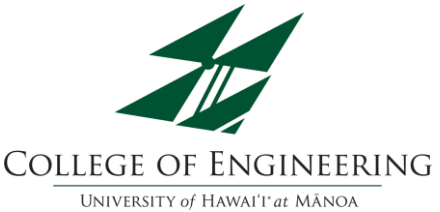
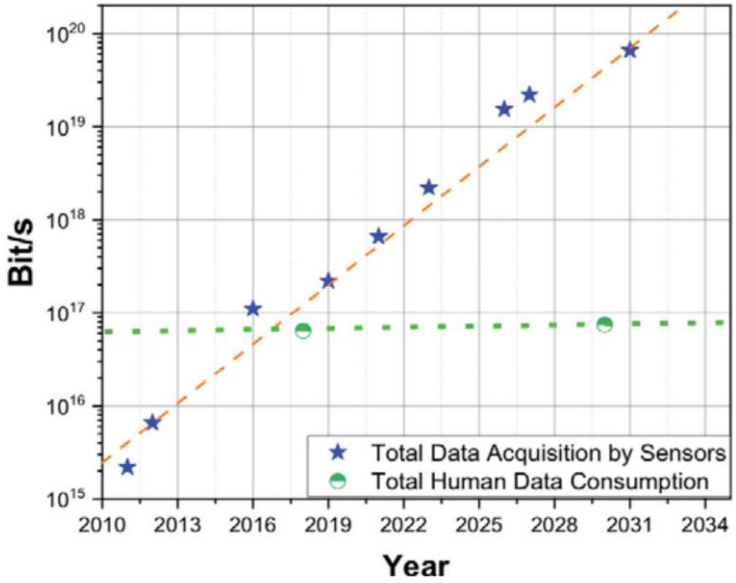
Mixed-Signal Interfaces and Compute Fabrics for tinyML Systems

Boris Murmann
bmurmann@hawaii.edu
May 19, 2025



(Sensor) Data is the New Oil!



- Today's sensors are generating orders of magnitude more data than can be consumed by humans

Figure from: SRC, Decadal Plan for Semiconductors, January 2021

Solution: Near-Sensor Data Distillation

- Computer vison example: Sensor device output is scene understanding

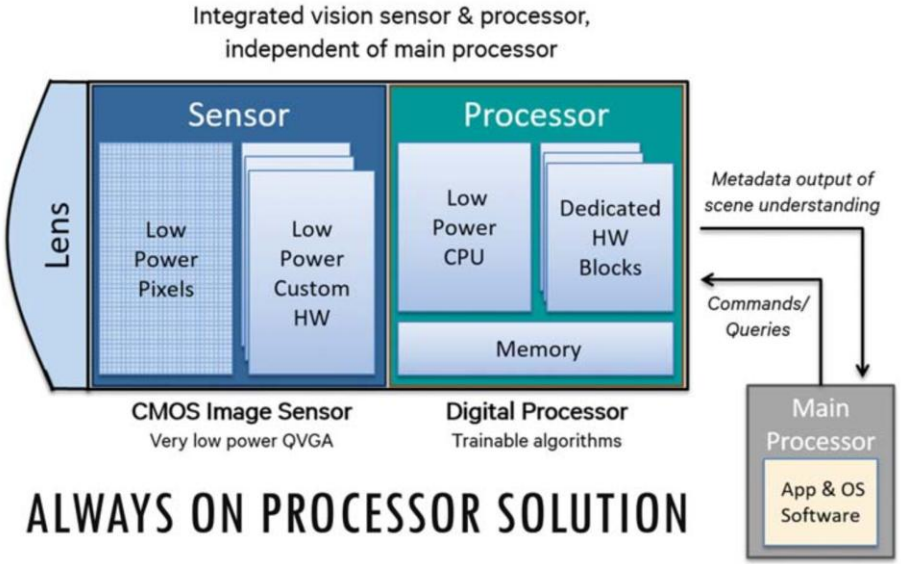
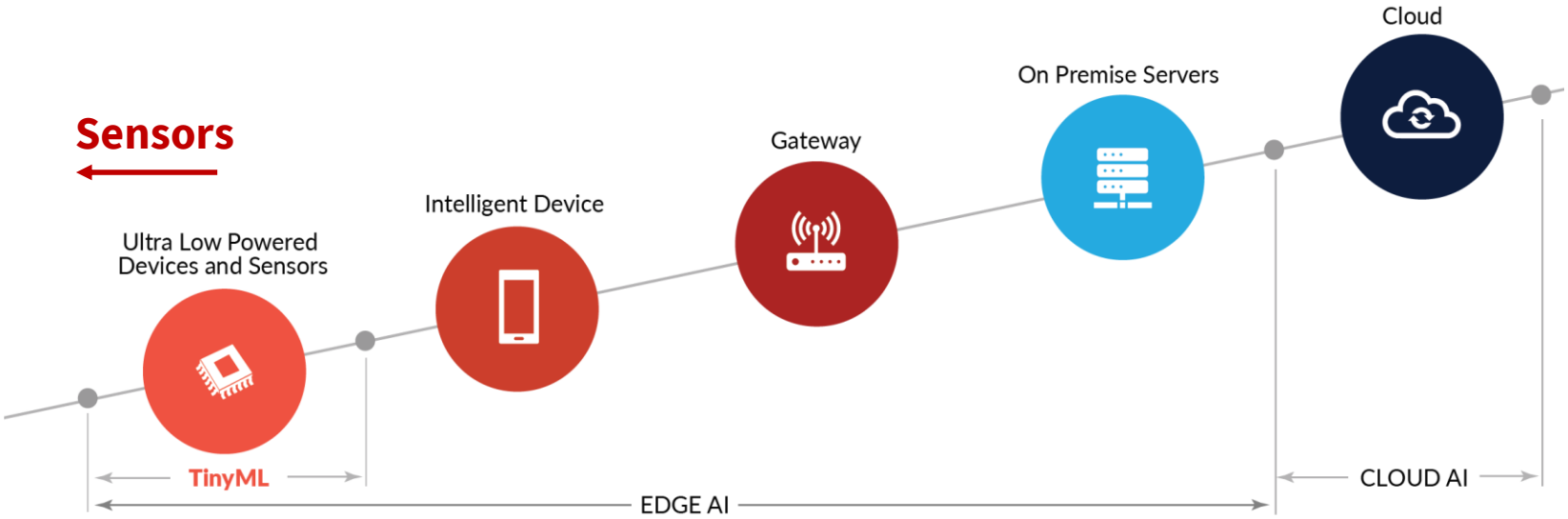


