EAI 805 · EMBEDDED AI HARDWARE

Comparison of Embedded AI Hardware Platforms

Week 3 · Two-hour lecture · Platform emphasis: AMD / Xilinx PYNQ-Z2 (Zynq-7020 SoC)

Emmanuel Eronu · University of Abuja

Session at a glance

A two-hour block, one short break

00–10 min	Recap of Week 2; why platform comparison is genuinely hard
10–35 min	The five quantitative axes: TOPS/W, latency, area, $/TOPS, model size
35–55 min	Platform survey: CPU · GPU · TPU · MCU+NPU · FPGA
55–65 min	*Break*
65–90 min	Roofline analysis and the ridge point — PYNQ-Z2 worked example
90–105 min	Choosing a platform for a given workload + case discussion
105–120 min	Measuring it yourself on the PYNQ-Z2 + Lab 2 preview

Protect Section 4 (roofline) and Section 6 (PYNQ measurement) if time runs short.

Learning objectives

By the end of this lecture you can...

- Define the five comparison axes and what each does — and does not — tell you
- Explain why peak TOPS is a poor single-figure metric
- Contrast CPU, GPU, TPU/ASIC, MCU+NPU and FPGA for edge inference
- Compute arithmetic intensity and place a layer on a roofline
- Construct an approximate roofline for the PYNQ-Z2
- Apply a structured framework to select a platform under constraints
- Measure latency and throughput on the PYNQ-Z2 — and read it honestly

Why comparison is harder than it looks

Peak ≠ achieved

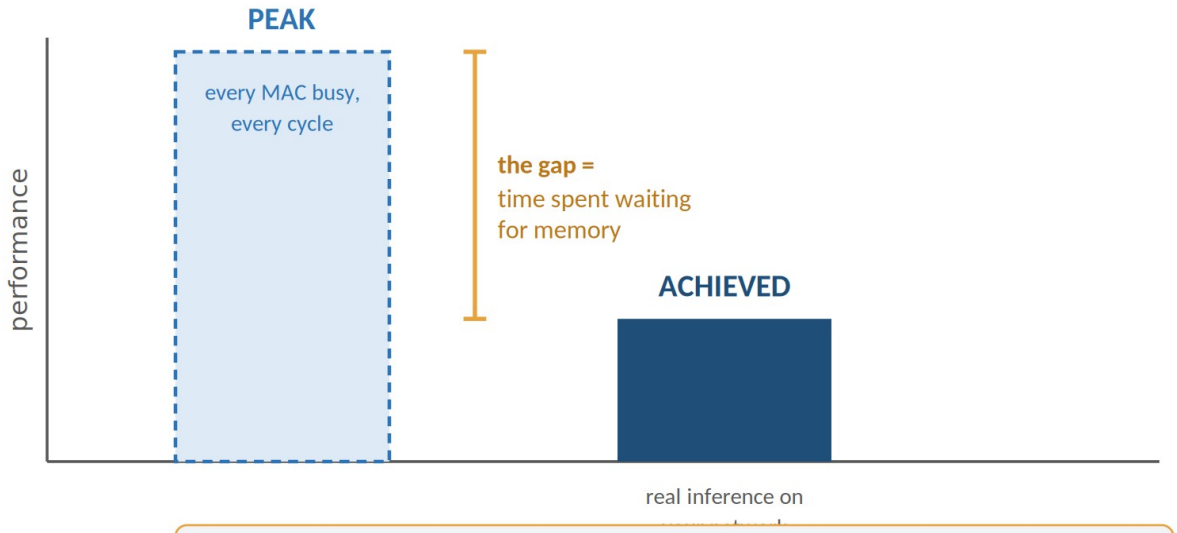
Vendors compete on one headline number — peak TOPS — which assumes every arithmetic unit is busy every cycle. Real inference spends much of its time waiting for weights and activations.

It is multi-dimensional

A platform that wins on throughput can lose badly on power, unit cost, or the largest model it can hold. State the constraints first, then find what satisfies all of them — don't maximise one number.

