



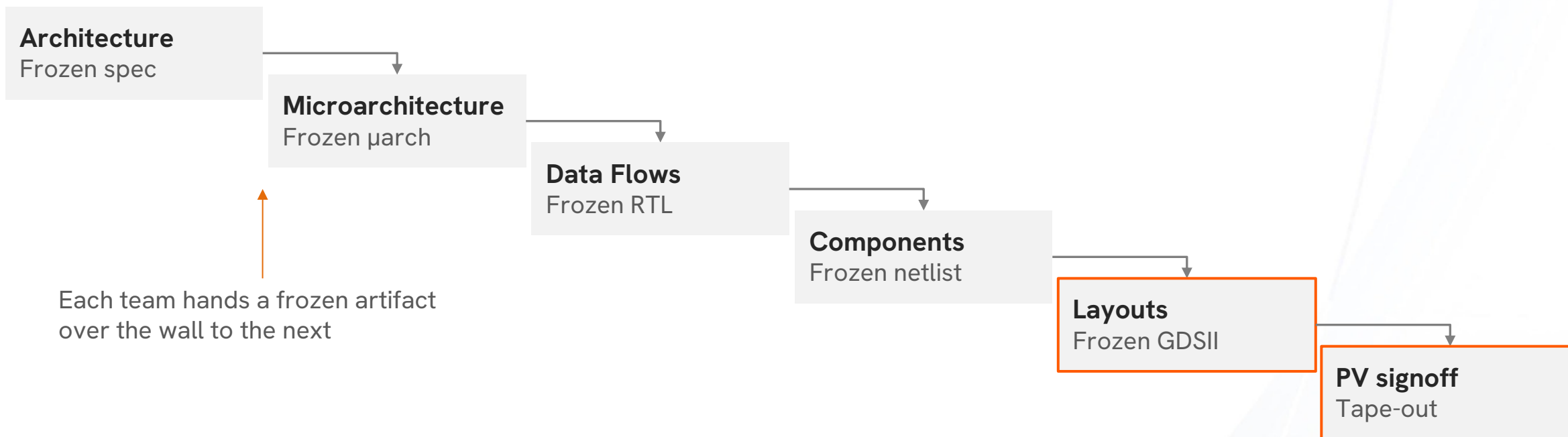
Stelios Diamantidis
Founding CPO

From Linear Pipelines to Spherical Optimization

Spec-driven silicon design
via physics-informed
foundation models

EDPS 2026

Chip design runs as a waterfall (linear pipelines)



