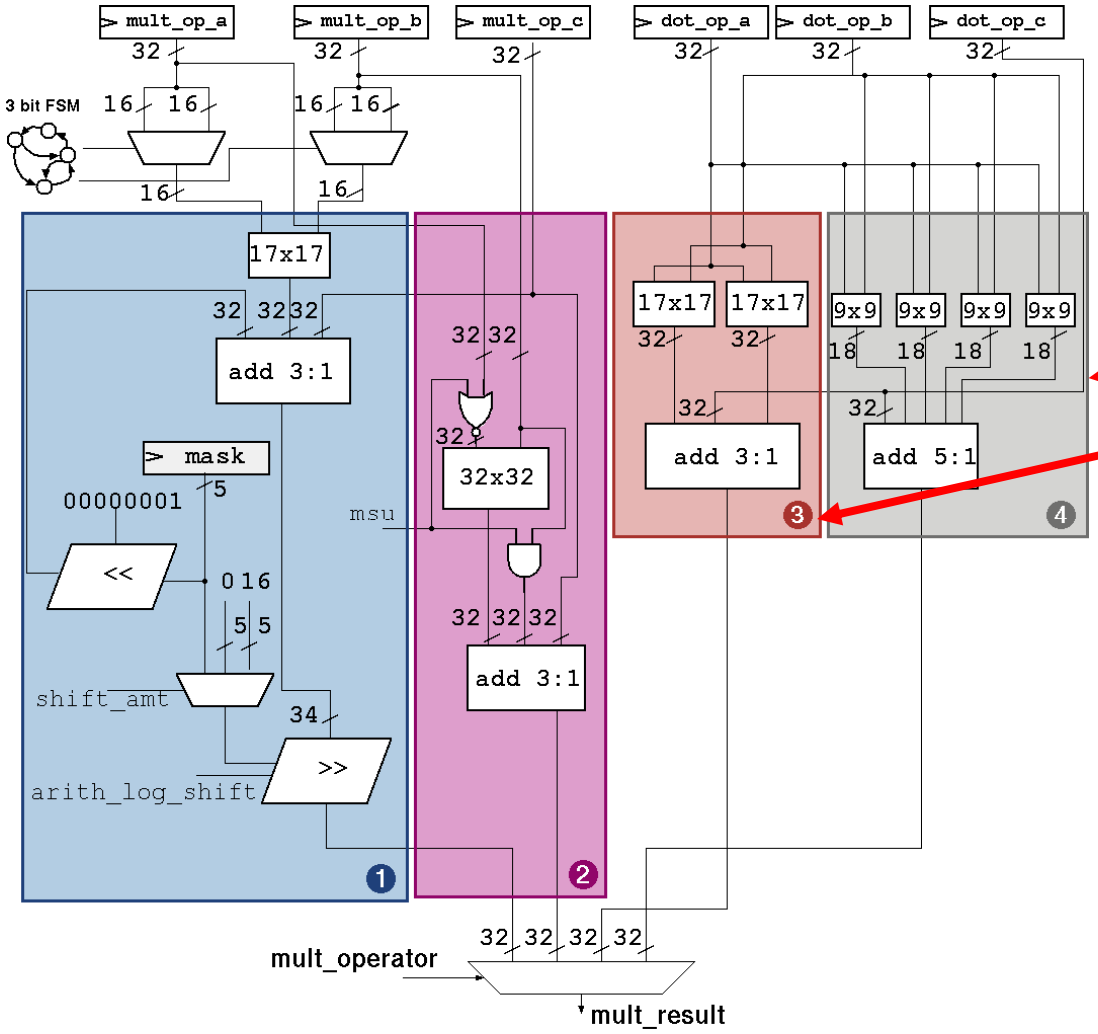
V1 Baseline RISC-V RV32IMC

V2
HW loops
Post modified Load/Store
Mac

V3
SIMD 2/4 + DotProduct + Shuffling
Bit manipulation unit
Lightweight fixed point

Small Power and Area overhead

Dot Product SIMD



- Processor datapath

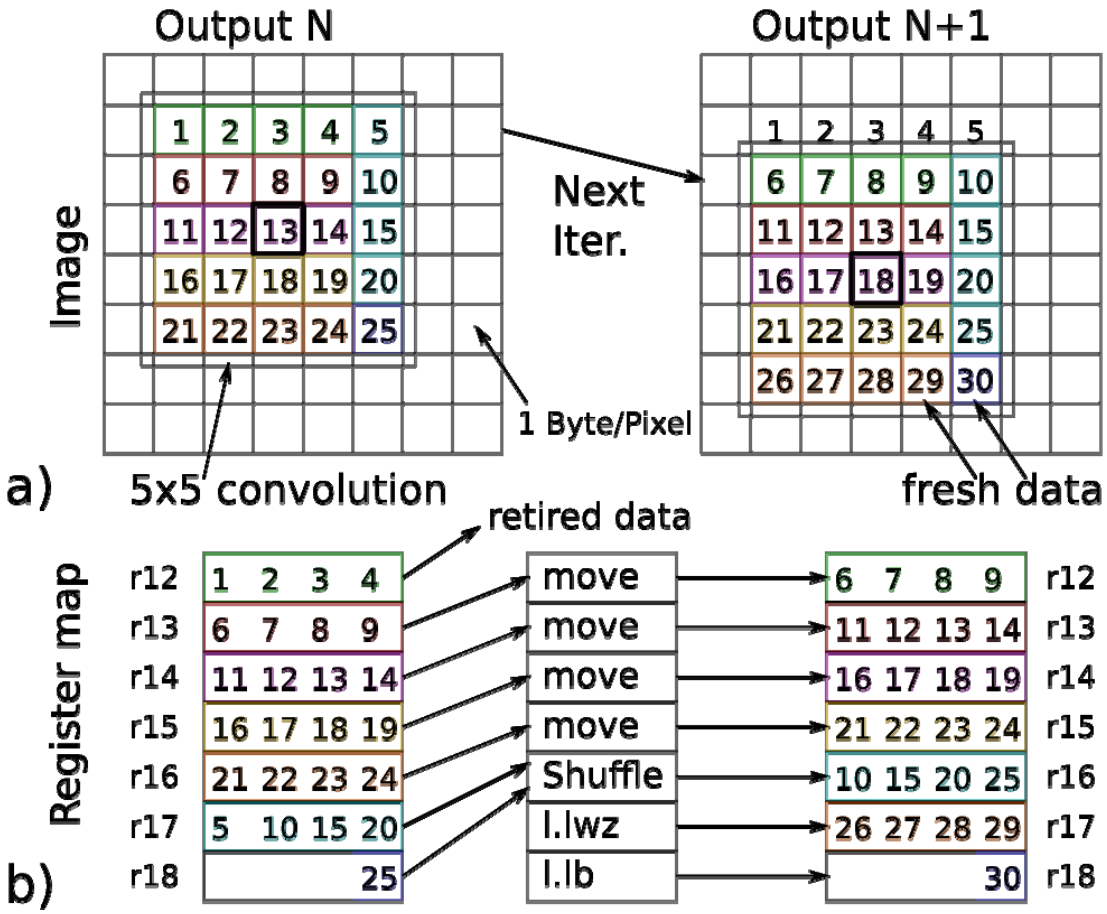
Sum-of-dot-product units:

8b version

16b version

1 cycle execution

Shuffle



- Convolution in registers
- 5x5 convolutional filter

- 7 Sum-of-dot-product
- 4 move
- 1 shuffle
- 3 lw/sw
- ~ 5 control instructions

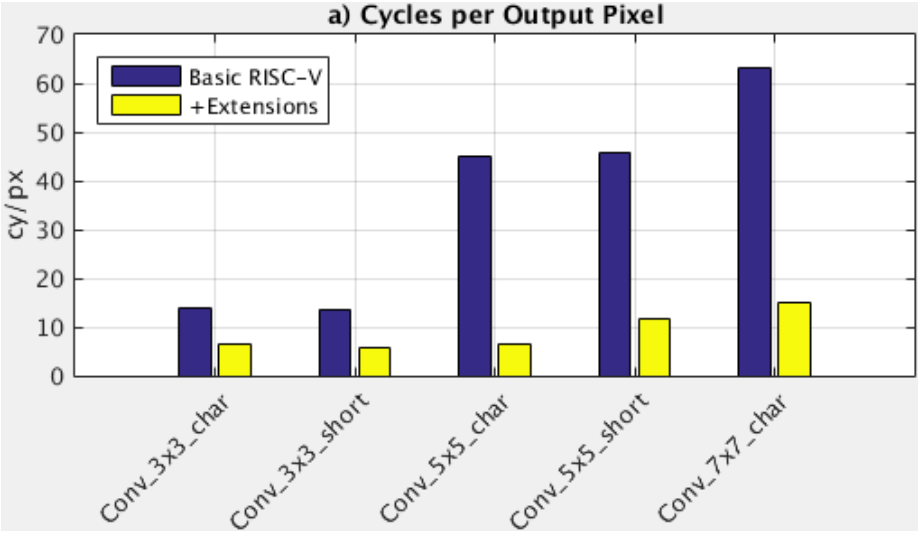
20 instr. / output pixel → Scalar version >100 instr. / output pixel