Peak TOPS is a ceiling, not a promise

What the datasheet quotes vs. what your model actually gets



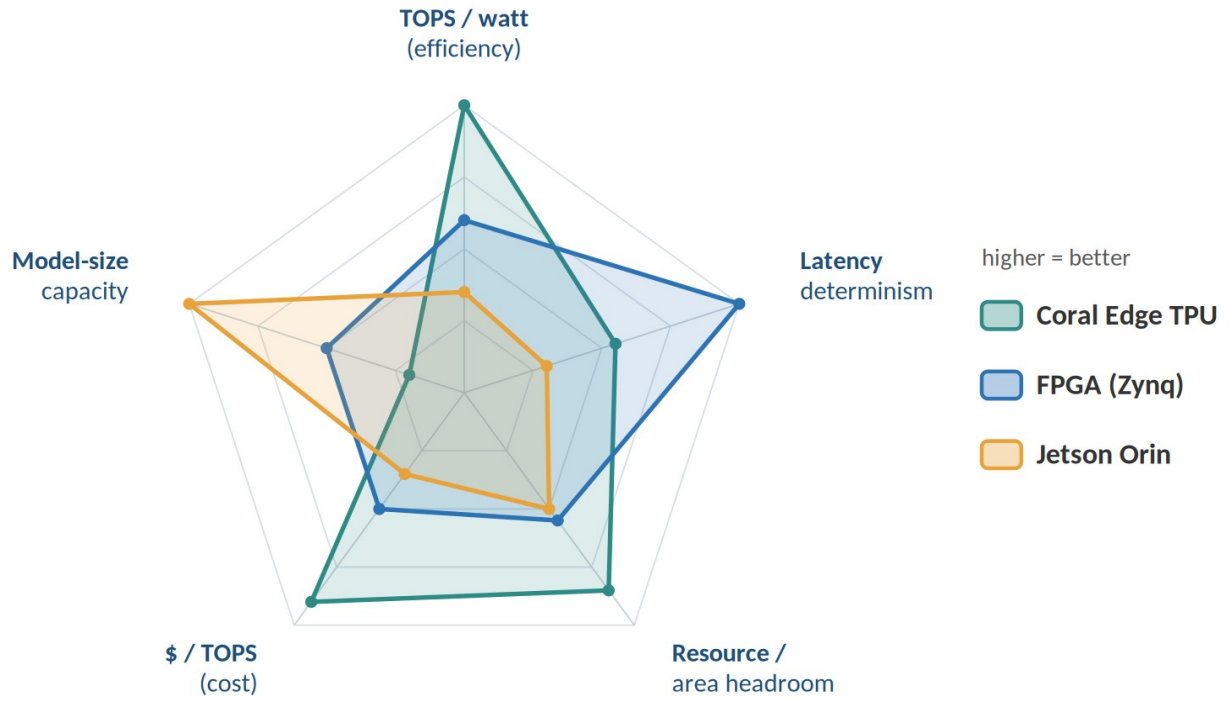
A 4-TOPS device can beat a 40-TOPS one — if it is better fed. Measure your own model.

No platform wins on every axis

- **TOPS / watt**
- **Latency**
- **Area / resources**
- **$ / TOPS**
- **Model-size limit**

Five axes at once — and no platform wins them all

Each family spikes on different axes; the engineer states the constraints, then finds what satisfies all of them



Read every number sceptically

TOPS / watt — inference per joule

The single most important axis for battery- or thermally-constrained deployment.

Always ask: at what precision? peak or measured? does the power figure include memory and host, or only the core? is sparsity assumed?

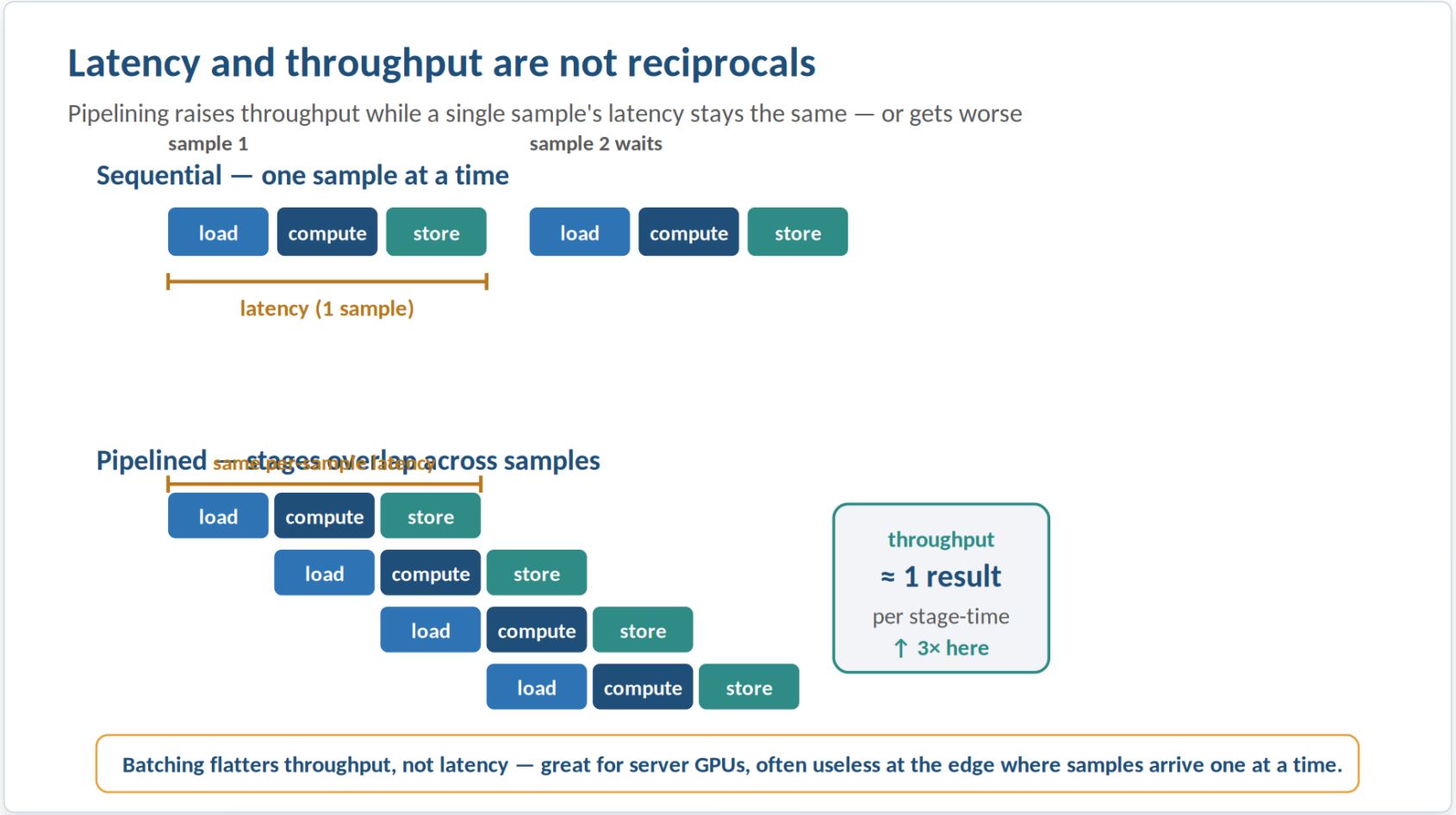
Two “equal-TOPS” devices can differ by an order of magnitude once these are pinned down.