When physical constraints become visible

Thermal, parasitics, geometries: estimated or unknown

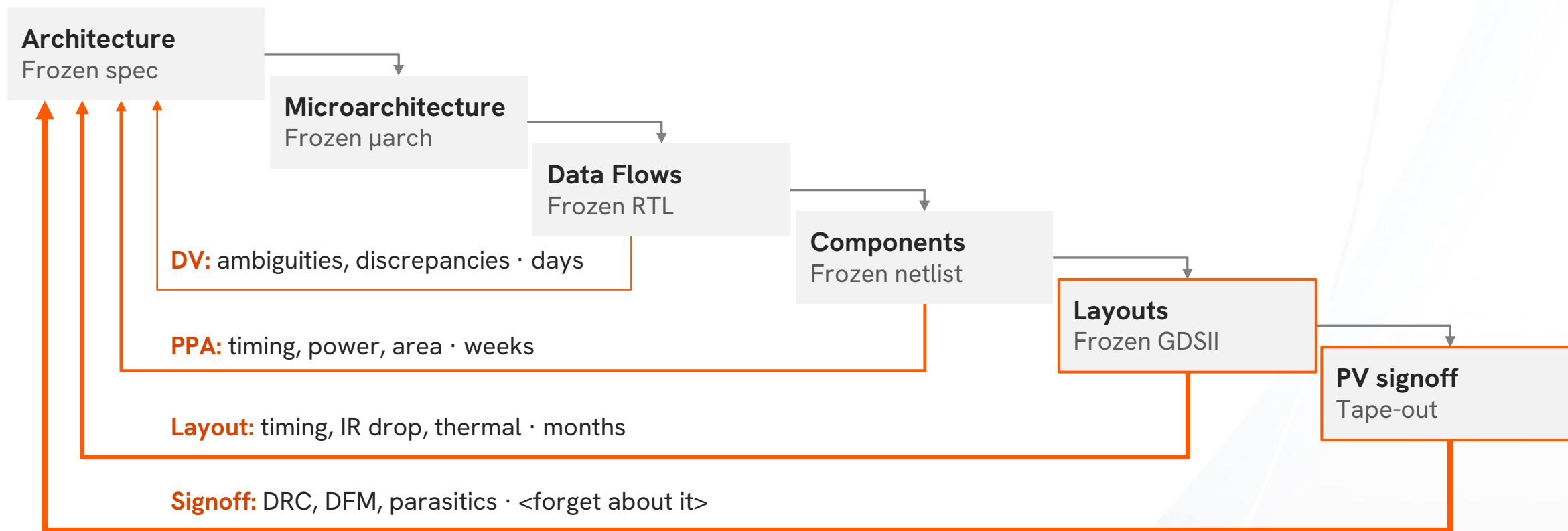
Physics measured for the first time

How much of the design can still change

High: architecture, μarch and logic still open

Low: only local fixes remain

The compounding penalty of the design waterfall



How much time and effort we spend

Architect

Execute: traverse the execution waterfall, address issues, deal with loops

Critical transition point from functionality to physics

1 Slow, under-sampled iteration

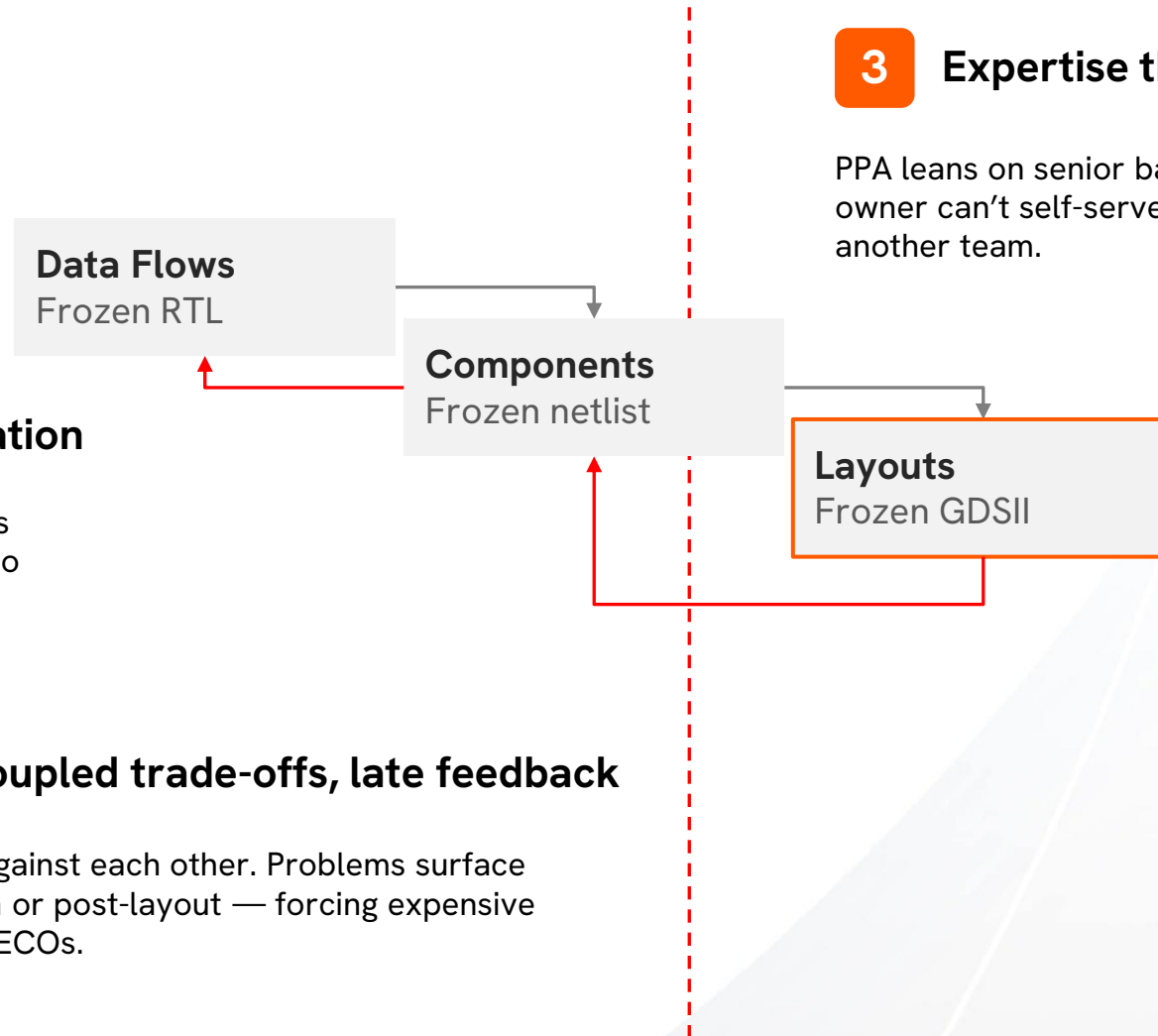
Every synth → report → tweak cycle runs hours to days, and full EDA sweeps are too costly to explore the space.

2 Coupled trade-offs, late feedback

PPA pull against each other. Problems surface post-synth or post-layout — forcing expensive back-end ECOs.