Figure from: SRC, Decadal Plan for Semiconductors, January 2021

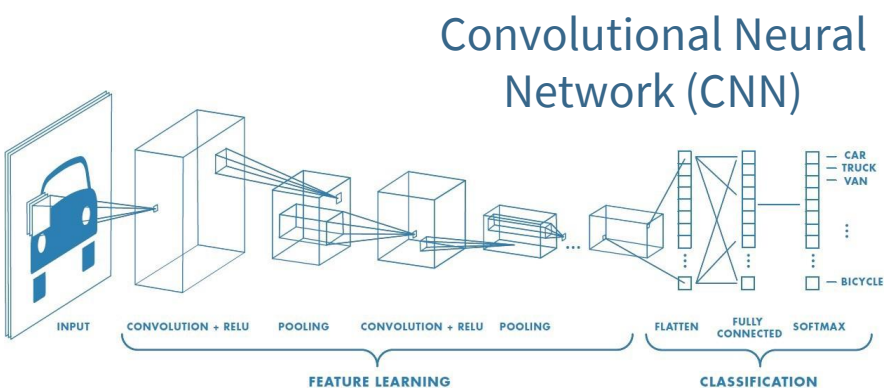
tinyML within the ML/AI Spectrum

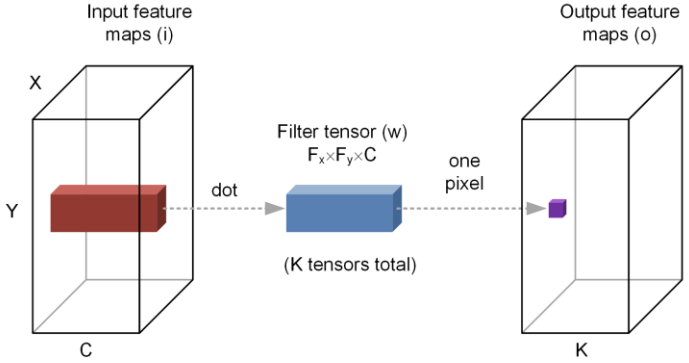


- In addition to data distillation: Low latency, improved privacy, autonomy
- Power ~1 mW, ML model size ~100+ kB

tinyML: The Next Big Opportunity in Tech, ABI Research Report, May 2021

What Do We Want?

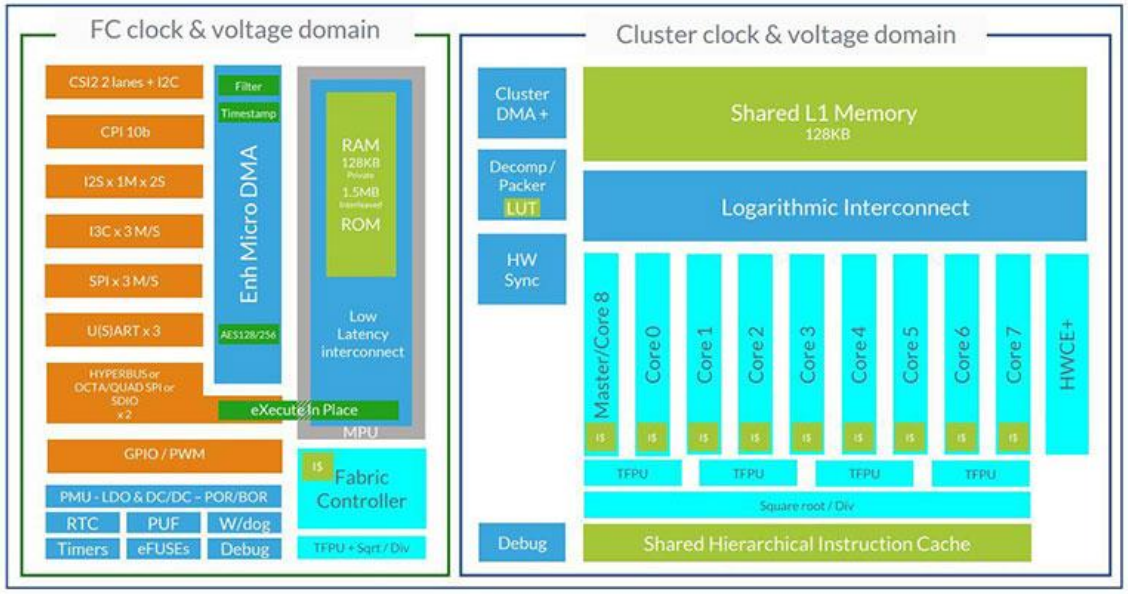




- ML inference is dominated by multiply & add operations (each counts as 1 OP)
- Need ~1 GOP for one neural network inference (can vary significantly)
- Want to perform ~100 inferences per second → 100 GOP/s
- Want to consume ~1 mW → 100 TOP/s/W → **10 fJ/OP**
- Even more aggressive goal → **1 fJ/OP**

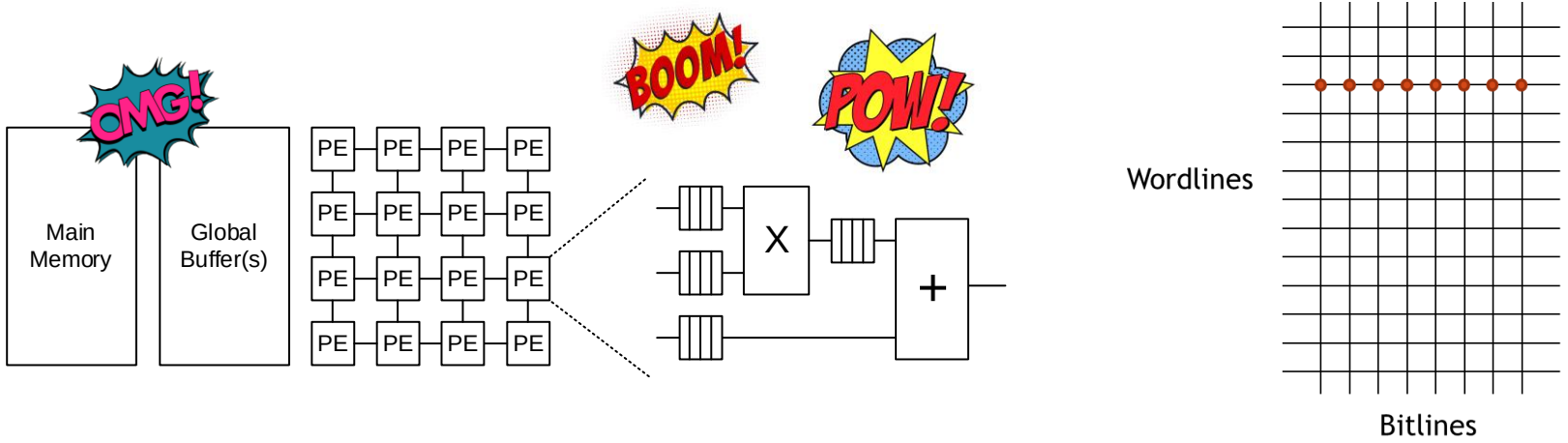
MCUs for tinyML

GreenWaves GAP9 (GF 22nm FDX)



- Blend of µC, DSP & NN accelerator
- Support of well-established toolchains
- MobileNetV1 inference (160x160input)
  - ~800 µJ/frame
  - ~1 GOP/frame
  - **~800 fJ/OP**
- How to lower energy?