Latency — time for one inference

The metric that matters for a control loop, a robot, or anything interactive. It is NOT the reciprocal of throughput — a batched design raises throughput while leaving single-sample latency untouched.

Report a distribution, not a mean. The p95 / p99 tail is what breaks real-time deadlines — and where OS-scheduled CPUs and GPUs look far worse than deterministic fabric.

Latency and throughput are not reciprocals



Area, cost, and the model-size filter

Area / resources

ASIC: mm² at a node. FPGA: LUTs, FFs, BRAM and above all DSP slices.

On an FPGA your “area” is the fraction of the budget you consume — and it directly caps parallelism.

$ / TOPS

Cost per unit of compute — the axis that decides whether a design ships at volume.

But the honest figure is total system cost: memory, power, thermals, board, certification, toolchain effort.

Model-size limit

Two ceilings: fast on-chip memory (SRAM / BRAM / cache) and off-chip (DDR / LPDDR).

Often decides the shortlist before any performance number — the FIRST filter, not the last.

PYNQ-Z2 fabric budget: **≈ 53k LUTs** · **140 BRAM blocks** · **220 DSP slices** — *why resource reports are part of every lab deliverable.*

§2.6 · Summary

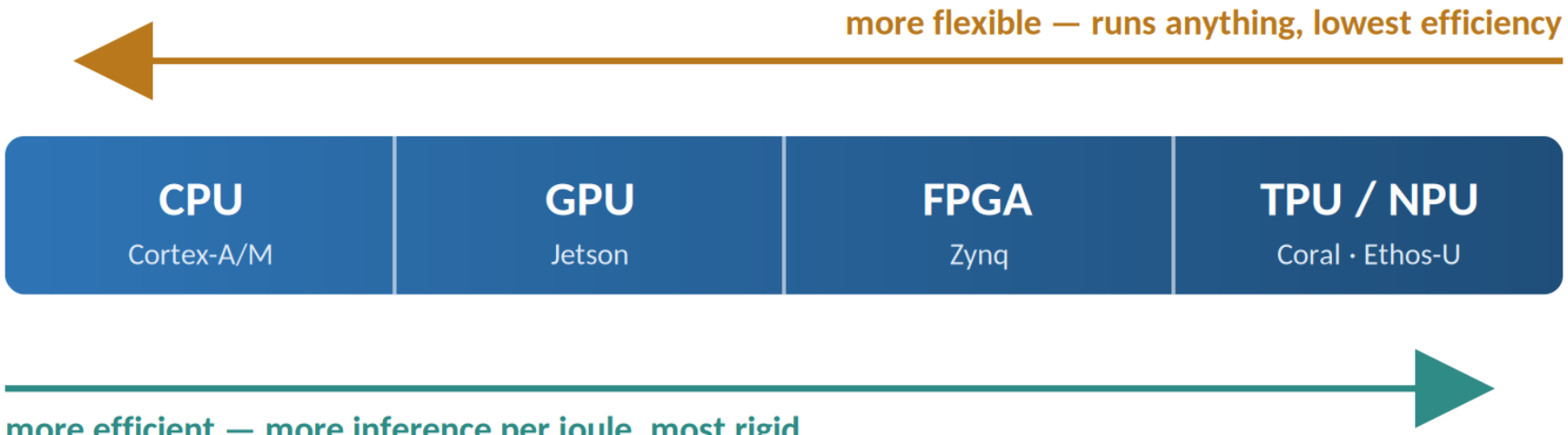
The five axes at a glance

Axis	What it measures	Main pitfall
TOPS / TOPS-per-watt	Peak arithmetic rate; efficiency per joule	Peak ≠ achieved; precision & power boundary often unstated
Latency	Time for one inference	Means hide the tail; batching flatters throughput, not latency
Area / resources	Die area (ASIC) or LUT / BRAM / DSP use (FPGA)	On FPGA it caps parallelism — must be reported with the design
$ / TOPS	Purchase cost per unit of compute	Ignores memory, board, thermals and engineering effort
Model-size limit	Largest model that fits and runs	On-chip vs off-chip capacity are different ceilings

Flexibility traded for efficiency

The flexibility–efficiency spectrum

Every family is a different answer to one question: how much flexibility do you trade for efficiency?