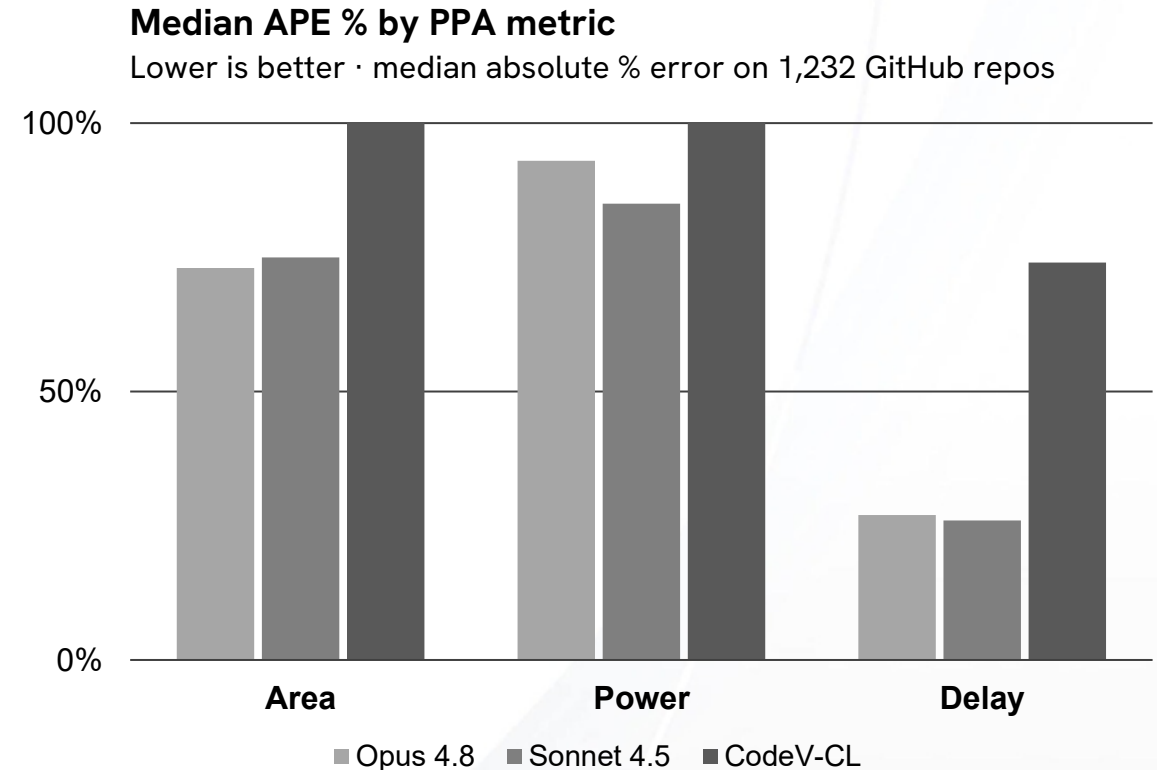
3 Expertise that doesn't scale

PPA leans on senior back-end intuition. The RTL owner can't self-serve; every "what if" waits on another team.



Agentic AI can speed up stages but fail physics

- Model Foundation
 - LLMs trained on tiny corpus of public designs; not grounded in physics
- Accuracy & QQR
 - Good on high level reasoning but hallucinate physical impact
- Convergence & Latency
 - Fragile – expensive retry loops, long prompts, slow tool runs
- The token cost trap
 - \$30k - \$50k per engineer / month
= paying models to think



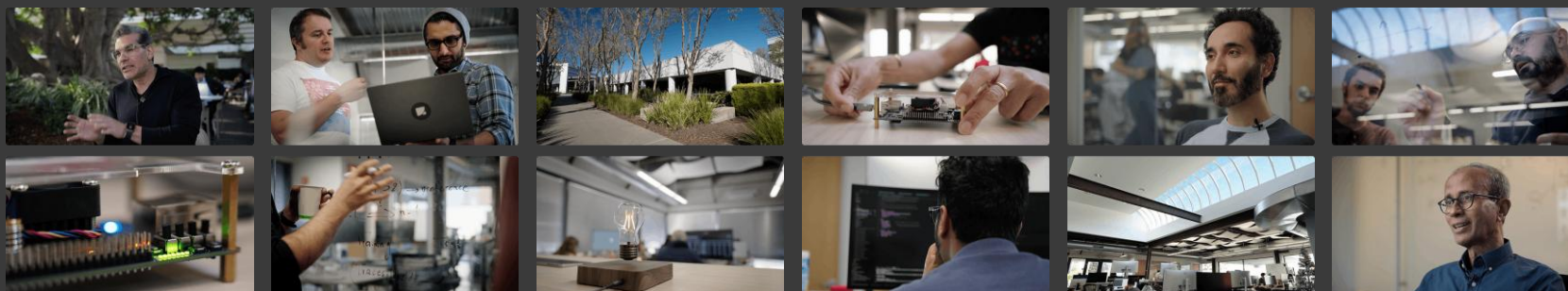
Cognichip – Builders of Artificial Chip Intelligence ACI™

Founded in 2024, \$93M in Total Funding

Deep Experts in Semiconductors and AI – Multiple IPOs



World class talent



100s of Tapeouts
Completed

>40% Staff
PhDs

4 Olympiad Medalists
9 Google Scholars

40,000+
Academic Citations



Cognichip ACI Enterprise

Four industry firsts in one system.