Memory Access Bottleneck



- Energy bound considering processing element's register files alone
  - 28nm CMOS, 8-bit multiply & add (MAC), ~100-Byte RF

$$\frac{Energy}{OP} = \frac{E_{RF} + E_{MAC}}{2} = \frac{4 \times 50fJ + 100fJ}{2} = \mathbf{150\ fJ/OP}$$

Opportunities

$$Power = Rate \times \frac{Energy}{Inference} = Rate \times \frac{Operations}{Inference} \times \frac{Energy}{Operation}$$

Ultimate care-about

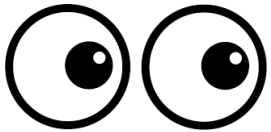
Reduce input data
Compress model
Exploit sparsity
…

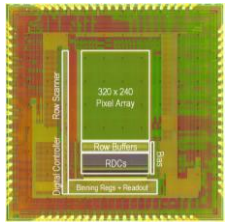
Better architecture
Better circuits
Better technology
…

- Armies of R&D engineers are working on these problems across multiple domains (HW, SW, Algorithms)

My Group's Work

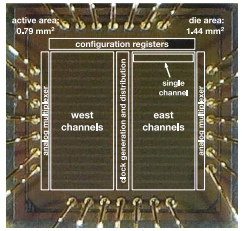
[Young, ISSCC 2019]



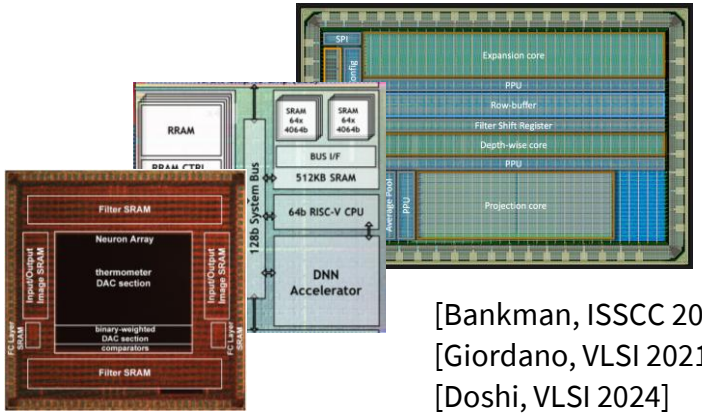


Video Preprocessing

[Villamizar, TCAS-I 2021]



Audio Preprocessing



[Bankman, ISSCC 2018]
[Giordano, VLSI 2021]
[Doshi, VLSI 2024]

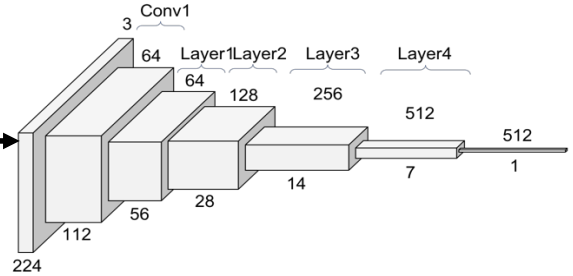
Custom Neural Network Accelerators

Computer Vision Pipeline

0.2-2 nJ/pixel
Imager output

1-2 nJ/pixel
Image Signal Processor

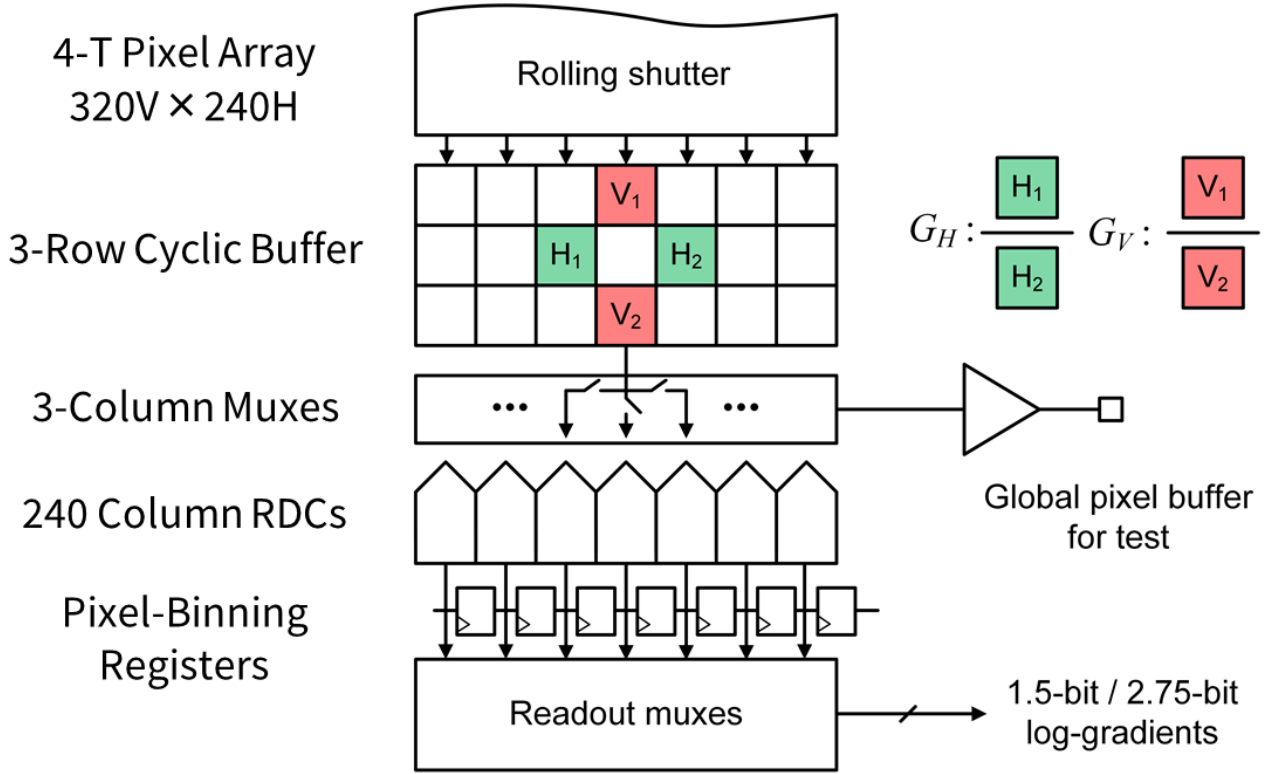
3-10 nJ/pixel
Convolutional Neural Network

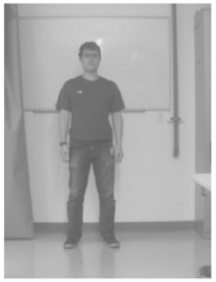


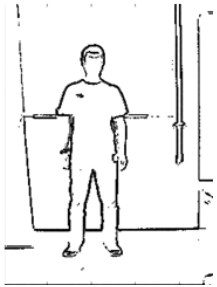
- Image data volume is already large
  - And CNN blows it up further
- For example, 224 x 224 x 3 → 112 x 112 x 64 (150,000 → 800,000)