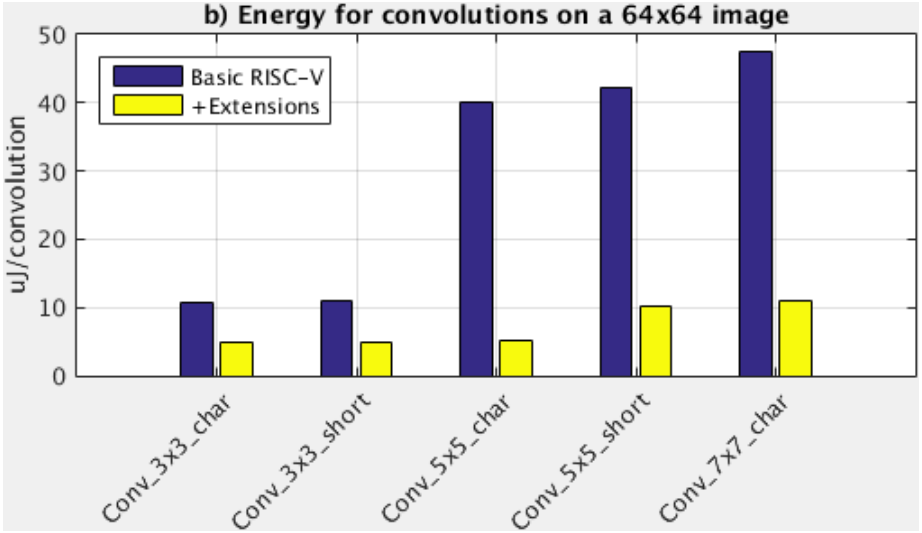
PULP-CNN ISA-Extensions

Convolution Performance on PULP with 4 cores

speedup: 3.9x

Energy gains: 4.2x



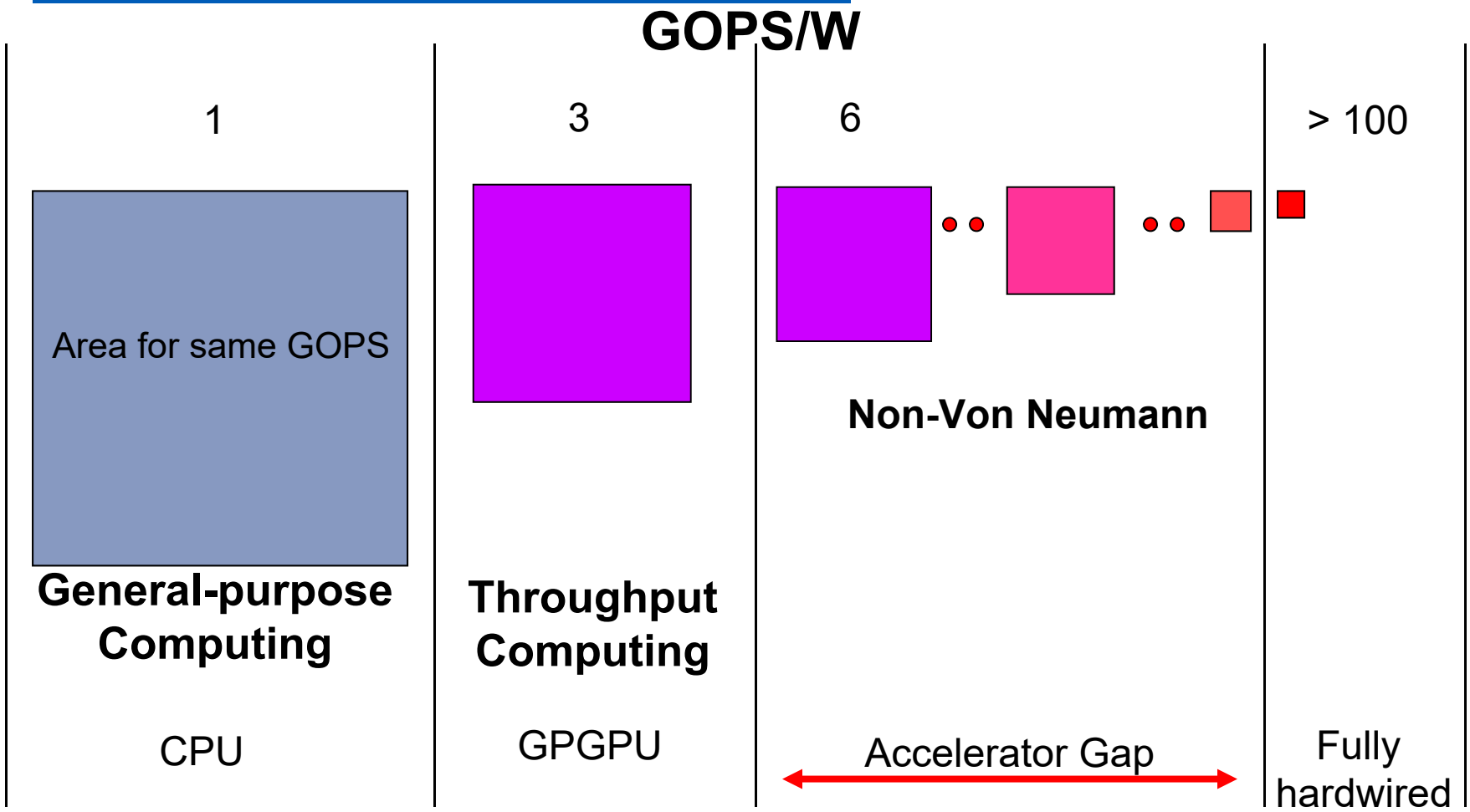


5x5 convolution in only 6.6 cycles/pixel

Outline

- Near Threshold Multiprocessing
- **Non-Von Neumann Accelerators**
- Aggressive Approximation
- From Frame-based to Event-based Processing
- Outlook and Conclusion

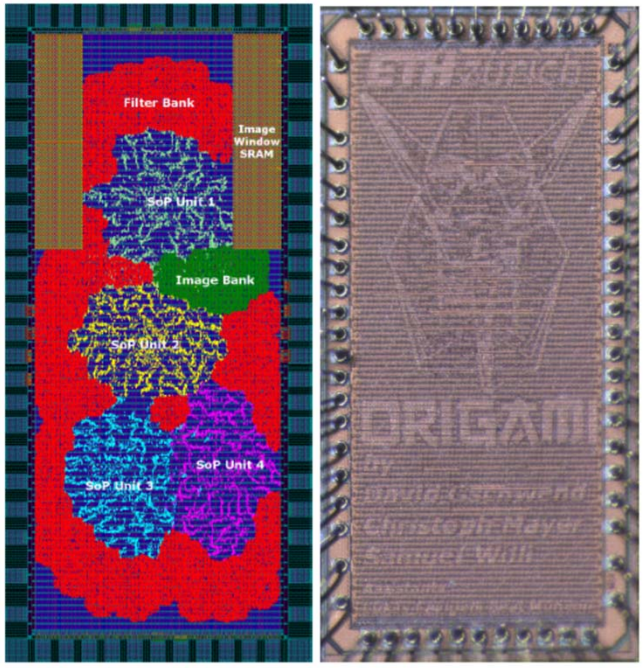
Recovering silicon efficiency

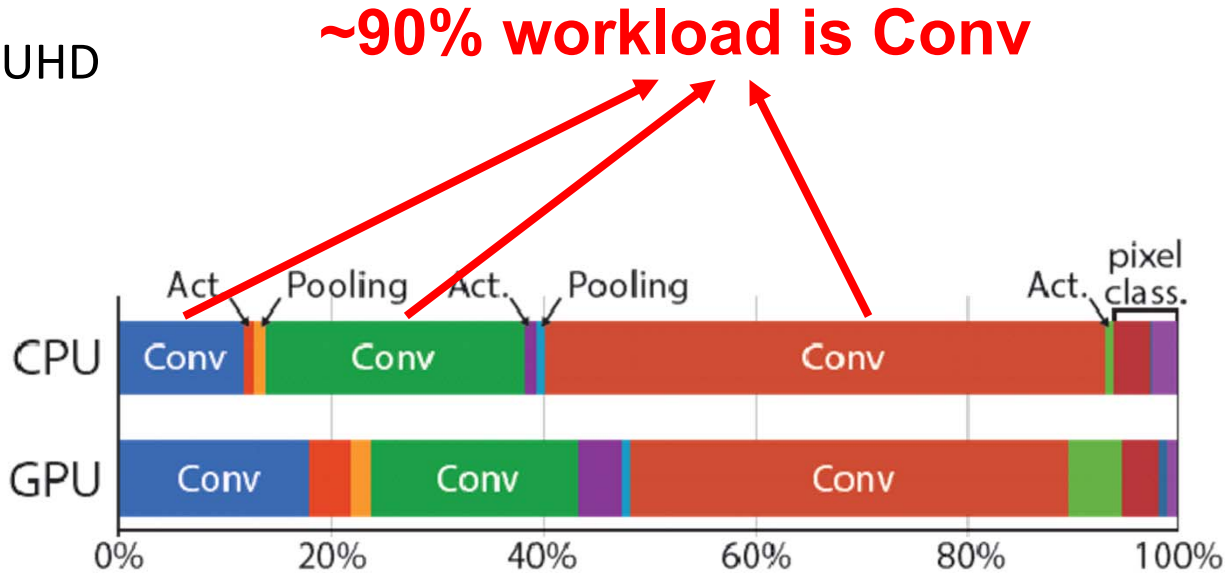


Closing The Accelerator Efficiency Gap with Agile Customization

Computational Effort

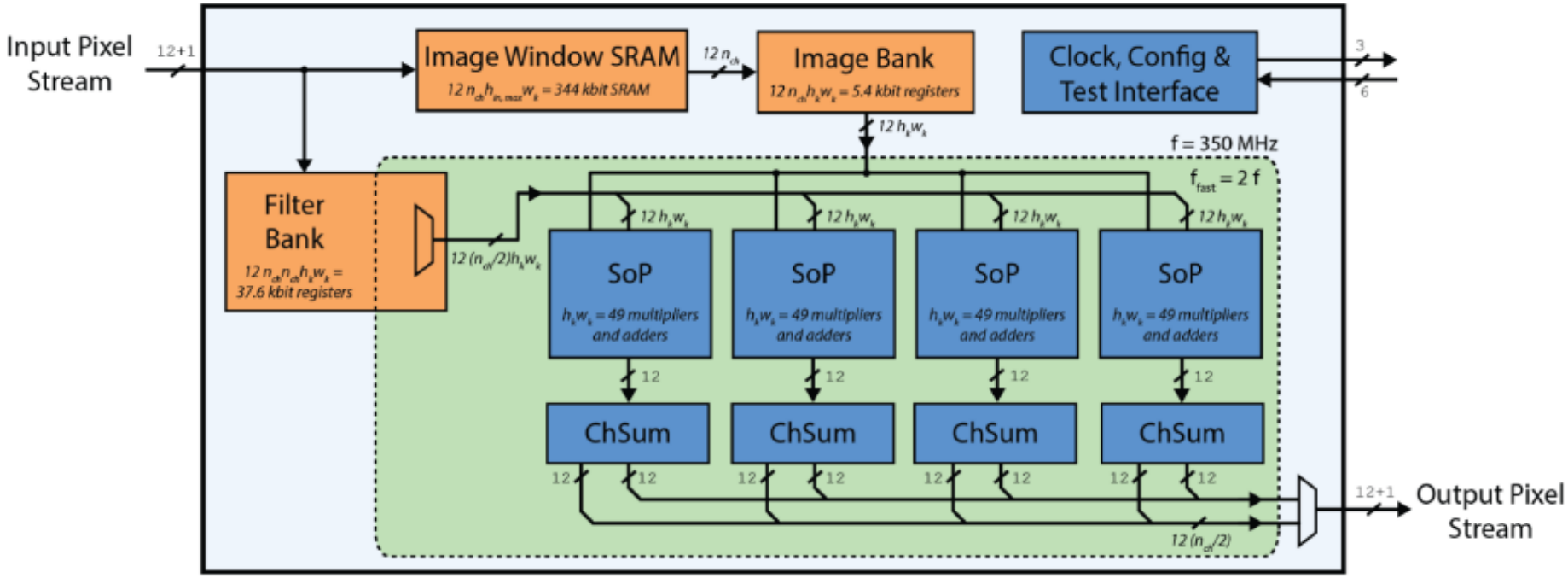
- Computational effort
  - 10-class scene labeling on Stanford-BG
  - 7.5 GOp/frame for 320x240 image (#Op=2 × #MAC)
  - 260 GOp/frame for FHD
  - 1050 GOp/frame for 4k UHD