Purpose-built models

Physics-informed — not retrofitted LLMs and wrappers



Full-stack intelligence

A unified, secure, self-learning flywheel for the enterprise



AI-Native workflows

Specs, RTL, netlists, circuits — the model is the design



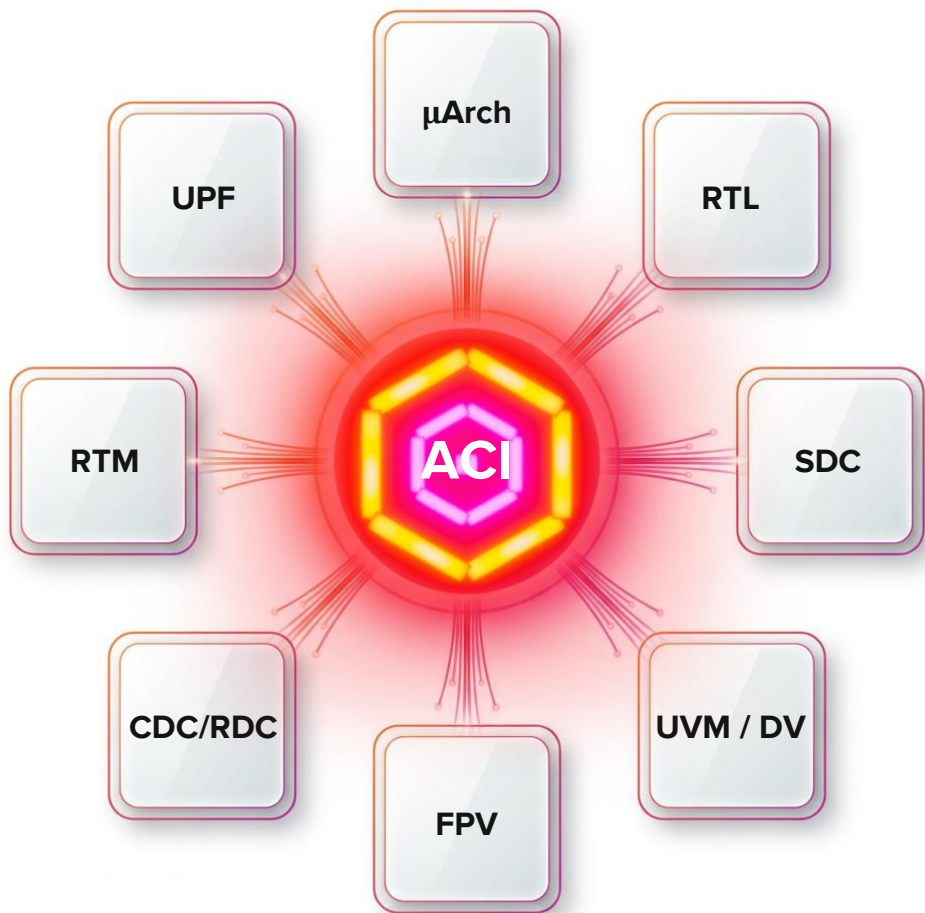
FPGA casting

Hardware development at software speed



Cognichip ACI Physics-informed Foundation Models

Three key technology innovations at the heart of Cognichip's design system



Mixture of Models (MoM) built for design tasks

Aligned to semiconductor data abstractions – specs, RTL, netlists, layouts, tests, PPA metrics, and more