Log Gradient Image Sensor

Chris Young

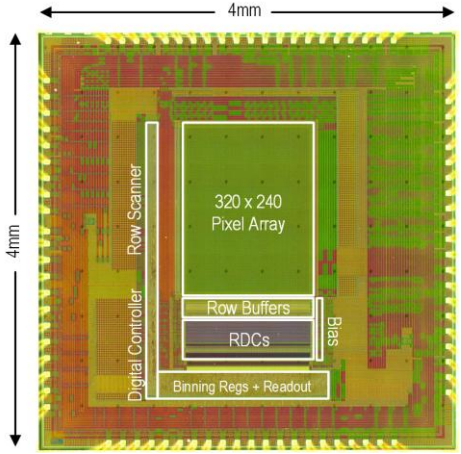






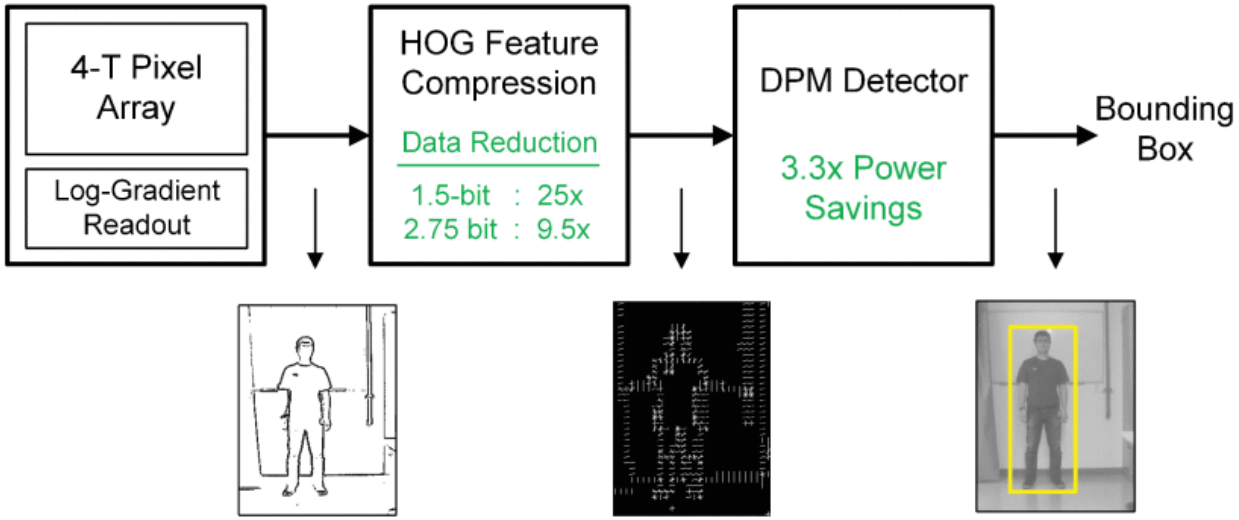
Young, ISSCC 2019

Prototype Chip with Processing Pipeline (Off-Chip)



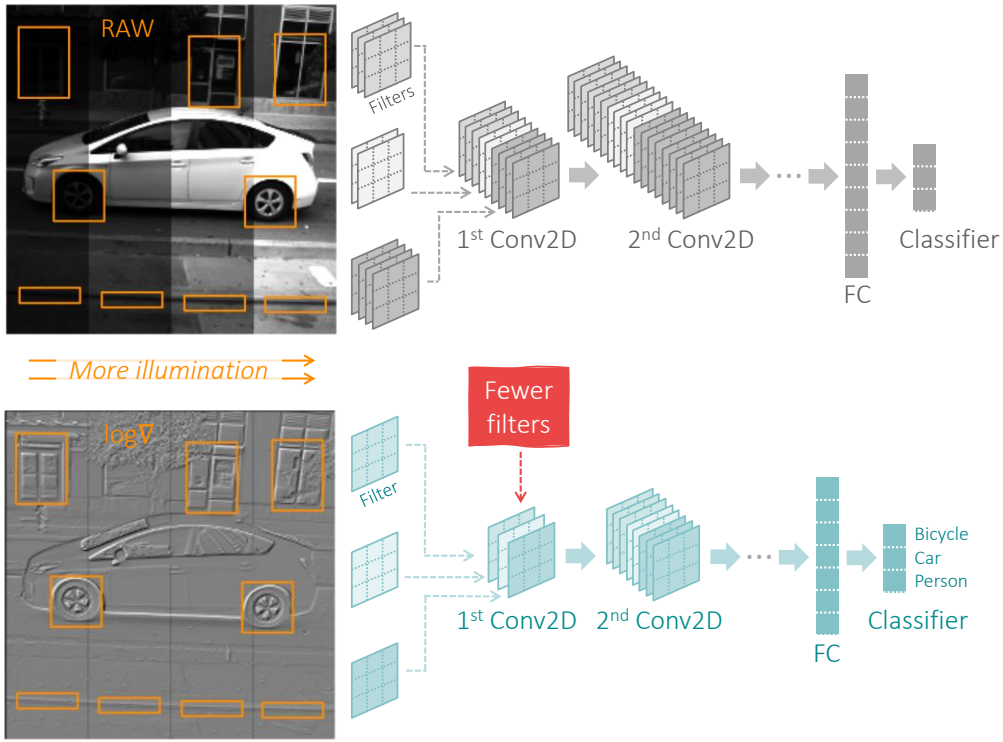
- 0.13 µm CIS 1P4M
- 5µm 4T pixels
- QVGA 320(V) x 240(H)
- 229 µW @ 30 FPS

Histogram of Oriented Gradients + Deformable Parts Model



Young, ISSCC 2019

Using Log-Gradients as CNN Inputs



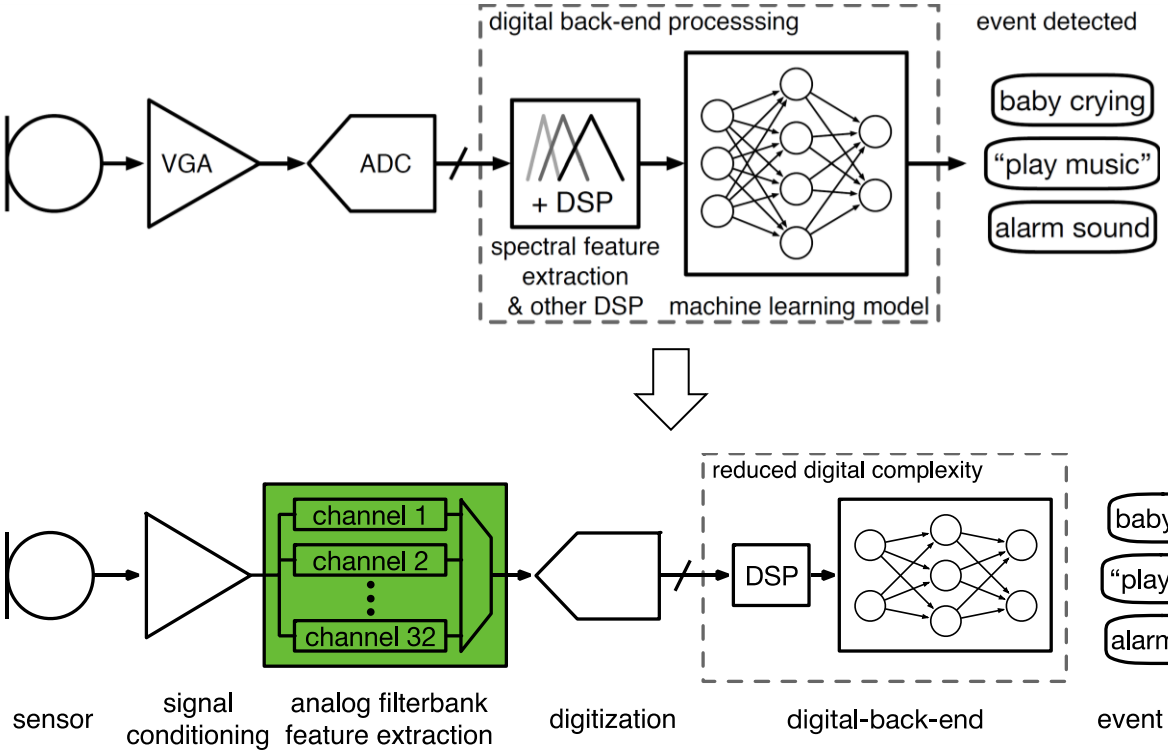
Qianyun Lu

- CNN needs fewer filters to discern relevant image features
- Can tolerate coarse quantization due to illumination invariance