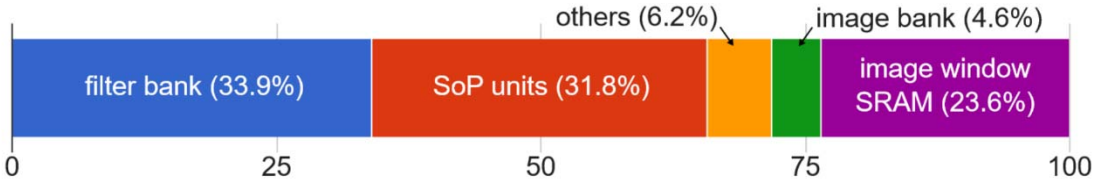


Origami CNN ASIC

Origami: A CNN Accelerator



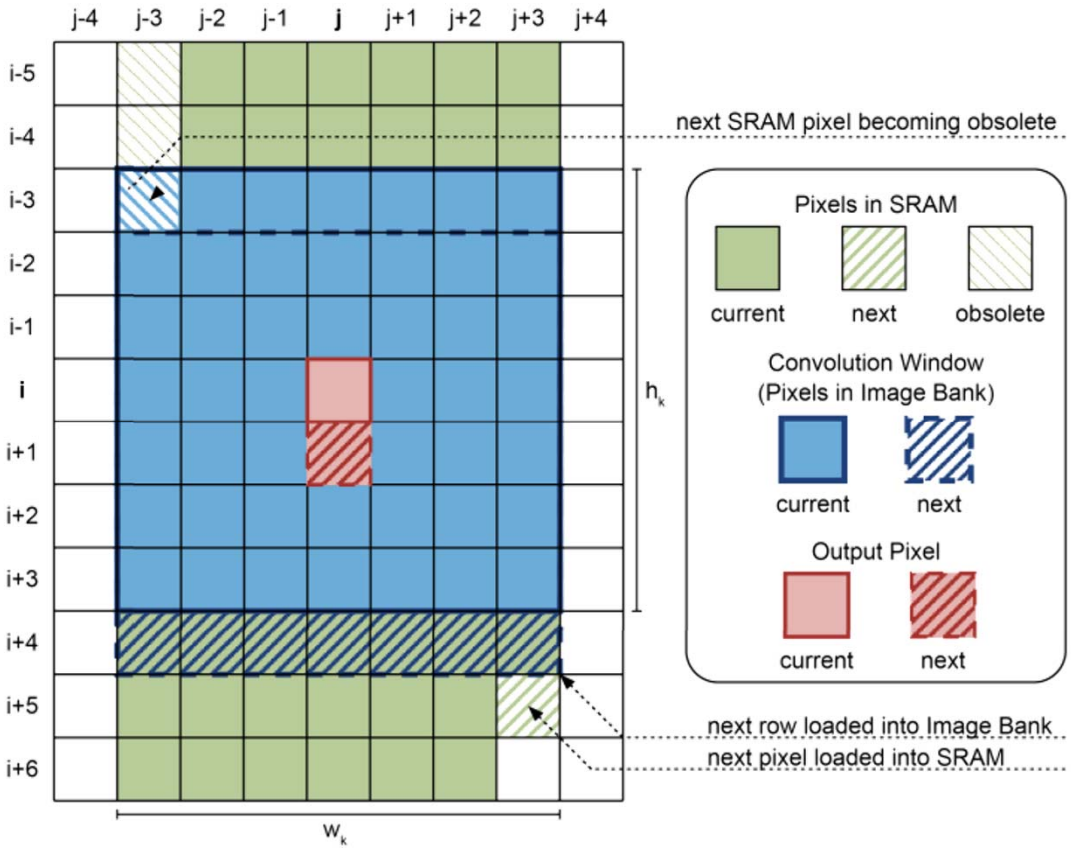
- **FP not needed**: 12-bit signals sufficient
- Input to classification double-vs-12-bit accuracy loss $< 0.5\%$ (80.6% to 80.1%)



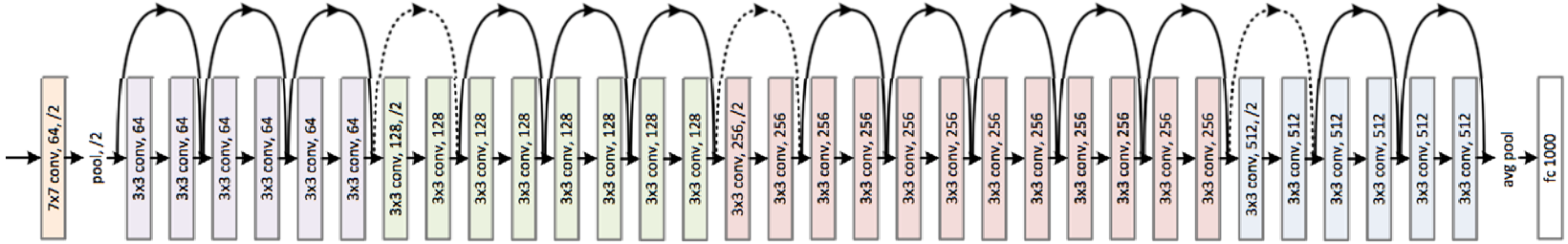
Origami: The Architecture

1. Keep moving window of all 8 input channels buffered on-chip
2. Perform all the convolutions
3. Transmit result immediately after summing up

Maximize locality and data reuse (weights and pixels)



CNNs: typical workload



Example: **ResNet-34**

- classifies 224x224 images into 1000 classes
- ~ trained human-level performance
- ~ **21M** parameters
- ~ **3.6G MAC** operations

Scaling Origami to 28nm FDSOI

Performance for 10 fps: **~73 GOPS/s**

Energy efficiency: **~2300 GOPS/W** efficiency

0.4pj/OP

Origami core in 28nm FDSOI → 10 fps ResNet-34 with ~32mW

Outline

- Near Threshold Multiprocessing
- Non-Von Neumann Accelerators
- **Aggressive Approximation**
- From Frame-based to Event-based Processing
- Outlook and Conclusion

Pushing Further: YodaNN[1]

- Approximation at the algorithmic side → Binary weights
- BinaryConnect [Courbariaux, NIPS15], XOR NET [Rastegari, arXiv16]
  - Reduce weights to a binary value -1/+1
  - Stochastic Gradient Descent with Binarization in the Forward Path

$$w_{b,stoch} = \begin{cases} -1 & p_{-1} = \sigma(w) \\ 1 & p_1 = 1 - p_{-1} \end{cases} \qquad w_{b,det} = \begin{cases} -1 & w < 0 \\ 1 & w > 0 \end{cases}$$

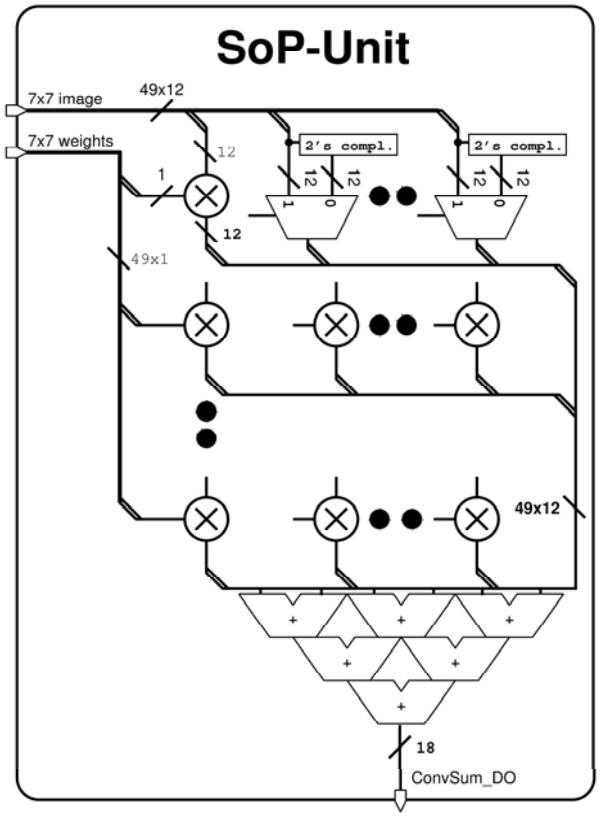
  - Learning large networks is still challenging, but starts to become feasible:

    ResNet-18 on ImageNet with **83.0%** (binary-weight) vs. **89.2%** (single-precision) top-5 accuracy; and **60.8%** vs. **69.3%** top-1 accuracy

- Ultra-optimized HW is possible!
  - Major arithmetic density improvements: MAC → 2s compl. & Accum.
    - Area can be used for more energy-efficient weight storage
  - Storage reduction → SCM memories for lower voltage → E goes with $1/V^2$

[1]After the Yedi Master from Star Wars - "Small in size but wise and powerful" cit. www.starwars.com