FPGA is the pivot: it buys arbitrary precision, dataflow, and deterministic latency — at the cost of design effort.

Five architectural families

CPU — Cortex-A / M

General-purpose baseline. NEON SIMD, runs anything. No MAC array — the number your accelerator must beat.

GPU — NVIDIA Jetson

SIMT + tensor cores, mature CUDA / TensorRT stack. Throughput and big models; non-deterministic under Linux.

TPU — Coral Edge TPU

Hardened systolic array, INT8 TFLite. ~4 TOPS at ~2 W. Superb in-niche, rigid the moment you leave it.

MCU + NPU — Ethos-U55

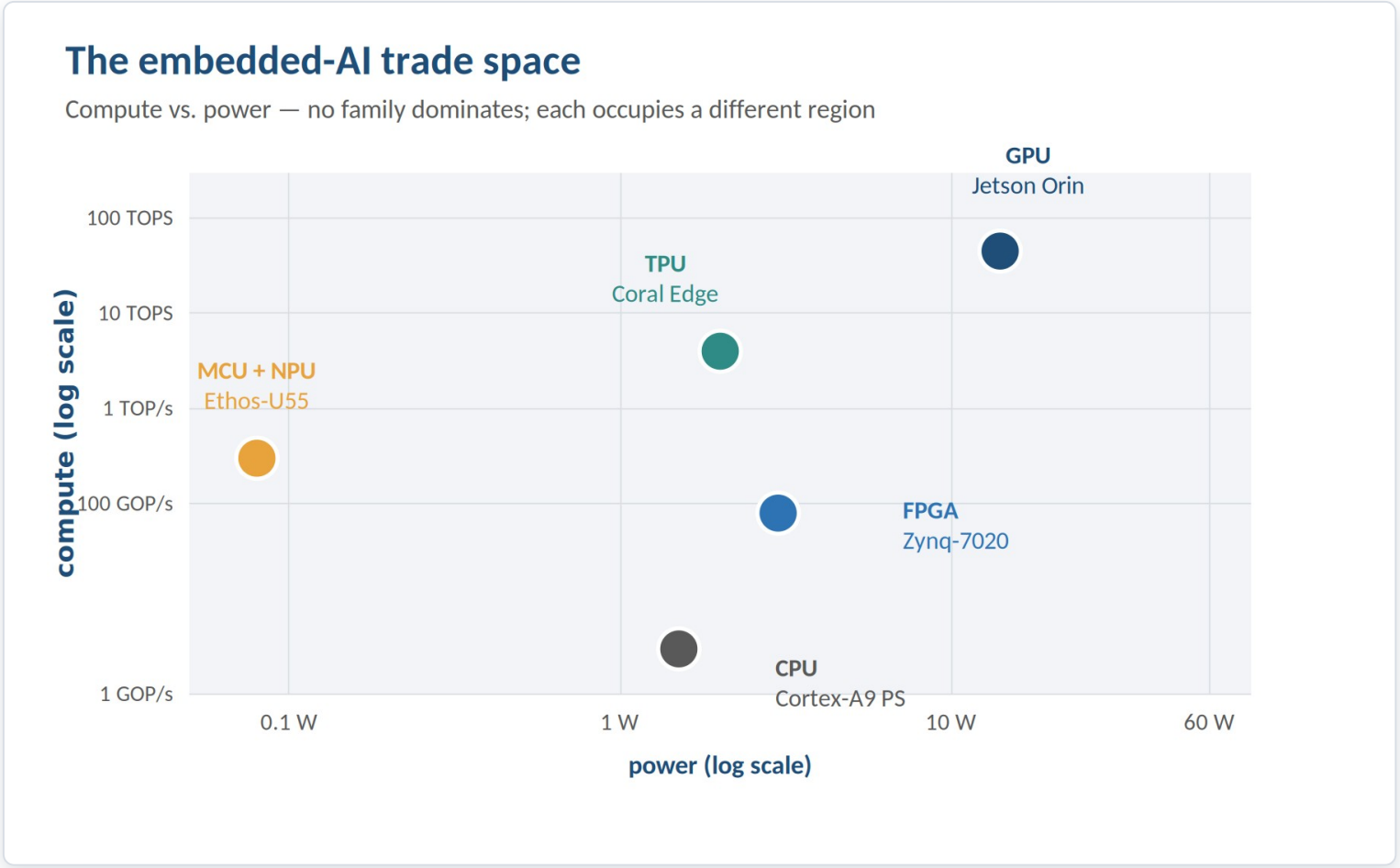
Configurable microNPU, ~0.1 mm^2, tens of KB SRAM. TinyML at milliwatts; incapable of anything large.