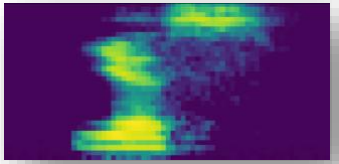
Q. Lu and B. Murmann, ACM Trans. Embed. Comput. Syst., May 2024

Sound Classification and Keyword Spotting

Dan Villamizar



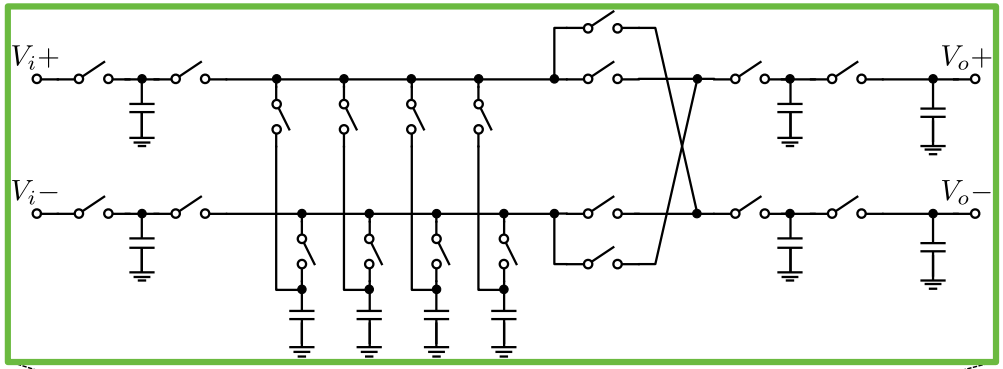
Spectrogram “Yes”

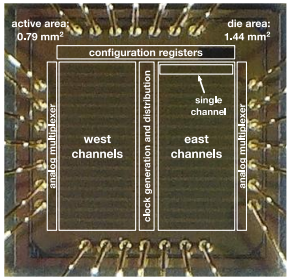


Information rate
~39 bits/sec

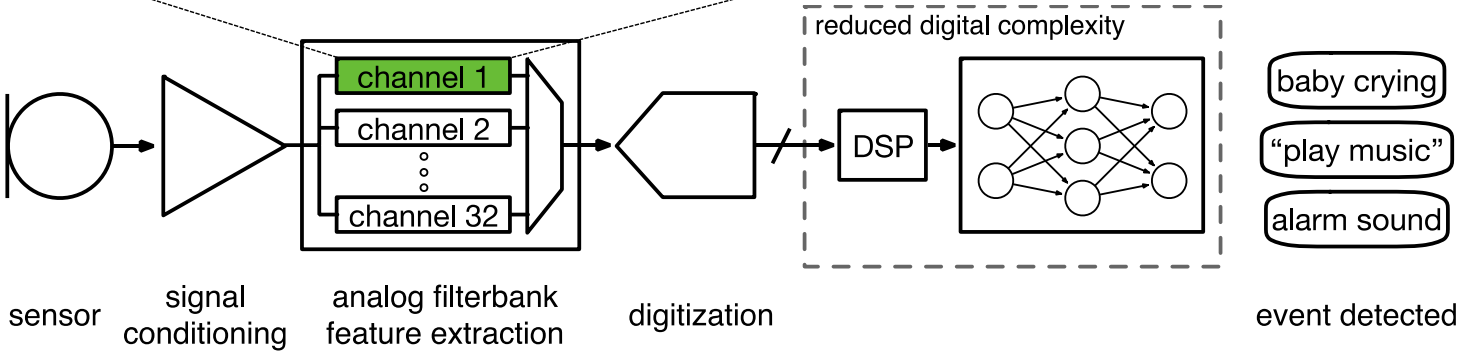
D. Villamizar, IEEE TCAS, 2021

Fully Passive Switched-Capacitor N-Path Filterbank



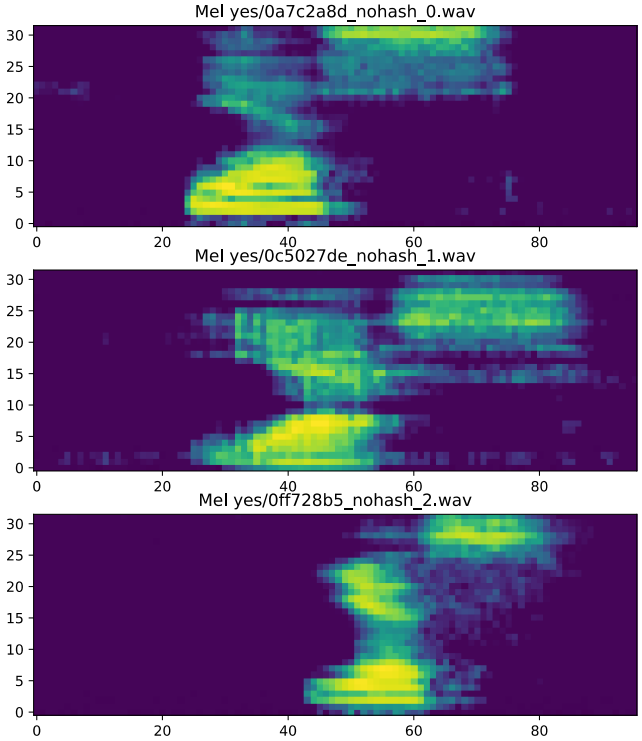


Ultra-low power (~800 nW)
(but ample nonidealities!)

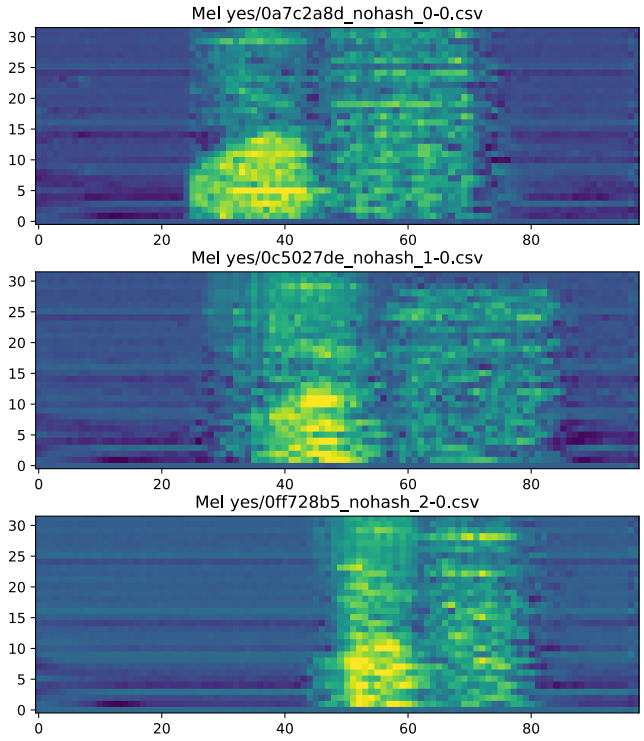


Voice Command “Yes”