SoP-Unit Optimization



Equivalent for 7x7 SoP

ImageBank

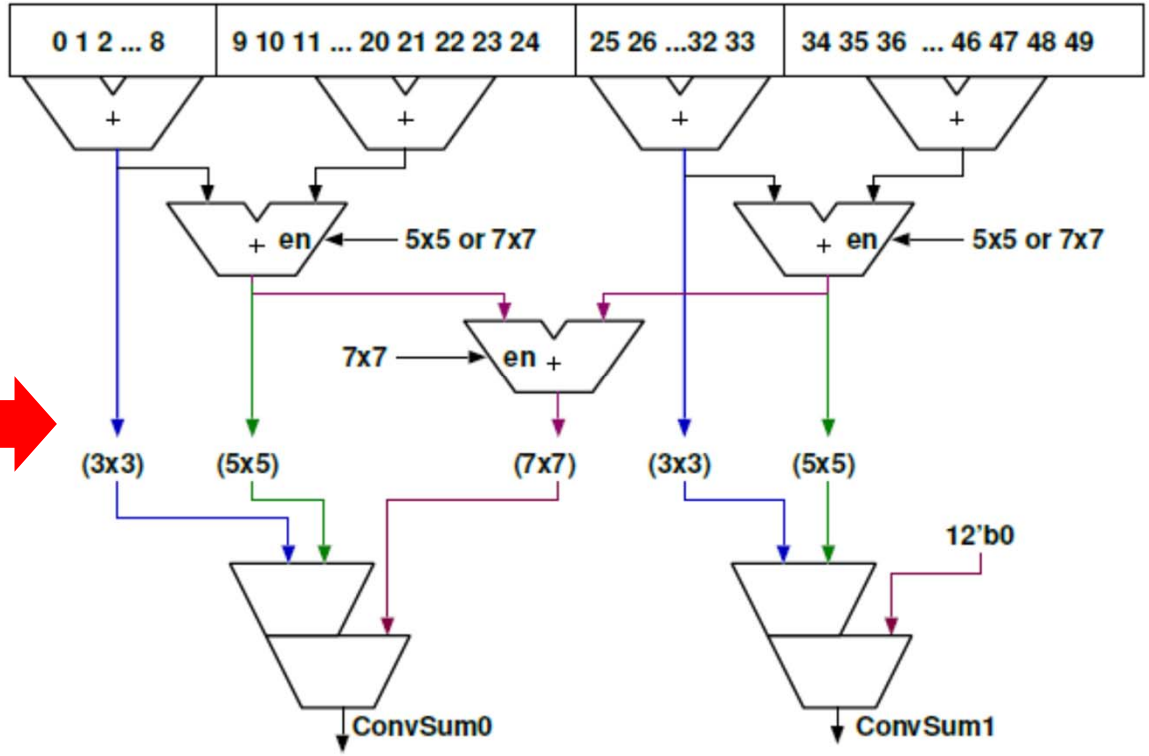
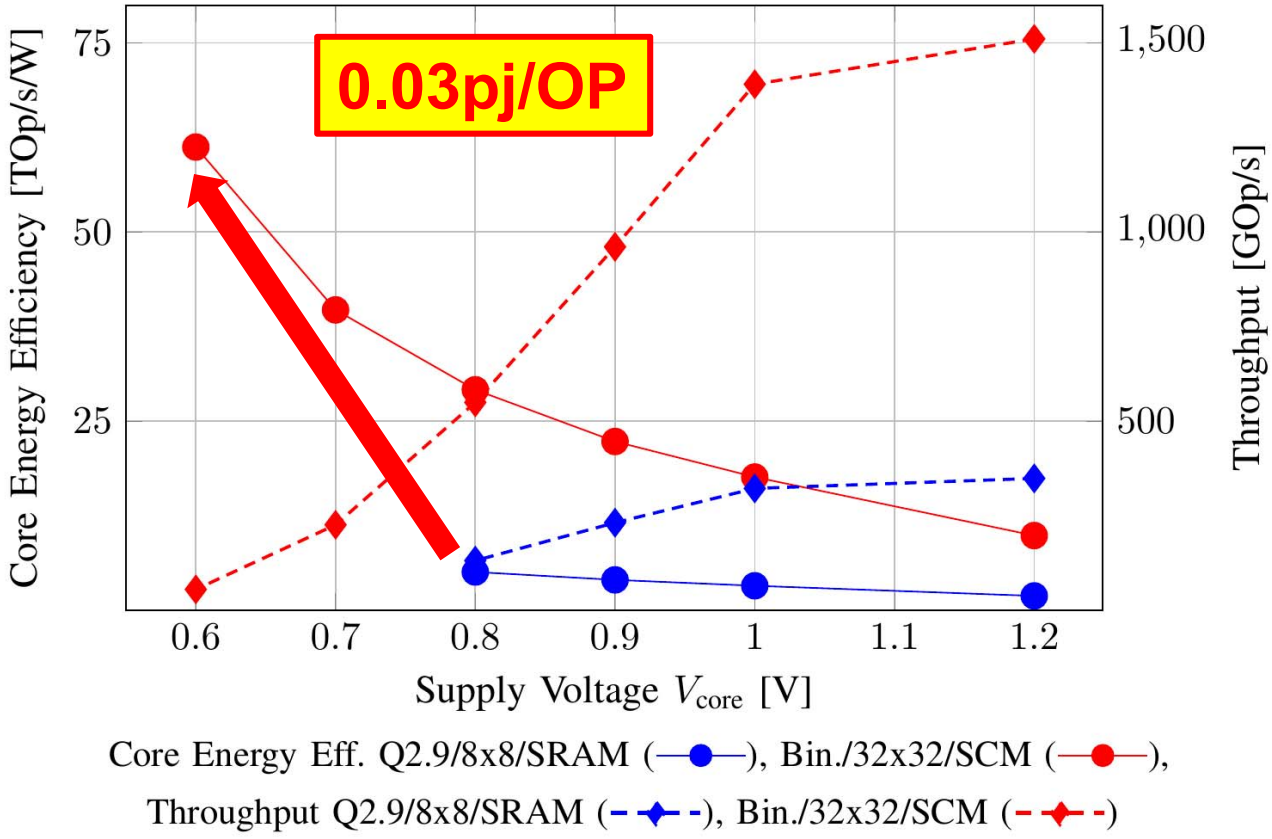


Image Mapping (3x3, 5x5, 7x7)

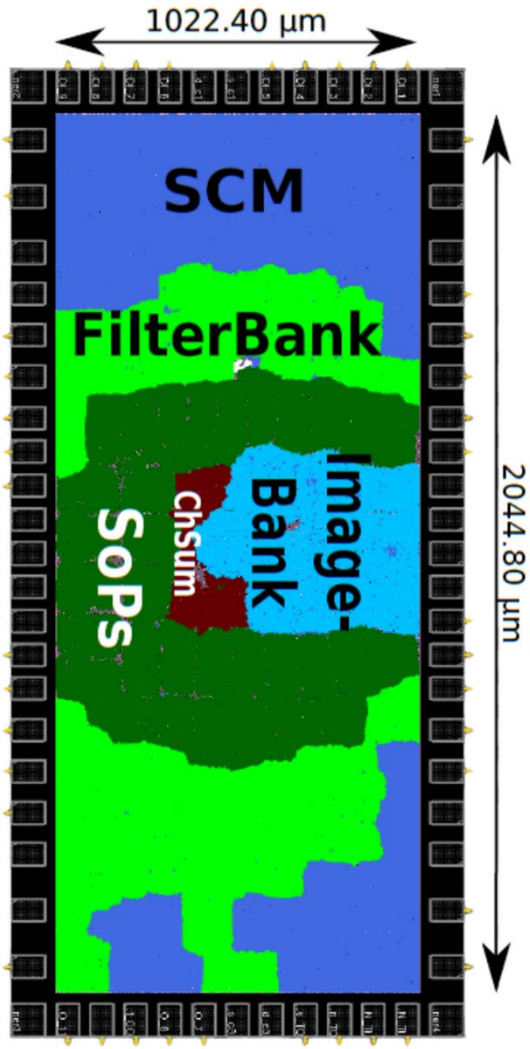
1 MAC Op = 2 Op (1 Op for the “sign-reverse”, 1 Op for the add).

YODANN Energy Efficiency

Same area 8→32 SoP units + all SCM



12x Boost in core energy efficiency (single layer)



Origami, YodaNN vs. Human

The «energy-efficient AI» challenge (e.g. Human vs. IBM Watson)

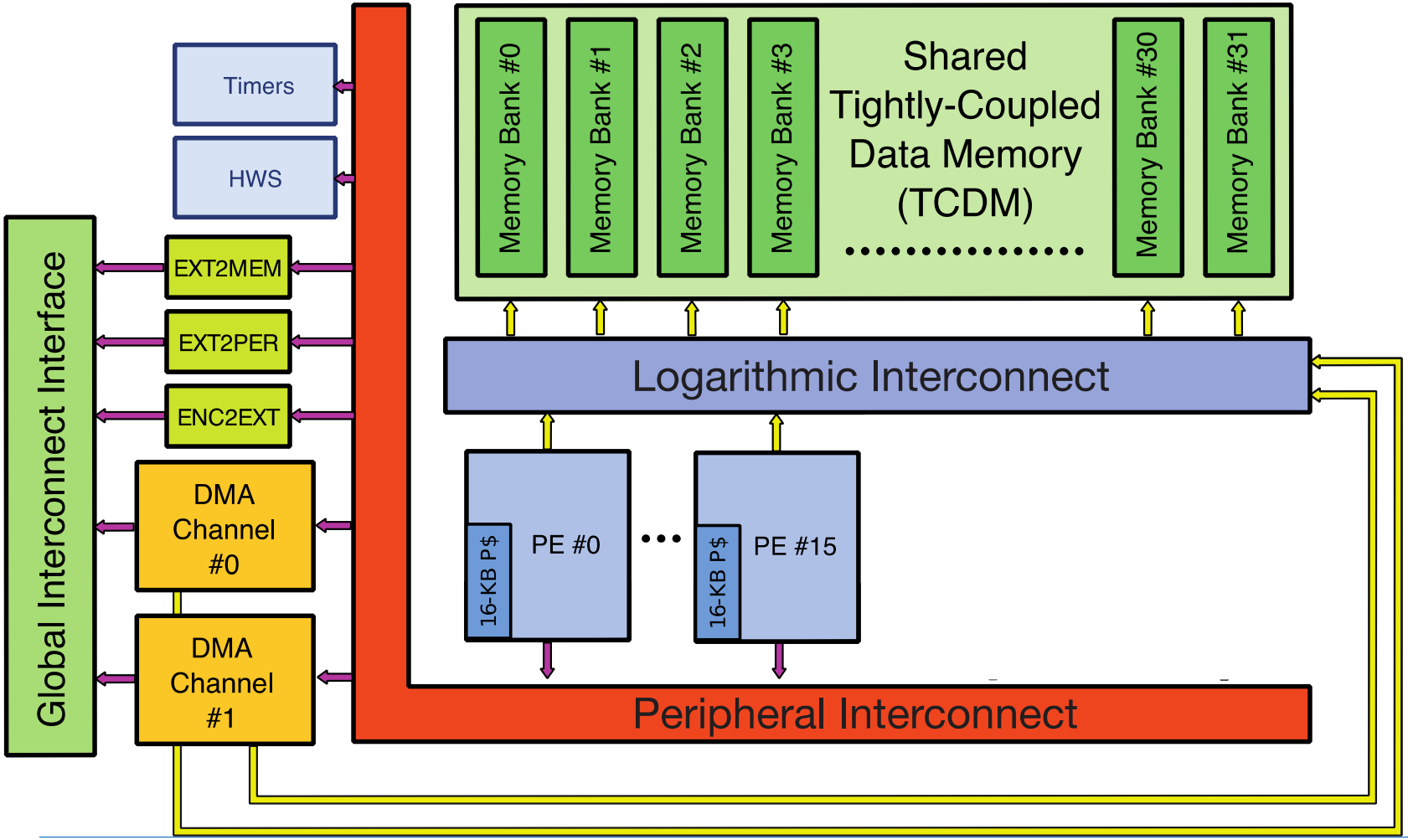
Type	Analog (bio)	Q2.9 Precision	Q2.9 Precision	Binary-Weight
Network	human	ResNet-34	ResNet-18	ResNet-18
Top-1 error [%]		21.53	30.7	39.2
Top-5 error [%]	**5.1**	5.6	10.8	17.0
Hardware	Brain	Origami	Origami	YodaNN
Energy-eff. [uJ/img]	100.000(*)	1086	543	**31**