FPGA — Zynq-7000 / US+

Reconfigurable fabric: arbitrary precision, dataflow, deterministic latency. Cost is development effort.

The toolchain is the barrier the rest of this course exists to lower.

Where the shortlist comes from



§3.6 · Comparison

Indicative figures — orientation only

Platform	Typical compute	Power	Precision	Character
CPU (Cortex-A9, PYNQ-Z2 PS)	~GOP/s class	~1–2 W	FP32 / INT8	Baseline; total flexibility, lowest efficiency
GPU (Jetson Orin Nano)	20–40 TOPS (up to ~67)	5–25 W	INT8 / FP16	Throughput & big models; mature stack
TPU (Coral Edge TPU)	4 TOPS	~2 W	INT8 only	~2 TOPS/W; superb in-niche, rigid outside
MCU + NPU (Ethos-U55)	64–512 GOP/s	mW class	INT8 / INT16	TinyML; ~0.1 mm^2, tens of KB SRAM
FPGA (Zynq-7020)	Design-dependent	~2–5 W	Arbitrary (1–8 bit)	Custom dataflow, deterministic latency

Always re-check current vendor data — and, more importantly, measure your own model.

§4 · ROOFLINE ANALYSIS

What is actually limiting me?

$$\text{attainable OPS} = \min\left(\text{peak compute},\ \text{arithmetic intensity} \times \text{peak memory bandwidth}\right)$$

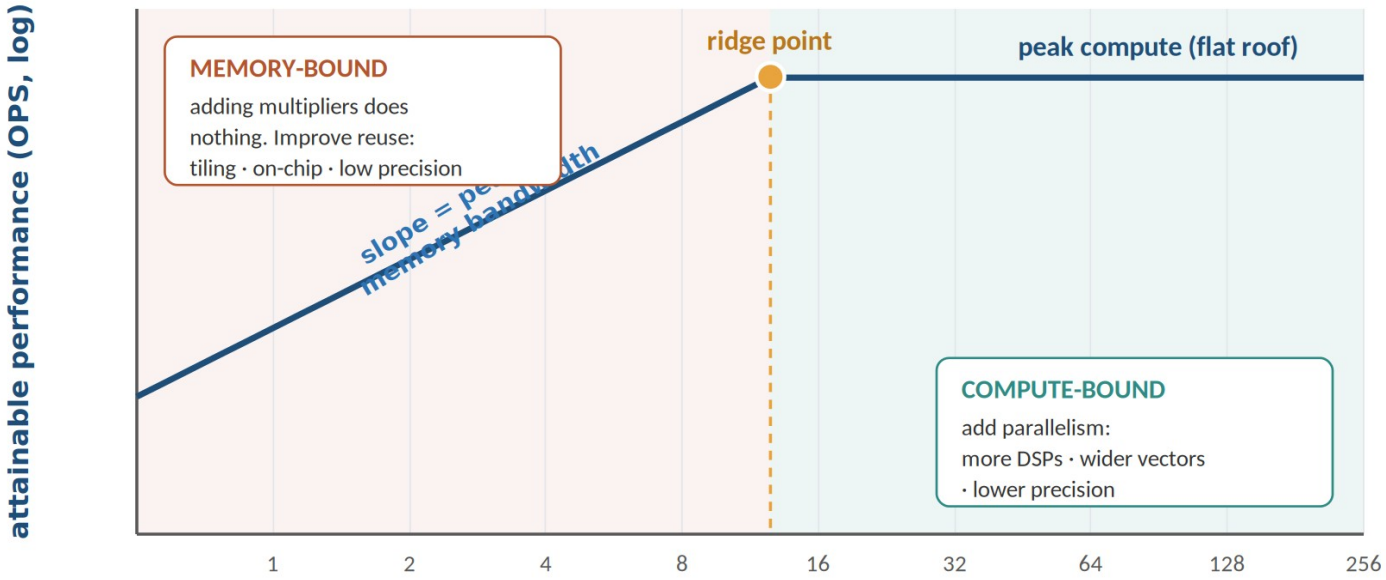
Plotted on log–log axes: a sloped memory-bound region on the left, a flat compute-bound plateau on the right. Where they meet is the ridge point.

Williams, Waterman & Patterson — Communications of the ACM, 2009

Two roofs, one ridge point

The roofline model: what is actually limiting me?

Attainable performance = min(peak compute , arithmetic intensity × peak memory bandwidth)



Which side of the ridge you land on tells you which optimisation will help — before you write any HLS.