Ideal

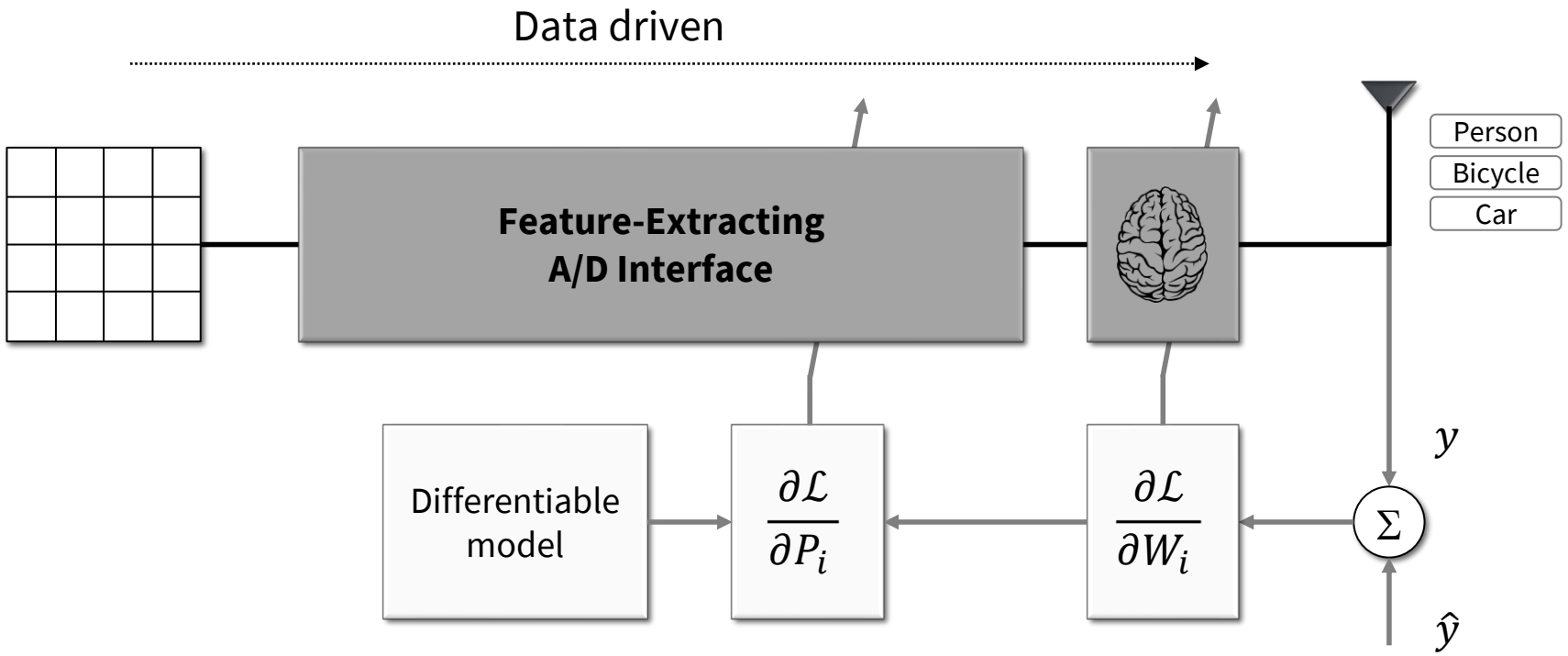


Our chip



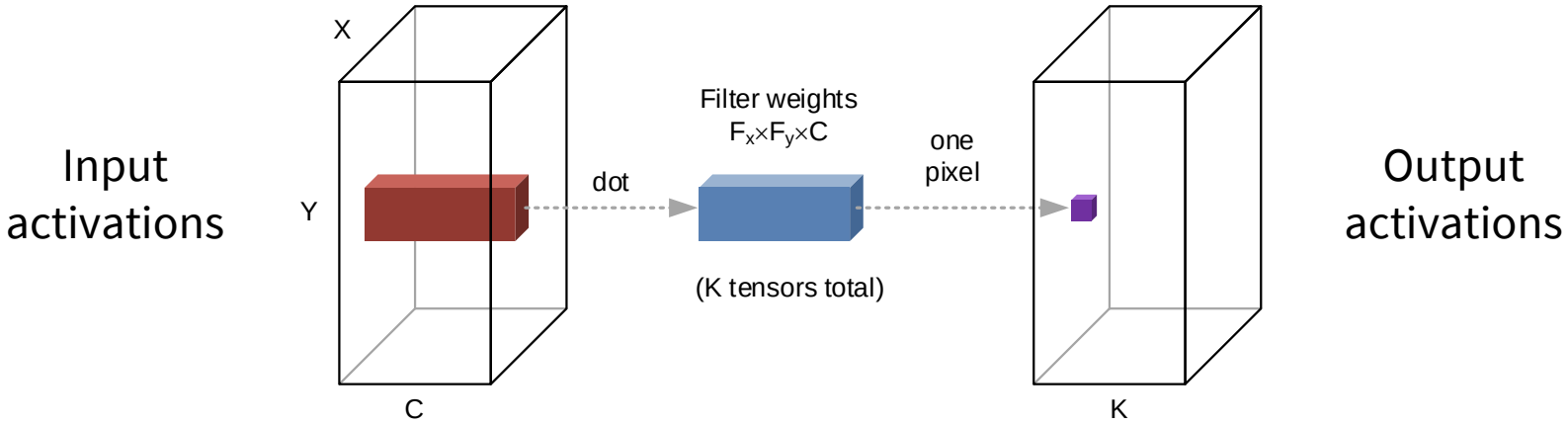
→ Retrain neural network to absorb nonidealities

The Next Frontier: End-to-End Training



Custom Neural Network Accelerators:
Should We also Embrace Analog Processing Here?

Elementary Convolution Layer



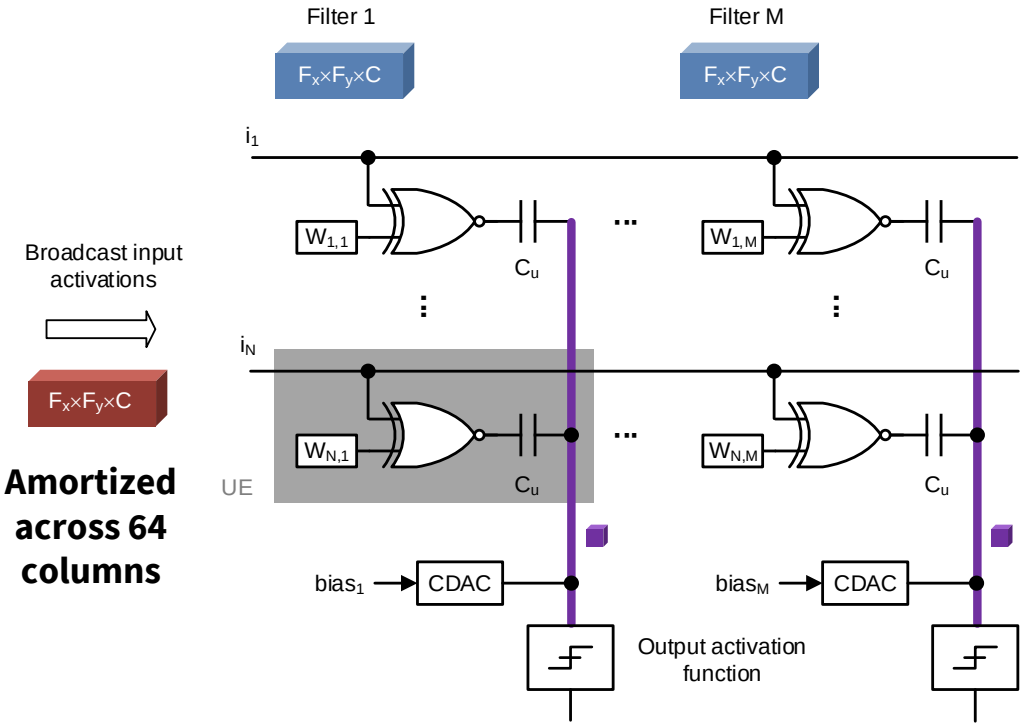
- Three-dimensional dot-product (multiply & add)
- Highly parallelizable computations (“embarrassingly parallel”)