*P_{brain} = 10W, 10% of the brain used for vision, trained human working at 10img/sec

- Game over for humans also in energy-efficient vision?
- …. Not yet! (object recognition is a super-simple task)

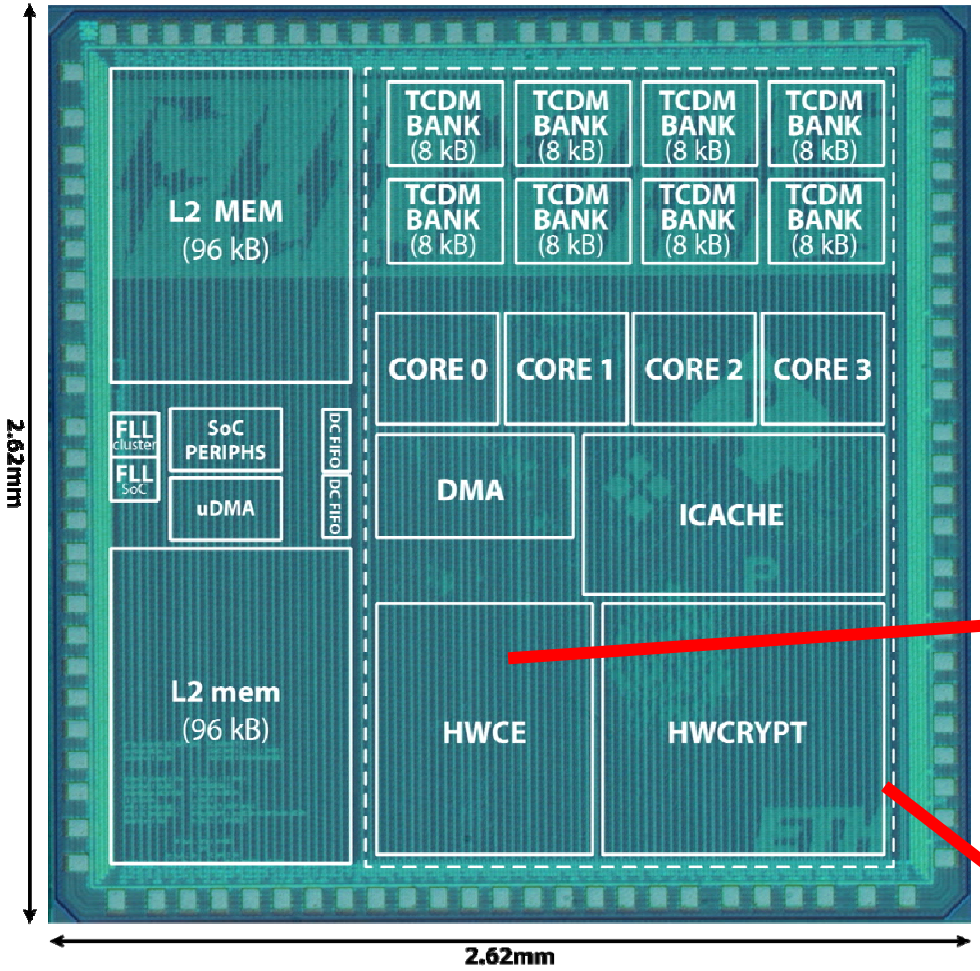
Heterogeneous PULP Cluster

Hardware Convolutional Engine (HWCE) in the Cluster



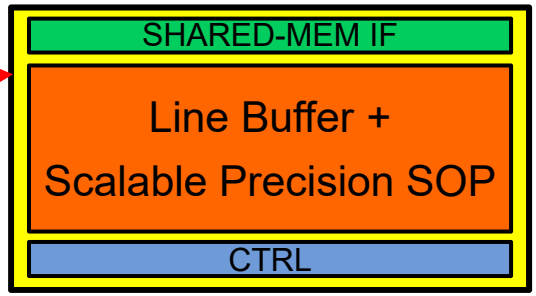
Heterogeneous PULP Cluster

Hardware Convolutional Engine (HWCE) in the Cluster



- **~6.9 mm2** PULP chip, 1/2016
- 4 cores
- 1 HWCRYPT accelerator
- 1 HWCE for 3D conv layers
- 64kB L1, 192kB L2
- QSPI (M/S), I2C, I2S, UART

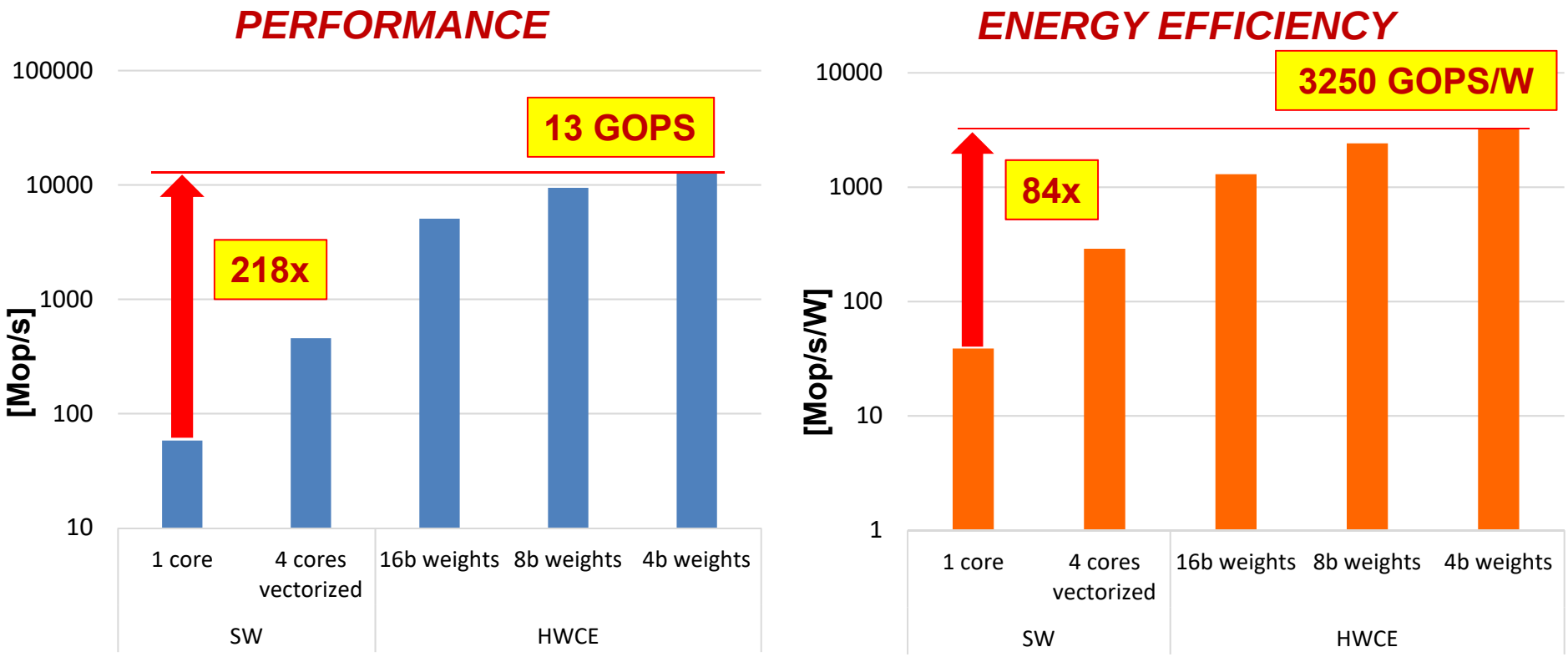
- **~1514** kGE for the cluster
- **232** kGE for the CNN HWCE



Crypto Engine → Secure Analytics

HWCE CNN Performance

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



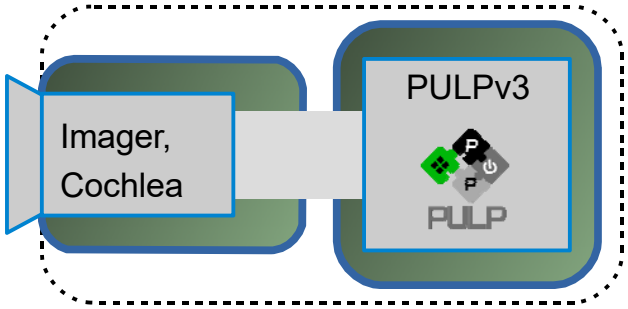
Scaled to ST FD-SOI 28nm @ Vdd=0.6V, f=115MHz

Outline

- Near Threshold Multiprocessing
- Non-Von Neumann Accelerators
- Aggressive Approximation
- **From Frame-based to Event-based Processing**
- Outlook and Conclusion

Back to System-Level

Smart Visual Sensor→ idle most of the time (nothing interesting to see)



Mixed-Signal Event-based Sensor

Digital Parallel Processor

- **Event-Driven Computation**, which occurs only when relevant events are detected by the sensor
- **Event-based sensor interface** to minimize IO energy (vs. Frame-based)
- **Mixed-signal event triggering** with an ULP imager, cochlea with internal processing AMS capability

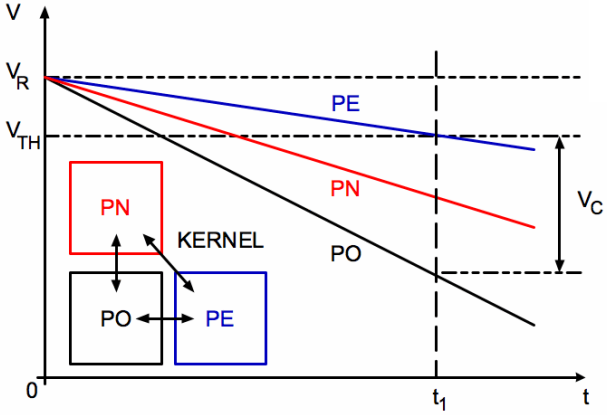
A Neuromorphic Approach for doing *nothing* VERY well