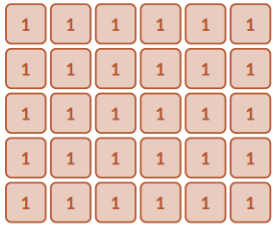
Intensity follows weight reuse

Why intensity depends on weight reuse

Intensity is a property of the layer and its dataflow — not of the hardware

Fully-connected

each weight loaded, used once



weight matrix W — every cell touched once

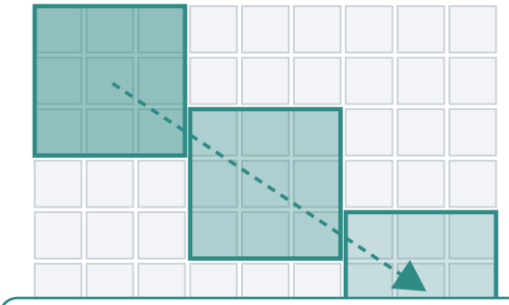
≈ 2 ops / byte → very low

almost always MEMORY-BOUND

MLPs & LSTMs stress bandwidth, not MACs

Convolution

one small kernel, reused everywhere



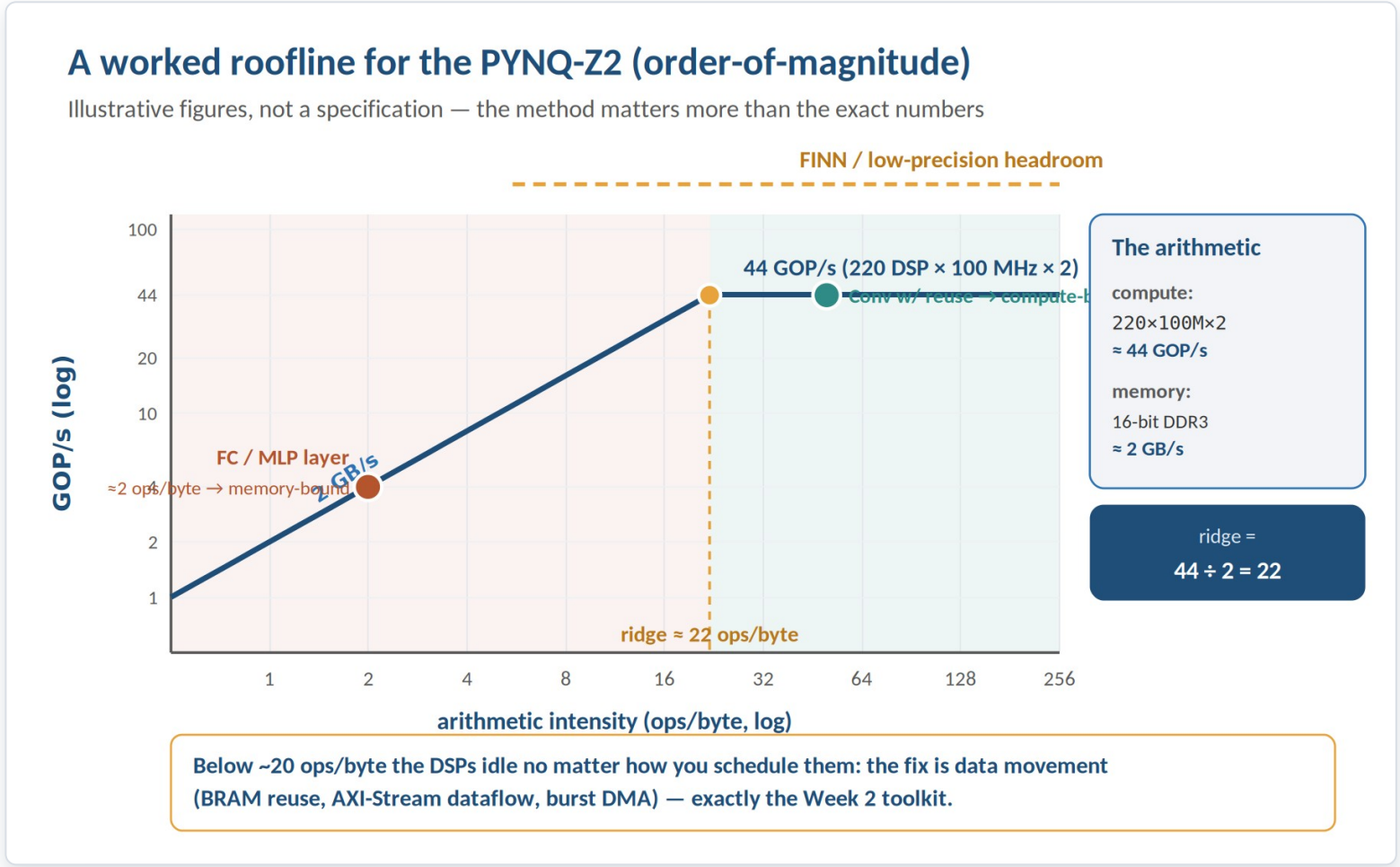
high intensity, grows with map size

usually COMPUTE-BOUND

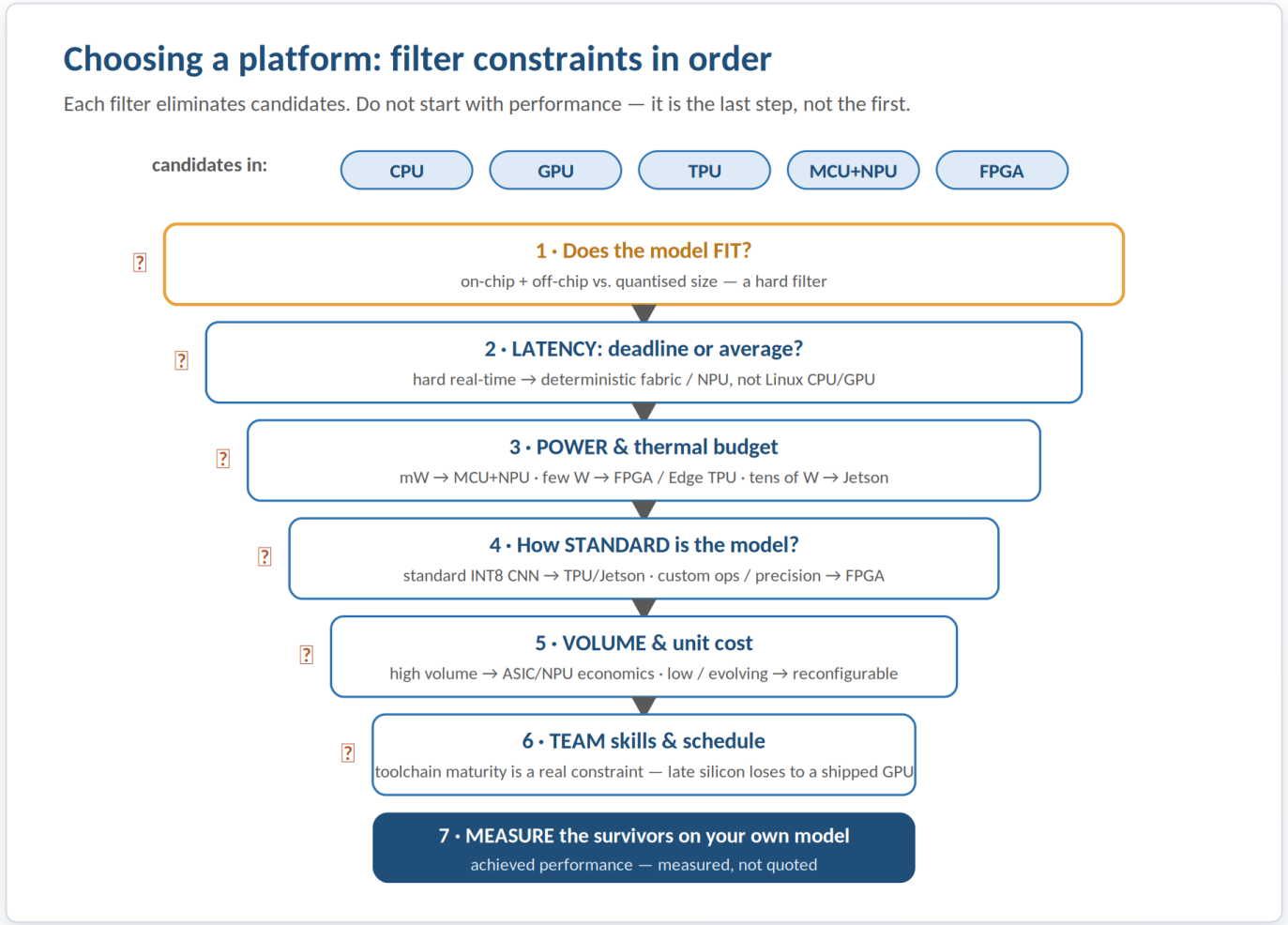
maps well onto MAC arrays

Trap: depthwise-separable convs cut FLOPs but also cut intensity — MobileNets are closer to memory-bound than their FLOP count suggests.

An approximate roofline for the PYNQ-Z2



Filter constraints in order



“Best” always means: for which workload?

Always-on keyword spotting

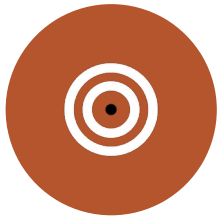
Coin-cell powered, tiny model, latency easily met.

MCU + Ethos-U55

Multi-camera robot

Several concurrent networks, large models, watts available.

Jetson Orin



Fixed industrial inspection

Hard 10 ms deadline, custom low-bit CNN, dataflow on-chip.