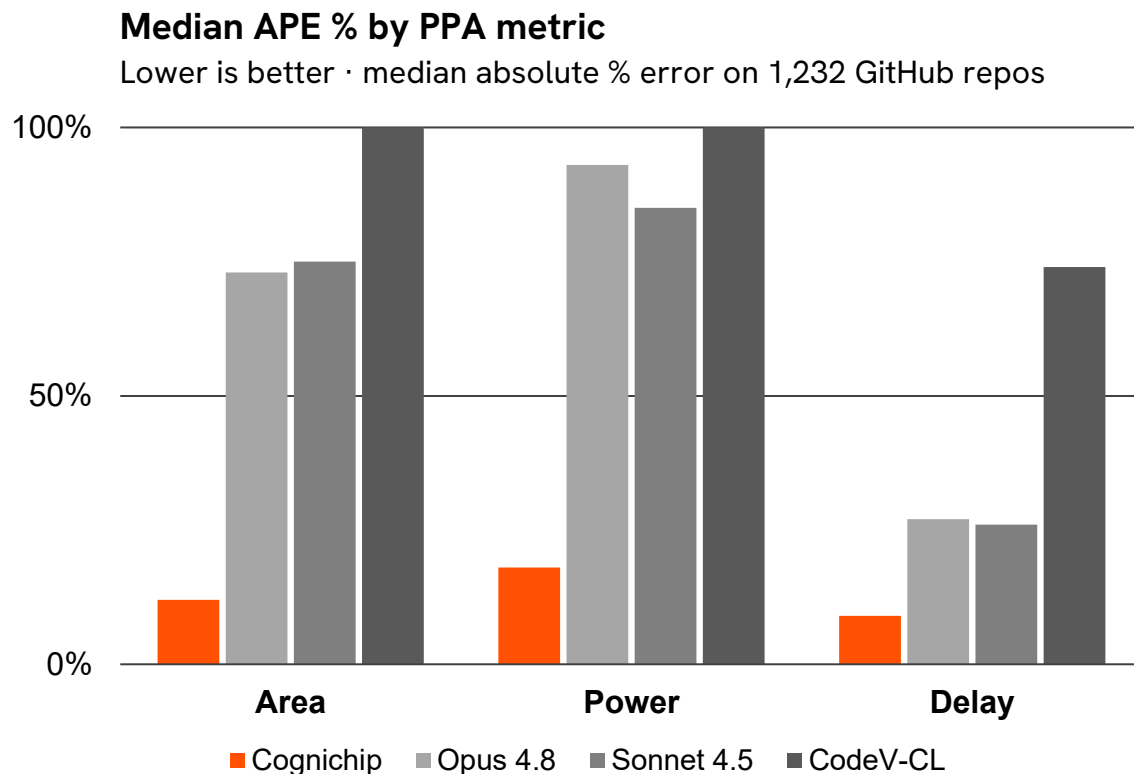
3M+ training samples under governance

Expansive dataset – Designed from scratch or synthetically generated, grounded by **synthesis, simulation, formal and silicon**

Highest accuracy

Design logic baked into the models through reinforcement learning for complex multi-step design reasoning

Cognichip ACI: Close to ground truth



- **Model Foundation**
 - Physics-informed, purpose-built and trained for chip design
- **Accuracy & QoR**
 - Stays close to ground truth, avoids excessive agentic guardrails
- **Convergence & Latency**
 - Highly convergent inference, physics-informed decomposition
- **Cost Efficiency**
 - Right model for the right task:
100x faster, cheaper

ACI: The Model is the Design.

The specification is not an input you consume once and discard. It is the connective tissue: the one authoritative source the model generates from, the testbench grades against, and every change flows through.

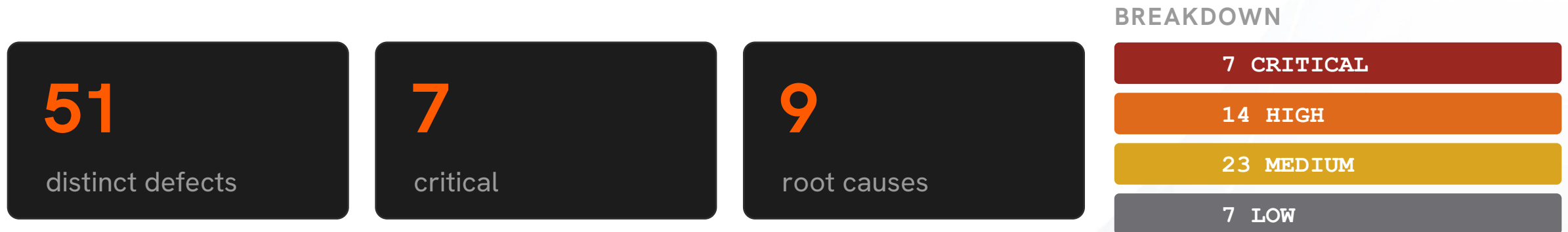
When the model, the spec, and the design are one and the same, you move fast and still sign off.

1. Identify spec problems before implementation

This is the real Ultra Ethernet interconnect specification, a live committee document still being written by the Ultra Ethernet Consortium.

The stakes: Companies are implementing this same committee document right now, independently, and we don't control how carefully each of them reads it.

Miss a contradiction and your interpretation diverges from theirs, silently, until two independently-built devices meet at a link and do not interoperate.



Spec contradictions surface before a single gate exists

One rule, three link-killing bugs

The rule is stated once, in one clause. The design has three separate places that must remember it, and don't.

§5.2.4.2

States the specification's own rule for comparing cyclic counters — then never propagates it into the three places that depend on it.

One rule.

CRITICAL

Credit-freed counter aliases to zero

At CreditSize = 32 B, CF_max_timer = 1 MB, exactly 2^{15} credits are freed per interval, wrapping the 15-bit counter. credits_returned computes 0 while 32,768 were freed. The lossless VC deadlocks.

CRITICAL

Consumed counter wraps silently

The CC update interval is not bound to the 20-bit counter modulus, so the consumed count wraps unnoticed.

CRITICAL

LostCredits sign test on a cyclic counter

A "> 0" arithmetic test on a modular value inverts the lossless guarantee into a buffer overflow.