GrainCam Imager

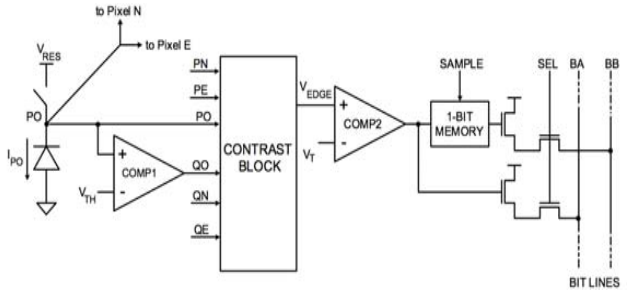
FONDAZIONE
BRUNO KESSLER

[Gottardi, JSSC09]

Pixel-level spatial-contrast extraction



$$V_C = V_{PE}(t_1) - V_{PO}(t_1) = (V_R - V_{TH}) \left(\frac{I_{PO} - I_{PE}}{I_{PE}} \right)$$



Analog internal image processing

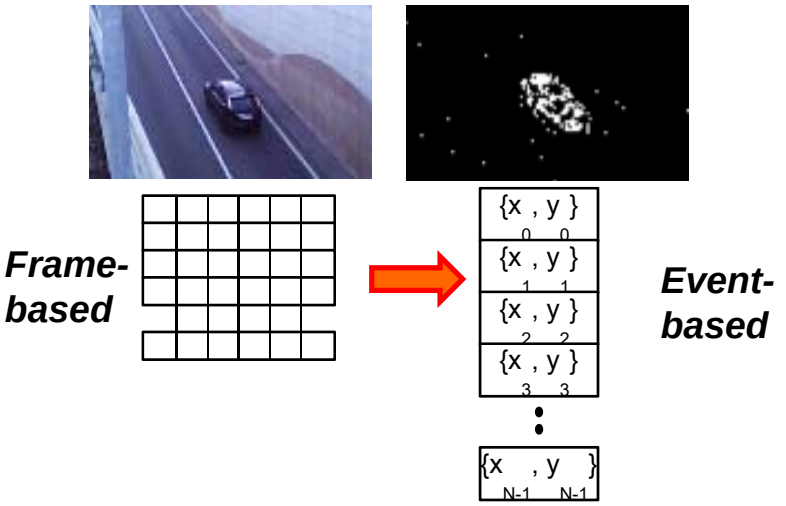
- Contrast Extraction
- Motion Extraction, differencing two successive frames
- Background Subtraction with the reference image stored in pixel memory

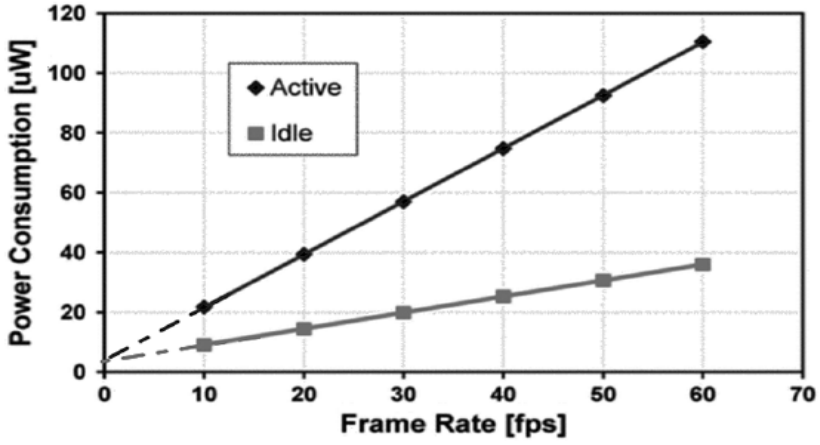
Graincam Readout

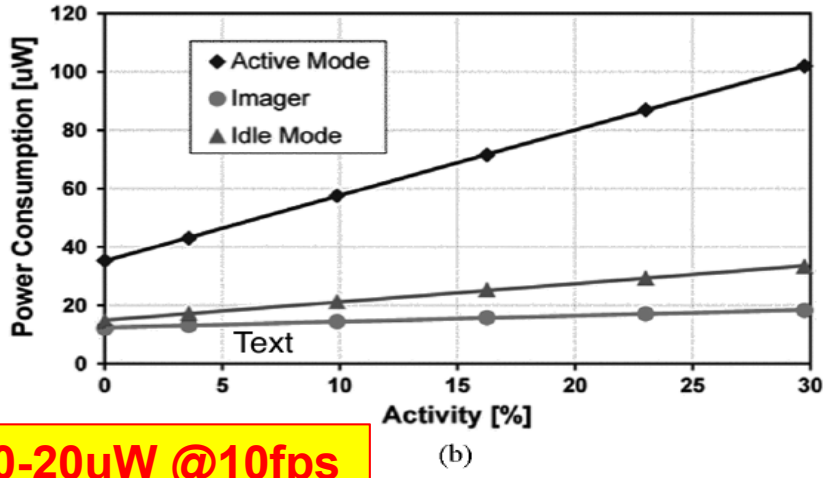
Readout modes:

- IDLE: readout the counter of asserted pixels
- ACTIVE: sending out the addresses of asserted pixels (address-coded representation), according raster scan order

Event-based sensing: output frame data bandwidth depends on the external context-activity

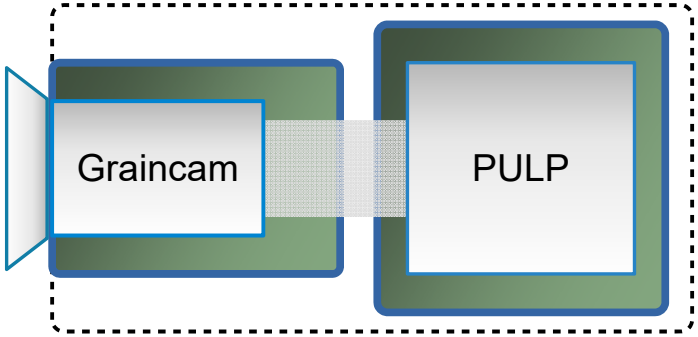


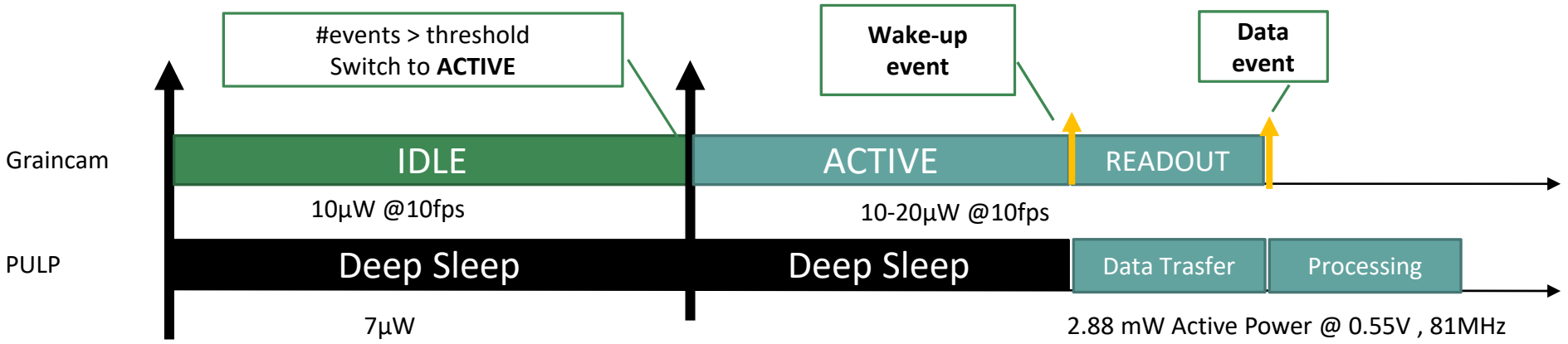




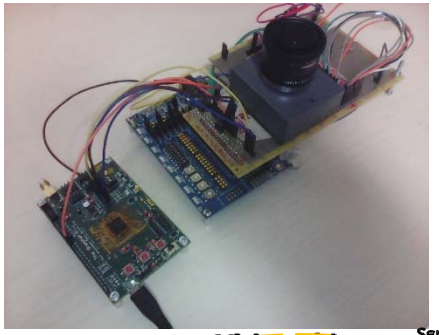
Ultra Low Power Consumption e.g. 10-20uW @10fps

Power Management

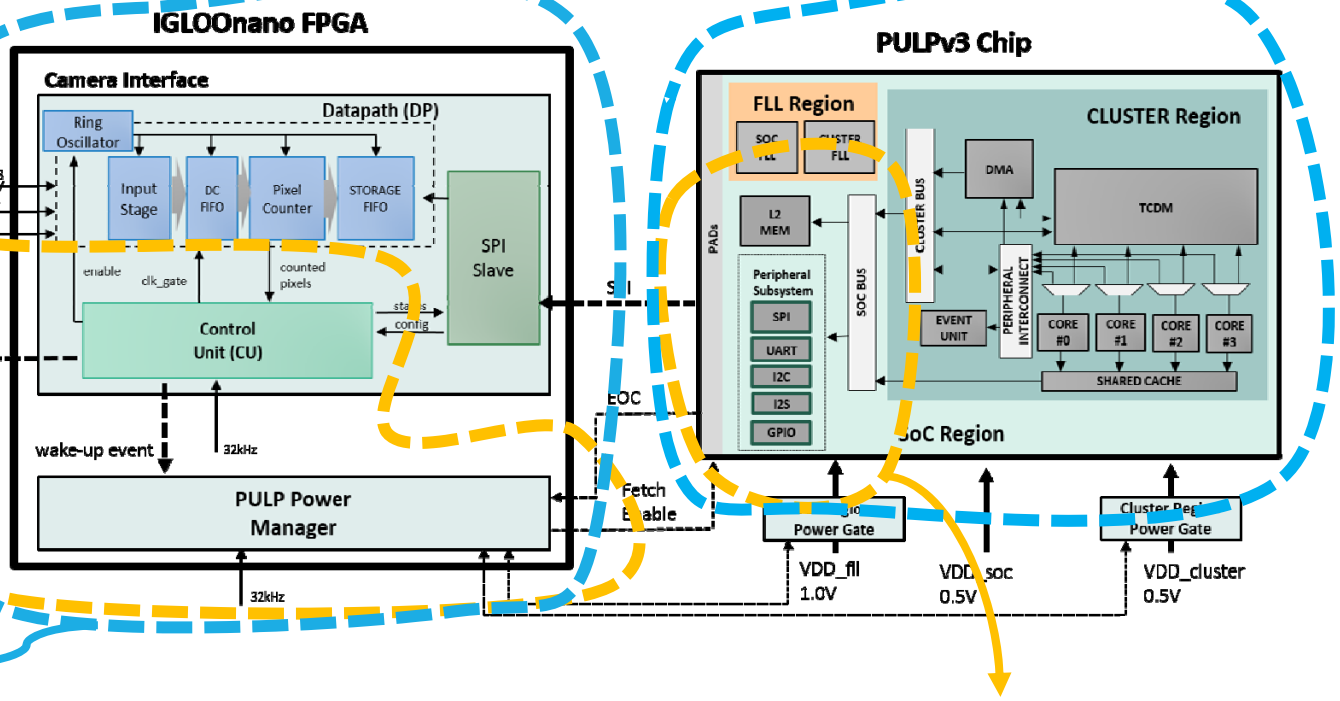




System Design



Active mode[30MHz@0.5V]:
4.6mW (FLL: 3.2mW)



Sensor IDLE: **10µW**
FGPA(@1.2V,32kHz): **68µW**
Sensor ACTIVE: **20µW**
FGPA(@1.2V,~25MHz): **3mW**

Idling at **193 µW** -> **SYSTEM IS SENSING!!**
Active mode **7.62mW** -> **when motion is detected**

SOC region idling for memory-state retention: **99µW**

M. **Rusci** et al. "A sub-mW IoT-endnode for always-on visual monitoring and smart triggering," in *IEEE Internet of Things Journal, 2017 (in print)*

Even-driven CNNs? Yes!