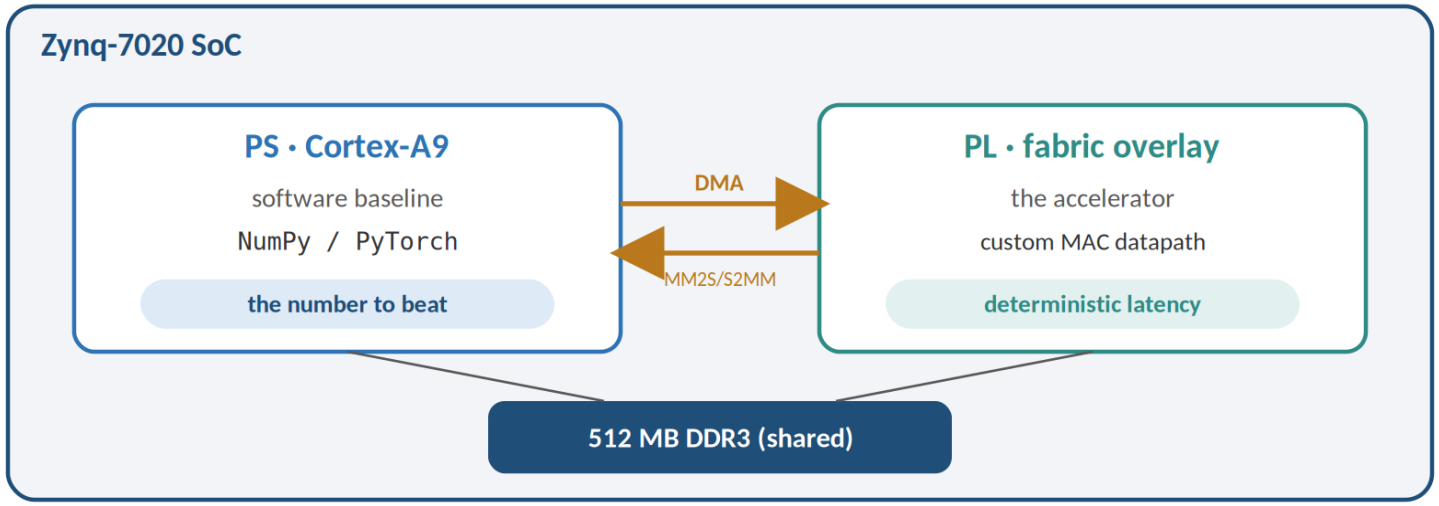
FPGA (Zynq)

A defensible answer cites measured numbers on the shortlisted candidates.

Both sides of the comparison, one board

Both sides of the comparison live on one board

PYNQ-Z2: a software baseline and an accelerator on the same silicon — no cross-vendor confounds



Measure the ratio — like for like

Time both paths the same way: warm up, discard the first run, report median AND the p95 tail.

Include the DMA transfer in the hardware timing — excluding it is the classic way to overstate a speed-up.

A controlled experiment

PS gives the software baseline; PL gives the accelerator — same silicon, no cross-vendor confounds. Measure the ratio.

```
median_ms, p95_ms, fps
```

warm up · discard first run · report median AND the tail. Include the DMA transfer — excluding it overstates the speed-up.

§7 · Watch out

Six ways to fool yourself

Comparing peak TOPS across vendors

Different precisions and power boundaries — compare measured latency on the same model.

Quoting a mean latency

Real-time systems fail on the tail — report median and p95 / p99.

Excluding data movement

If DMA and host copies aren't timed, the speed-up isn't real.

Assuming fewer FLOPs = faster

Depthwise-separable convs cut intensity, pushing layers memory-bound.

Speed-up without resource cost

On FPGA, performance is meaningless without the LUT / BRAM / DSP budget.

Forgetting the model-size filter

A platform that can't hold the model doesn't belong on the shortlist.

§8 · In one slide

Key takeaways

- Peak TOPS is a ceiling, not a promise — achieved performance on your model settles the argument.
- Compare on five axes at once: TOPS/W, latency (with its tail), area/resources, $/TOPS, model-size.
- Families trade flexibility for efficiency: CPU → GPU → FPGA → TPU / NPU.
- The roofline separates memory-bound from compute-bound; the ridge tells you which side you're on.
- PYNQ-Z2 (≈44 GOP/s over ≈2 GB/s, ridge ≈22): low-reuse layers need data-movement fixes, not more MACs.
- Filter constraints in order — fit, latency, power, standardness, cost, schedule — then measure the survivors.

References & resources

Verified against live sources

PYNQ — Getting Started
Board images, overlays, notebooks — basis for Lab 2.
pynq.readthedocs.io

PYNQ — PS/PL Interfaces
GP/HP ports, MMIO, allocate, DMA — the §4.3 data paths.
pynq.readthedocs.io

Xilinx FINN
End-to-end quantized inference on PYNQ — low-precision route.
github.com/Xilinx/finn

Zynq-7000 TRM (UG585)
PS, PL and memory subsystem behind the roofline estimate.
docs.amd.com

Williams, Waterman & Patterson (2009)
“Roofline: an insightful visual performance model” — source of §4.
dl.acm.org

Coral — Edge TPU benchmarks
4 TOPS, ~2 TOPS/W, per-model inference times.
coral.ai/docs

NVIDIA — Jetson Orin family
Orin Nano → AGX Orin: TOPS and power envelopes (§3.2).
nvidia.com

Arm — Ethos-U55 microNPU
Area and performance claims (§3.4).
arm.com

NEXT

Lab 2 — measure it yourself

Run the same small CNN on (a) the PS with NumPy/PyTorch and (b) a pre-built PYNQ overlay. Record latency, throughput and — where measurable — energy, then place both on the roofline you built today and explain the gap.

"Which platform is best?" always means **"for which workload, under which constraints?"**