The corrected spec becomes single source of truth

The output is the corrected specification itself. Where a decision resolves the spec, it is recorded in-line, tagged, and traceable, verbatim:

ENRICHED SPEC · ENGINE BIT COUNTER



[ADDED — DISPOSITION AF-01 — Credit-freed counter.] The credit-return counter shall be sized such that its modulus exceeds the maximum number of credits that can be freed within any single reporting interval, under all supported configurations of CreditSize and CF_max_timer. This addition addresses the error in section §5.2.4.2 stating the rule for comparing cyclic counters.

The **[ADDED]** tag states the rule, cites the source, and explicitly names which clause it replaces: here, §5.2.4.2's rule on comparing cyclic counters . One authoritative answer, where the ambiguity used to be.

All 51 defects (7 critical, 14 major, 23 minor) are folded into the spec the same way, read once by the coder, once by the testbench.

2. Reveal impact of design decisions on PPA

Two design-time interactions on the tensor FMA block, 2.0 ns target.

Example 1 · Predict PPA

> What’s the PPA of the tensor FMA block?

Here are the **PREDICTED** PPA metrics for the **tensor FMA block** (`txfma_top_isolated`) at a 2.0 ns clock period:

Metric	Value	Notes
Cells (area proxy)	6,435.55	Synthesized gate count
Max Frequency	232.43 MHz	Predicted achievable clock frequency
WNS	-2.30 ns	Timing violation — predicted critical path is 4.30 ns, exceeding the 2.0 ns target
Power	82,160.30 μ W	Total predicted power

Key takeaway: The design has a significant **timing violation** (WNS = -2.30 ns) at the 2.0 ns target period. The critical path is roughly 4.30 ns, so the design currently closes at ~232 MHz rather than the ~500 MHz target implied by the 2.0 ns constraint. This is a datapath-heavy FP FMA array — pipelining and retiming are the natural levers to improve WNS.

A predicted baseline in seconds; pointing directly at the adders.

Example 2 · Optimize PPA

> Collapse the normalization shifter’s two-barrel shifters into one using bit reversal and regenerate PPA.

Here's a clean before/after comparison:

Metric	Baseline	Bit-Reversal (this change)	Delta
Cells	6,435.55	6,525.01	+89 (+1.4%)
Max Freq	232.43 MHz	245.23 MHz	+12.8 MHz (+5.5%)
WNS	-2.30 ns	-2.08 ns	+0.22 ns improvement
Power	82,160 μ W	75,565 μ W	-6,596 μ W (-8.0%)

Summary: The bit-reversal collapsing is a net win. The synthesizer fused the two barrel-shifter inference paths into a single shared shift tree — resulting in:

- -8% power (fewer switching nodes in the shift datapath)
- +5.5% max frequency / +0.22 ns WNS (shallower mux fan-in on the critical path)
- +1.4% cells — a small overhead from the two sets of 2:1 pre/post reversal muxes, which is well worth the power and timing gains

We're still 2 ns off closure at the 2.0 ns target. Want to continue optimizing — perhaps pipelining the shift stage or attacking another part of the critical path?

A directed edit, re-scored instantly: **-8% power, +5.5% freq** — but timing still isn’t closed.

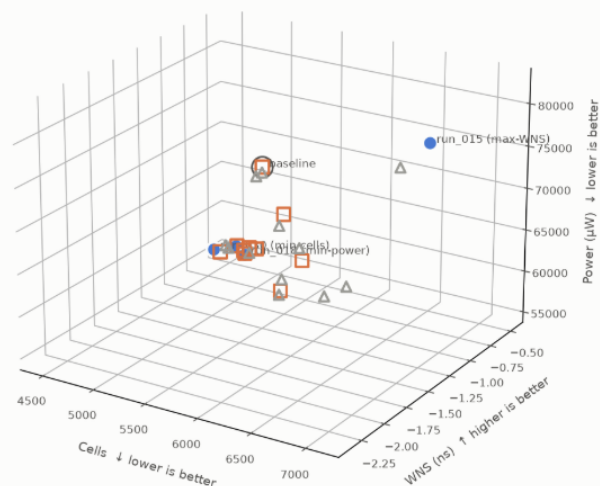
From one data point to a full Pareto frontier

> Now that we have some headroom in our power from the shifter modification, can you find opportunities to improve FMAX while not increasing power more than 8%?