Binary Neural Networks reduce precision of weights and post-activation neurons to 1-bit precision while leading to a limited performance drop

Benefits:

Reduced storage costs by **32x** either for synaptic weights and temporary input/output data

Reduced computation complexity

weight filter

1	0	0
0	1	1
0	1	0

Convolution reduce to:

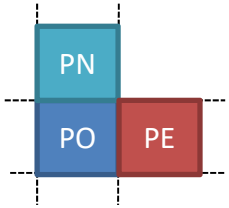
C = popcount (NOT (*weights* XOR *image*))

Binarization:

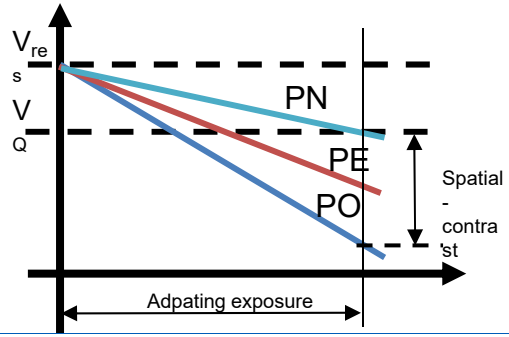
output = C >0 ? 1 else 0

Performing spatial filtering and binarization on the sensor die through mixed-signal sensing! → in-sensor first stage of the binary NN!!

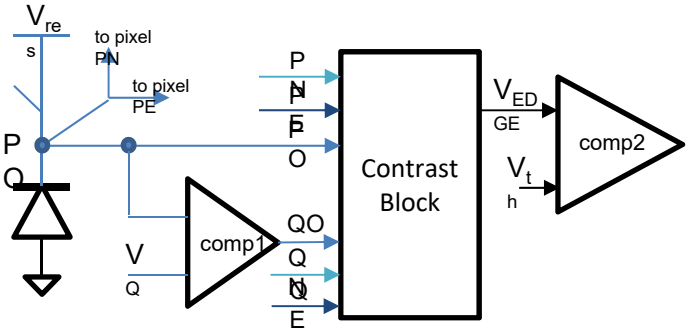
'Moving' pixel window



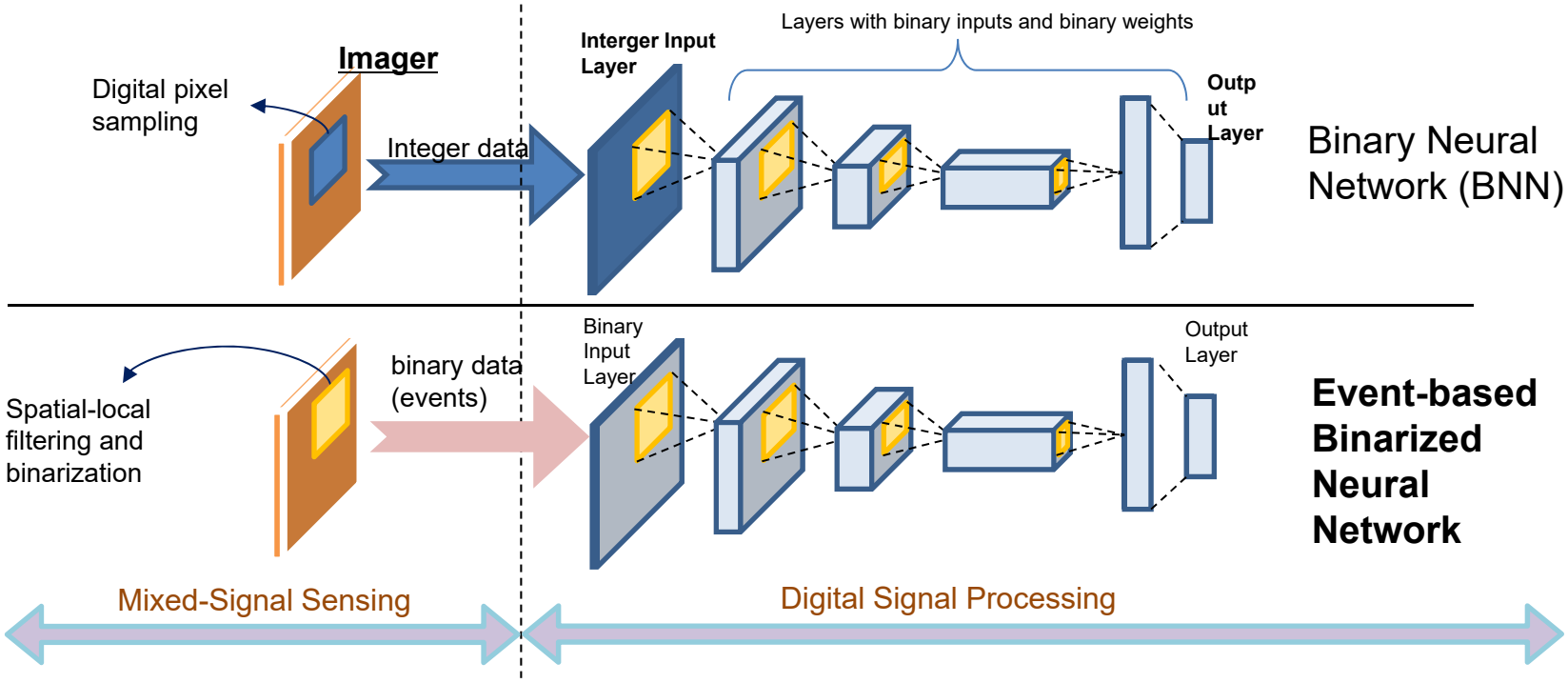
Gradient extraction



Per-pixel circitut for filtering and binarization



Event-Driven Binary Deep Network



Training challenge

Training Event-based Binarized Neural Network:

[ISSUE] Absence of huge amount of data for training

Modelling the ''graincam filter'' as a digital filter

Contrast Value $V_C \sim \frac{\max(|p_E - p_O|, |p_N - p_O|)}{\max(p_E, p_O, p_N)}$ Binary Output $V_O = sgn(V_C - V_{th})$

Original RGB image Synthetic image Graincam image

Evaluation on **CIFAR-10** (10 classes, 45k training, 5k valid, 10k testing)

Baseline with RGB input	92%
BNN with RGB input	86%
Baseline with binary input	72%
BNN with binary input	68%

Model VGG-like with 12 Convolutional laters and 3 Fully Connected Layers

18% performance drop because of input representation but still converges

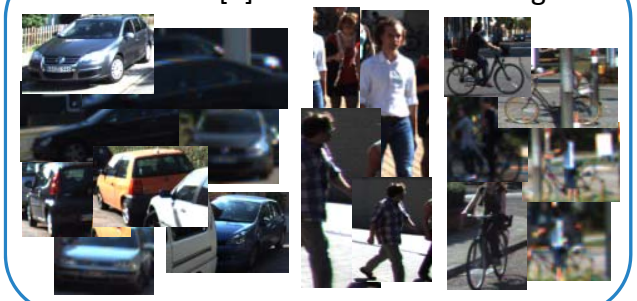
Results

TRAINING DATASETS

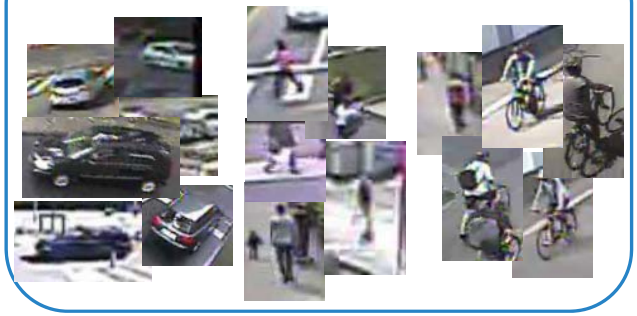
Training:
60k samples

Validation:
900 samples

KITTI dataset [1] – autonomous driving



MIO-TCD dataset [2] – surveillance system



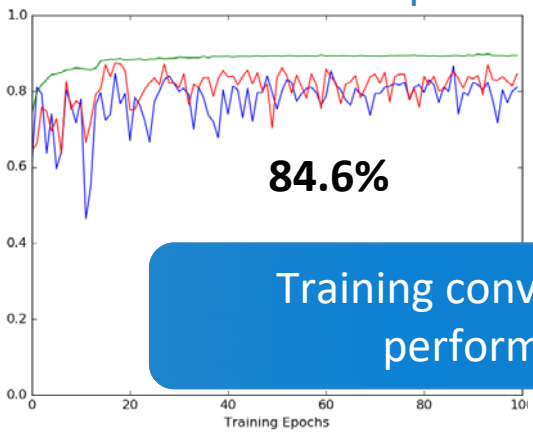
ACCURACY

Model: VGG-like architecture

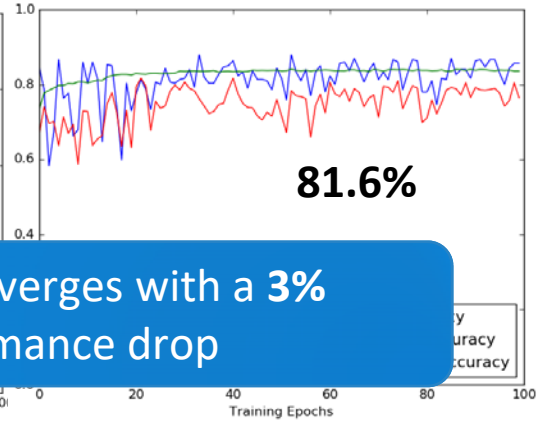
<20kB ''binary weight program''