Just a Big For-Loop

- Custom DNN accelerators leverage parallelism and data re-use
  - Loop unrolling
  - Optimum not tractable

```
for (k=0 to K-1); each output channel
   for (c=0 to C-1); each input channel
      for (x=0 to X-1); each input column
         for (y=0 to Y-1); each input row
            for (fx=0 to Fx-1); each filter column
               for (fy=0 to FY-1); each filter row
                  o[k, x, y] += w[k, c, fx, fy] × i[c, x+fx, y+fy]
```

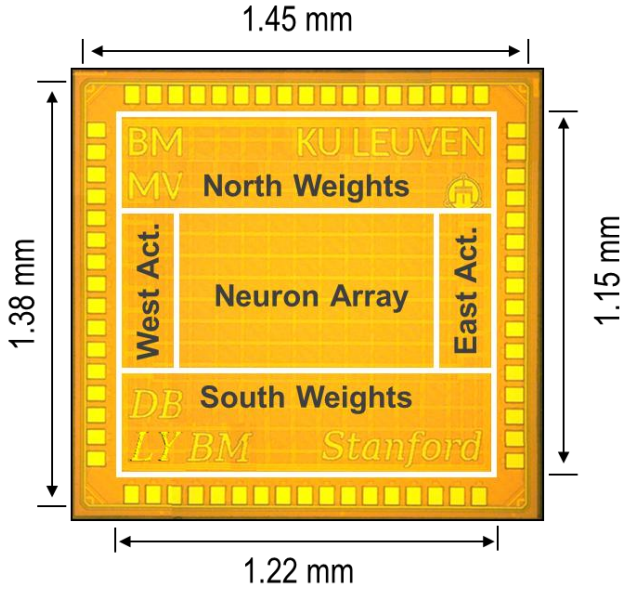

Mixed-Signal BinaryNet → Fully Unrolled (1024 x 64)



- Aggressive quantization
  - Binary weights and activations
- Analog accumulation
  - Bankman, ISSCC 2018
- Digital accumulation
  - Moons, CICC 2018

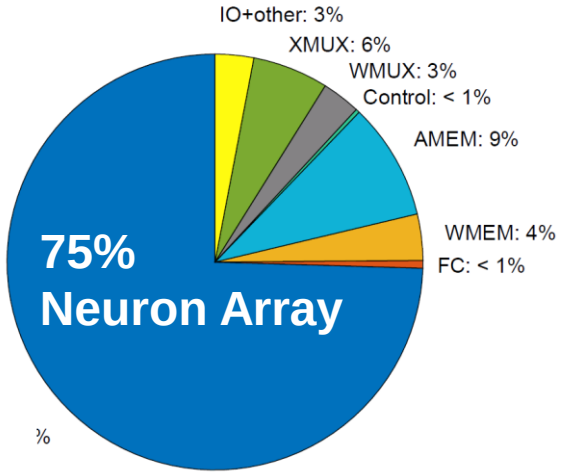
Fully Digital Implementation

Energy dominated by neuron array adder tree



[Moons, CICC 2018]
TSMC 28 nm, 328 KB SRAM

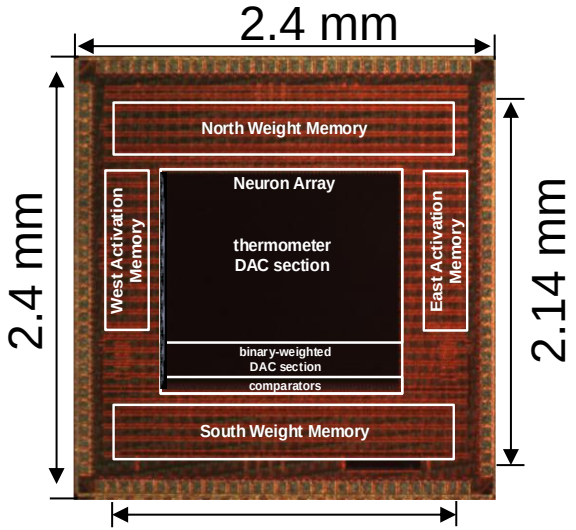
14.4 µJ/classification (CIFAR-10)



Bert Moons

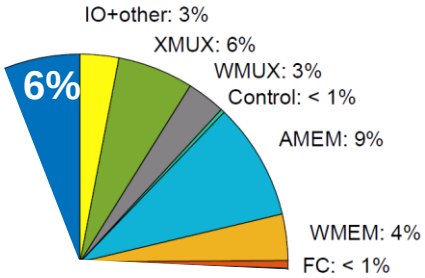
Mixed-Signal Implementation

Energy consumption balanced



[Bankman, ISSCC 2018]
TSMC 28 nm, 328 KB SRAM

3.8 μJ/classification (CIFAR-10)



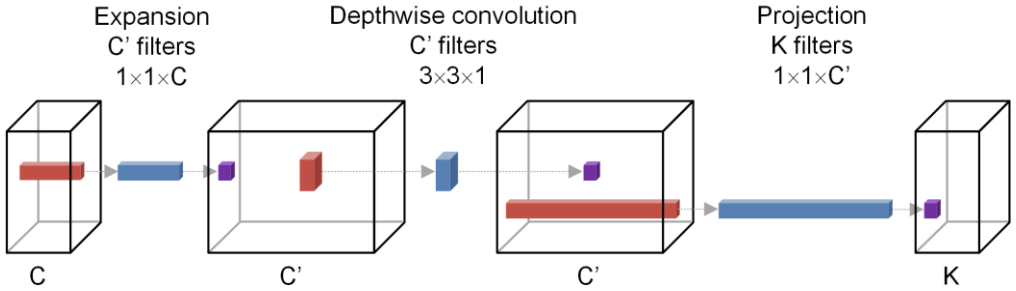
Neuron Energy / 12.9
System Energy / 3.8