PPA Pareto search

26 measured states - 4 on the front

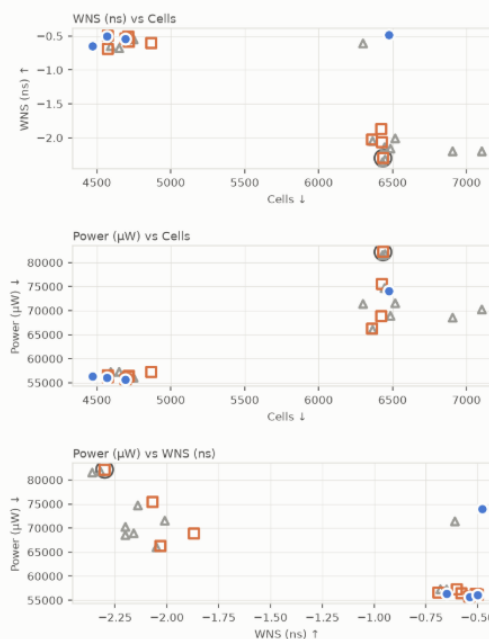
search target: 24 — 26/24 measured states reached



△ measured (rejected/unlabeled) □ accepted (dominated) ● Pareto front ○ baseline (run_000)

Pareto front — nothing measured beats it on all three axes accepted (dominated) — a move the search kept, since beaten by another state measured — tried, not kept

baseline — the state the search started from



BASELINE → BEST FRONT STATE

WNS	-2.30 → -0.54 ns	+1.76 ns
Max freq	232 → 394 MHz	+70%
Cells	6,436 → 4,691	-27%
Power	82,160 → 55,653 μW	-32%

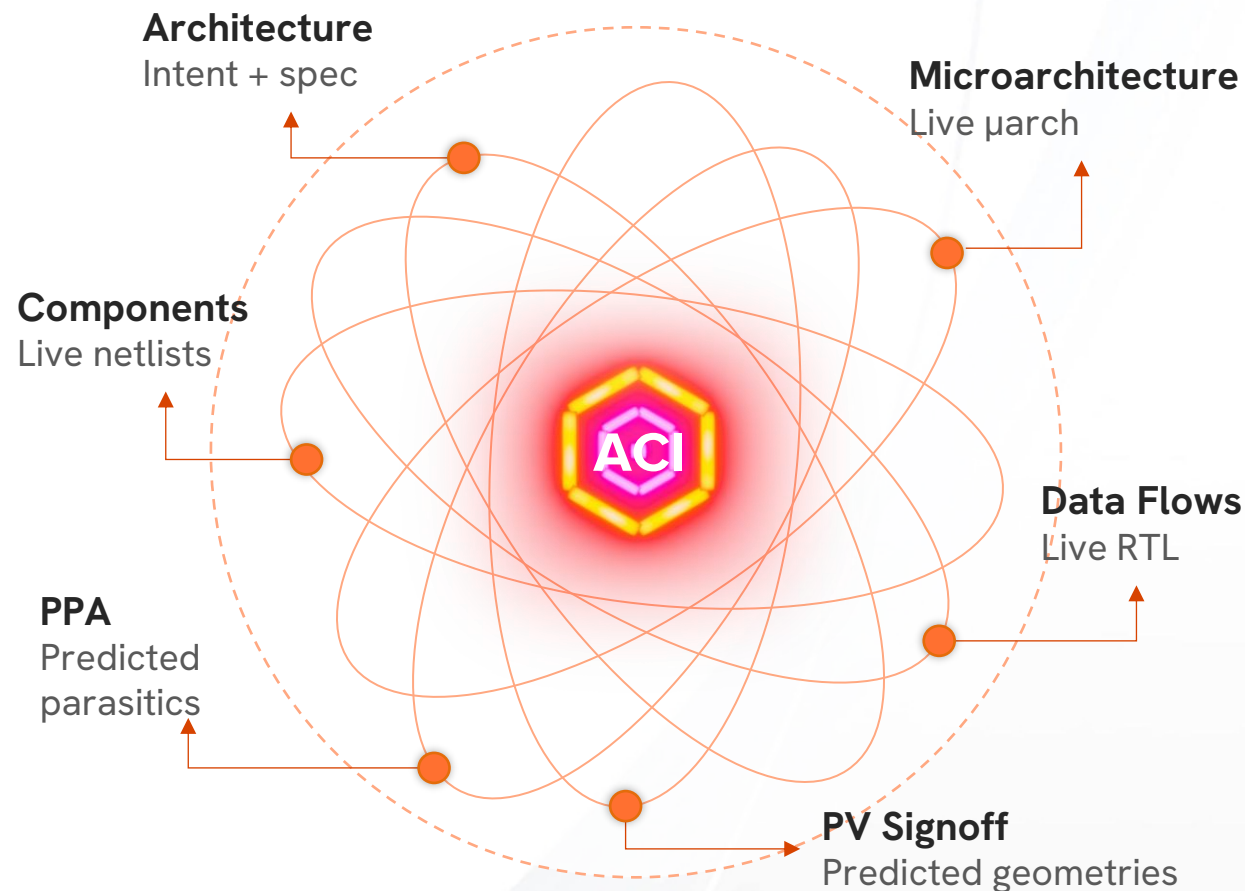
Budgeted +5% power — the search cut power ~32% and recovered ~77% of the WNS gap.

Every move is scored and logged

Impact to functionality is checked against every proof certificate.