CONV3x3(#c,16) + POOLING
CONV3x3(16,32) + POOLING
CONV3x3(32,48) + POOLING
CONV3x3(48,64) + POOLING
CONV3x3(64,96) + POOLING
FULLY-CONNECTED (384,64)
FULLY-CONNECTED (64,3)

BNN with RGB input



BNN with binary input



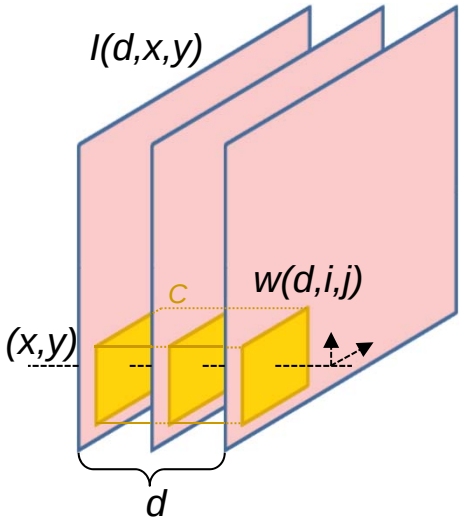
Training converges with a **3%** performance drop

[1] http://www.cvlibs.net/datasets/kitti/eval_object.php?obj_benchmark=2d
[2] http://podoce.dinf.usherbrooke.ca/

BNN implementation on PULP

Basic **Convolution Operation**:

$$\rho(x,y) = \sum_{(d,i,j)\in C} w(d,i,j) * I(d,i,j,x,y) \qquad I, w \in \{0,1\}$$



bdot_3d

$$\rho(x,y) = \boldsymbol{popcount}\{\ \boldsymbol{NOT}\ (w\ \boldsymbol{XOR}\ I(x,y))\ \}$$

Batch **Normalization** and **binarization**:

$$o_z(x,y) = \frac{\rho(x,y) + b - \mu}{\sigma}\gamma + \beta \geq 0 \qquad \rho(x,y) \in N$$

if $\gamma \geq 0$ **then** $o_z(x,y) = \rho(x,y) \leq \left\lfloor \mu - b - \frac{\beta * \sigma}{\gamma} \right\rfloor$ **else** $o_z(x,y) = \rho(x,y) \geq \left\lceil \mu - b - \frac{\beta * \sigma}{\gamma} \right\rceil$

just logic operation and integer comparison!

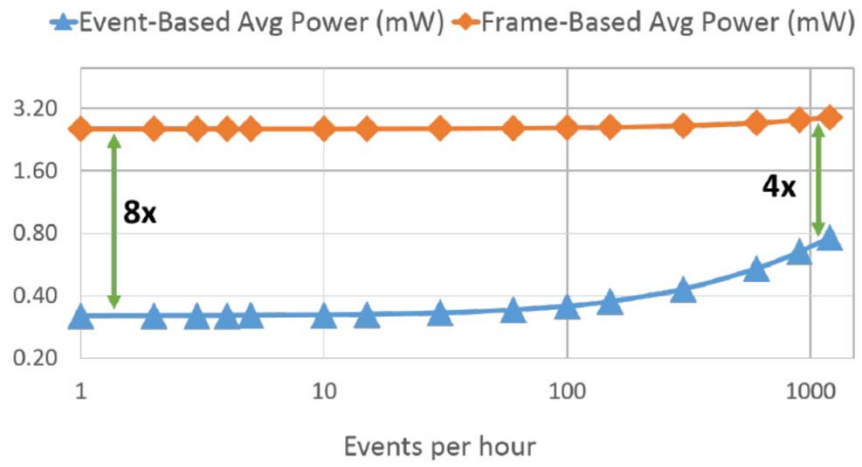
Major opportunity for HW acceleration!

Preliminary Results

Scenario	BNN with RGB input	Event-based BNN
Image Sensor Power Consumption	1.1mW @30fps	100μW @50fps
Image Size	632446 bits	8192 bits
Image Sensor Energy for frame capture	66.7 μJ	2 μJ
Transfer Time (4bit SPI @50MHz)	3.1 msec	0.04 msec
Transfer Energy (8.9mW @0.7V)	28 μJ	2 μJ
BNN Execution Time (168MHz)	81.3 msec	75.3 msec
BNN Energy consumption (8.9mW @0.7V)	725 μJ	671 μJ
Total System Energy for Classification	820 μJ	674 μJ

84.6% vs. 81.6% Accuracy

Statistics per frame	Frame-Based	Event-based
Idle (no motion)		
Sensor Power	1.1mW	20μW
Avg Sensor Data	19764 Bytes	-
Transfer Time	790μsec	-
Processing Time	3.02 msec	-
Avg Processor Power	1.45mW	0.3mW (sleep)
Detection		
Sensor Power	1.1mW	60μW
Avg Sensor Data	19764 Bytes	~536 Bytes
Transfer Time	790μsec	21.4μsec
Processing Time	3.47 msec	187.6μsec
Avg Processor Power	1.57mW	0.511mW
Classification		
Sensor Power	2mW	60μW
Avg Sensor Data	79056 Bytes	1024 Bytes
Transfer Time	3.16 msec	41μsec
Processing Time	81.3 msec	75.3 msec
Processor Energy	760 μJ	677 μJ

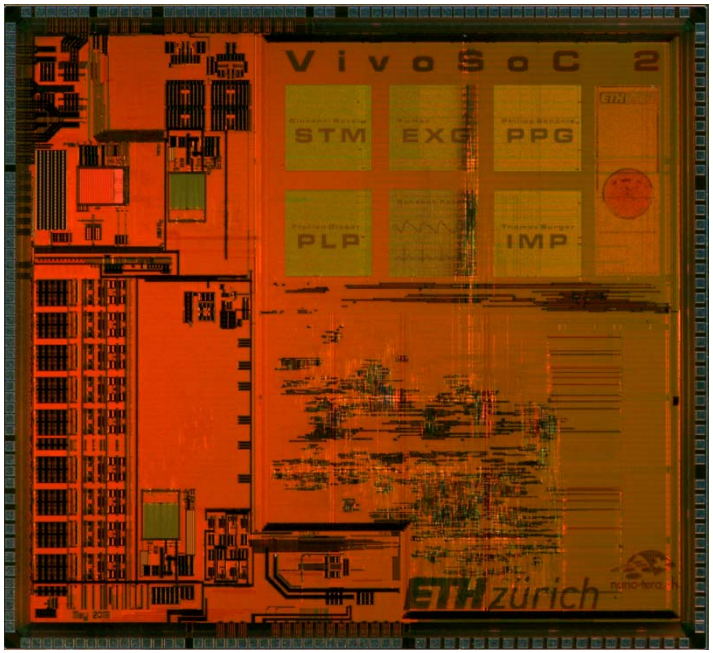


Outline

- Near Threshold Multiprocessing
- Non-Von Neumann Accelerators
- Aggressive Approximation
- From Frame-based to Event-based Processing
- **Outlook and Conclusion**

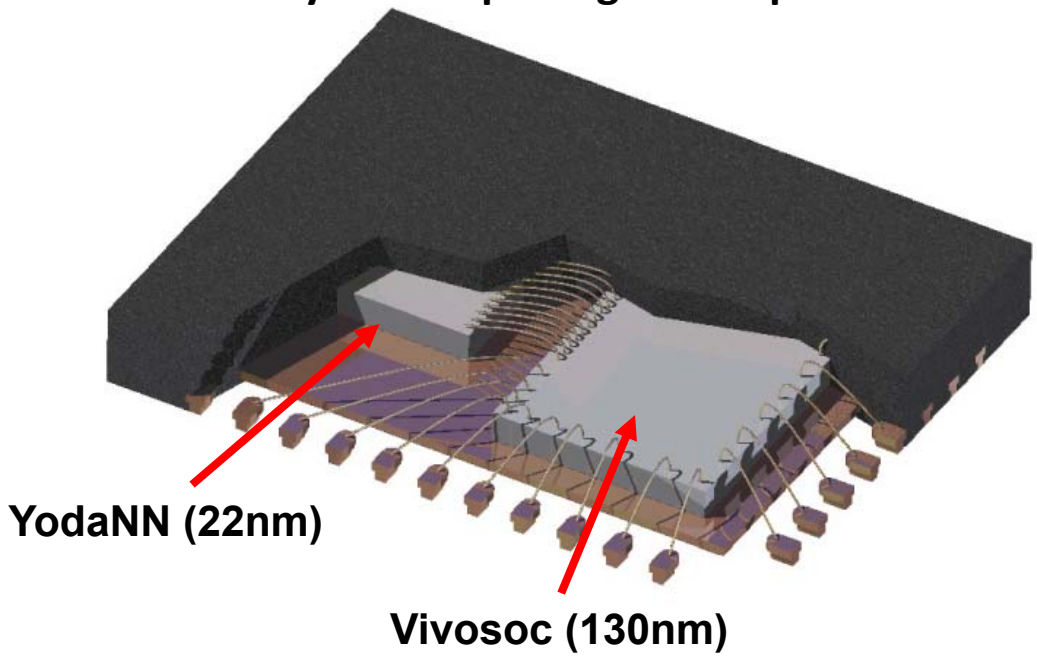
System Integration: SoC & SiP

VivoSoC2 – SoC AFE integration



Multiple Biomedical Interfaces:
8 ExG channels with lead-off detection, Photoplethysmography (PPG), Bio-impedance monitoring, temperature sensing, 6 channels nerve blocker and stimulator, standard digital interfaces

Full system in package concept



Low Cost Heterogeneous Integration:
CNN accelerator using highest density digital process for max GOPS/mW/mm^2. AFE A2D and GP processing in a good mixed-signal process, **Sensor and Memory** dies too…

Conclusions

- Near-sensor procesing for the IoT
  - CNN are top performers for «sensor understanding» in the cloud today
- Energy efficiency requirements: pJ/OP and below
  - Technology scaling alone is not doing the job for us
  - Ultra-low power architecture and circuits are needed
  - Most promising technologies: 3D integration, low-leakage, non-volatile silicon-compatible mem-computing devices, but LOW maturity and HIGH cost → large-scale CNN engines will be early adopters
- CNNs can be taken into the ULP (mW power envelope) space
  - Non-von-Neumann acceleration
  - Very robust to low precision computations (deterministic and statistical)
  - fJ/OP is in sight!
- Open Source HW & SW approach →innovation ecosystem

Morale:
Plenty of room at the bottom

Thanks!!!

www.pulp-platform.org
www-micrel.deis.unibo.it/pulp-project
iis-projects.ee.ethz.ch/index.php/PULP