Danny Bankman

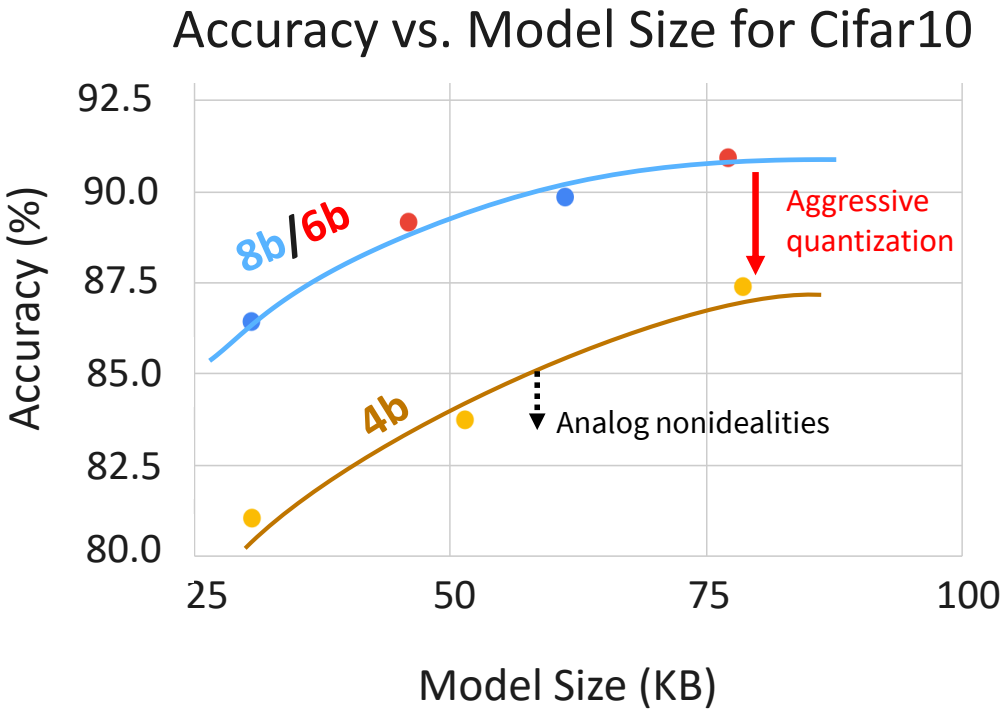
Lita Yang

Critical Review

- Analog CIM macros have great block-level specs, but tend to be one-trick ponies
  - Limited programmability
  - Efficient only for relatively large, fixed kernels
  - Energy benefits diminish for multi-bit compute
- Modern CNNs are less overprovisioned, tend to require multi-bit compute
  - Example: Bottleneck layer in MobileNetV2



Compute Precision Affects Model Size

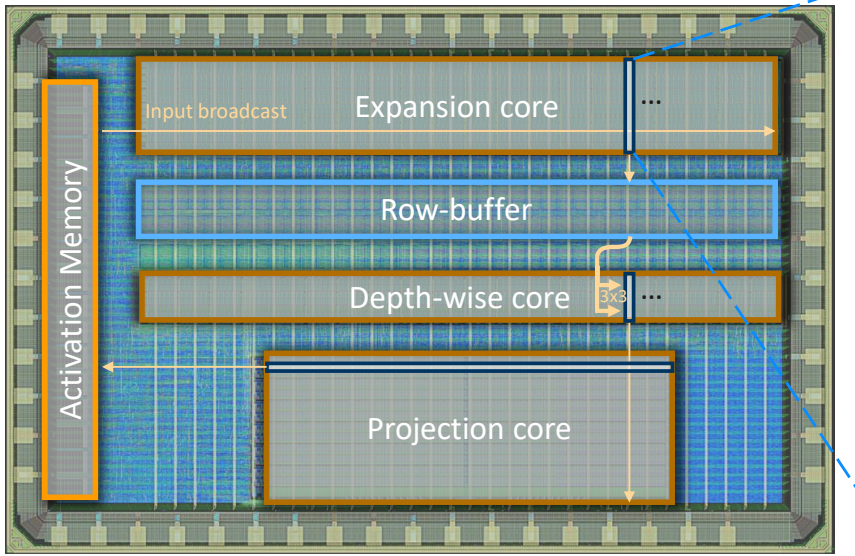


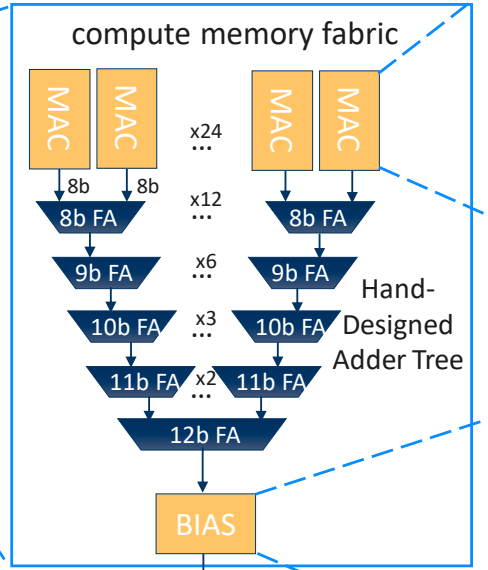
Massimo Giordano

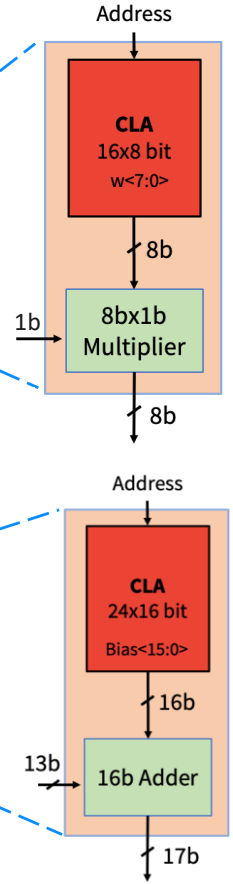
Rohan Doshi

- 8b digital arithmetic requires smaller model
  - At ISO-accuracy
- Our next-gen design uses fully digital arithmetic…

Medusa – Fully Digital Accelerator for tinyML







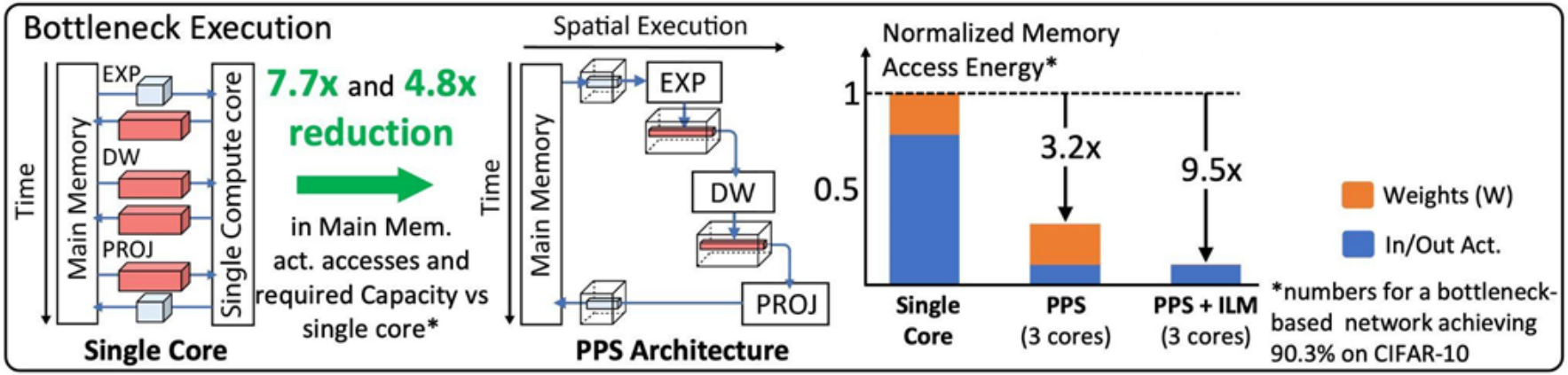
8bx8b multiplication computed over 8 cycles

bit-serial accumulator

[Doshi, VLSI 2024]
[Giordano, ASPLOS 2024]

Techniques for Reducing Memory Access Energy

- Pipelining reduces large memory access overhead of bottleneck layer activations
- Local memory (Inner Loop Memory) reduces weight access energy



Summary

- tinyML systems are gaining relevance due to sensor data deluge
- Custom chips for tinyML
  - Analog feature extraction → Data reduction
  - Custom computing for deep neural networks → Lower energy, improved density, reduced data movement
- Expect significant progress as application drivers emerge
  - Application targets and ML architectures are in constant flux