Solving the compounding penalty of waterfall

- Every design dimension evaluated at once
- Design abstractions calculated, not deferred
- Omni-directional visibility and explainability
- Dynamic updates and inline formal validation
- From Serial to *Spherical* optimization



How much time and effort we spend

Architect: direct intent, choose from the trade-off frontier

Execute: resolved concurrently

Beyond agents: AI-native design workflows

Design Requirements

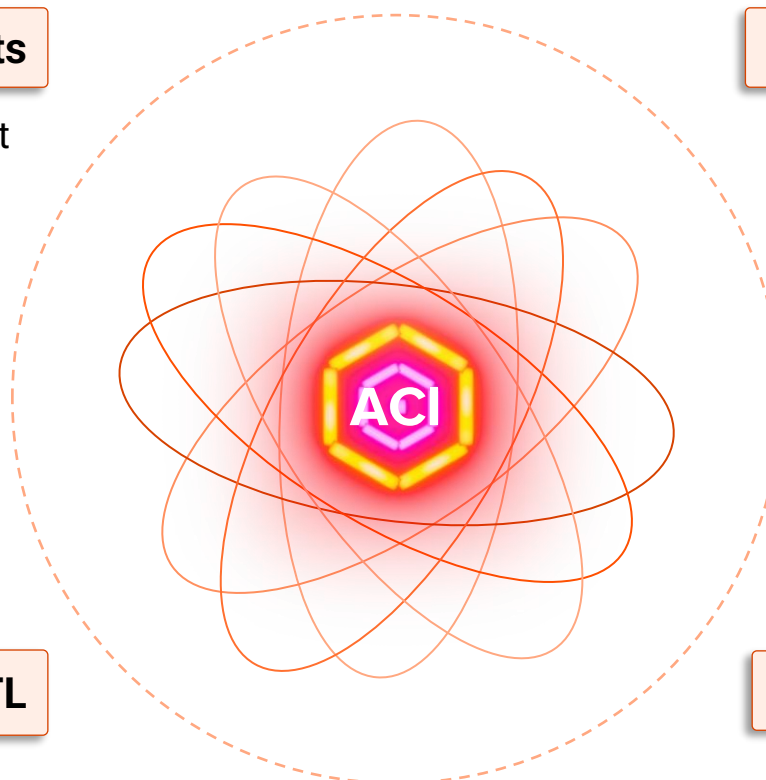
Prior RTL, IP, specs, csv, traces — any data input

Microarchitecture Specs

Every atomic requirement is formally verified and held throughout the design lifecycle

Formally Verified RTL

Emitted — not generated; built to spec every time



Zero-touch Functional Closure

Intelligence-driven UVM testbenches drive tests to coverage closure, tracing flaws back to the spec.

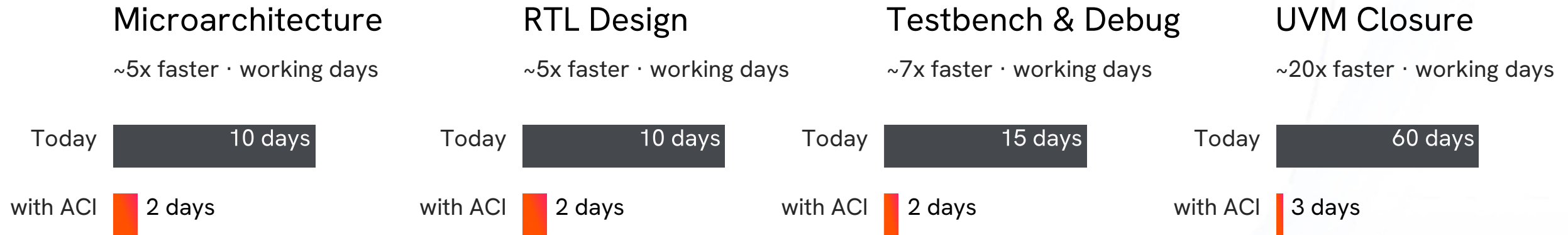
PPA Optimization

Critical transformations happen at inference speed — not as retroactive synthesis layers.

Pareto-aware Design

No zero-sum trade-offs; ACI pushes the performance-power-area frontier outward.

Months of Effort Collapse into Days



One detailed 55-page spec | One engineer | 20 weeks down to a few days

Microarchitecture spec, RTL, middleware, testbench authoring, functional closure, debug and finalize — all completed in a few days

Spec-driven, physics-informed, concurrent

1 The waterfall pays for physics late

Timing, IR drop, thermal and DFM surface at layout and signoff, and each surprise loops the design back to architecture.

2 Encode physics at spec time

A physics-informed model predicts PPA, formal, parasitics and DFM from the spec, before any gate exists.

3 Optimize spherically, direct intent

Every design dimension moves at once; engineers set intent and pick a point on the trade-off frontier.

Consider the possibilities

Bring one block's spec to the problem. Compare what formal validation and physics-aware prediction at spec time finds against your current flow.



**What will YOU
design today?**

Thank you