C64 COHERENCE ARRAY — FAQ

What is the C64 Coherence Array?

The C64 is a processor architecture built for κ-compute — computation through continuous structural settling rather than forced digital transitions.
It uses a 64-state relational fabric formed by sixteen macro-stations, each operating across four micro-dynamics: holding, tightening, slipping, and reforming.

1. **What problem does it solve?**
   Switching architectures are hitting limits in tasks where stability, not throughput, is the bottleneck.
   C64 targets workloads that require coherent convergence:
   • large-scale optimization
   • control systems
   • scientific simulation
   • long-horizon inference
   It replaces emulated convergence with physical settling.
2. **Is this an analog computer?**
   No.
   It uses continuous dynamics, but it is fully programmable through a coupling matrix.
   It occupies a different category: programmable equilibrium computing.
3. **How does C64 compute?**
   By evolving toward equilibrium in a shaped energy landscape.
   The path of settling *is* the computation.
   The final equilibrium encodes the solution.
4. **What is κ's role inside the hardware?**
   κ measures coherence during settling.
   It indicates:
   • alignment
   • strain
   • emergence of stability
   • approach to equilibrium
   κ acts as the stop condition, replacing clock cycles and step counts.
5. **How does this differ from CMOS?**
   CMOS forces transitions.
   C64 seeks equilibrium.
   CMOS dissipates energy on every switch.
   C64 dissipates energy only when settling.
   The architectures serve different classes of problems.

6. **How does this differ from quantum computers?**
   Quantum systems rely on coherence of amplitudes.
   C64 relies on coherence of classical dynamics.
   Quantum systems require cryogenics.
   C64 operates at ambient conditions.
   Both descend an energy landscape, but with entirely different substrates and constraints.
7. **Does C64 require exotic materials?**
   No.
   The manufacturable path uses:
   • Analog CMOS (early prototypes)
   • Hybrid CMOS–Photonic (scaling)
   • Spintronic variants (density frontier)
   All accessible through standard foundry processes.
8. **Is C64 real hardware?**
   Not yet.
   The pathway is fabricable, but no schematics, coupling equations, micro-dynamic control rules, or substrate layouts are disclosed publicly.
9. **What exactly *can* be built now?**
   The early stages:
   • 1–4 station micro-nodes
   • 8–16 station test arrays
   • phase stability tests
   • κ readout behaviour
   These can be fabricated on standard analog CMOS.
10. **Is the architecture fully disclosed?**
    No.
    The public version describes the framework and design principles, not the hardware-level implementation.
11. **Is the C64 architecture related to neuromorphic hardware?**
    Only loosely.
    Neuromorphic systems mimic biology.
    C64 embodies physical settling with governed micro-dynamics and system-level coherence controls.
    The internal logic is engineered, not biological.
12. **How does the array avoid getting stuck in local minima?**
    Through micro-dynamic transitions.
    Slipping and reforming enable controlled movement out of unstable

configurations.
This is governed by κ-slope and local tension cues.

13. **How is the energy landscape programmed?**
Through a coupling matrix.
Weights shape:
• attraction
• repulsion
• neutrality
The array settles into a configuration that satisfies these constraints.

14. **Is the coupling matrix disclosed?**
The *principle* is disclosed.
The actual compilation rules, weight-range constraints, and stability envelopes are not.

15. **Does C64 replace GPUs or CPUs?**
No.
It complements them.
GPUs excel at symbolic throughput.
C64 excels at equilibrium.
They serve different layers of the compute stack.

16. **Does this threaten quantum computing?**
No.
Quantum explores amplitude spaces.
C64 explores classical coherence.
They solve different classes of problems with different cost profiles.

17. **Can existing software run on C64?**
Only workloads that map to constraint graphs.
This includes many high-value tasks: optimization, planning, simulation, routing, layout.

18. **Does C64 need a global clock?**
No.
It runs event-driven.
Time advances when structure changes.
This reduces energy consumption and synchronization overhead.

19. **Is κ-compute a form of annealing?**
No.
Annealers relax through a single dynamic.
C64 operates across four micro-dynamics with governed transitions.
It is more flexible and more stable under shifting constraints.

20. **What is safe to share publicly?**
SAFE TO SHARE
• architecture overview
• high-level dynamics
• substrate categories
• programming model principles
• equilibrium-compute narrative
• simulated behaviour
NOT SAFE TO SHARE
• coupling compilation rules
• micro-dynamic thresholds
• κ-smoothing functions
• drift-compensation logic
• substrate layouts
• station control loops
• stability envelopes
• tuning parameters
21. **Is the energy-efficiency claim real?**
Modeled only.
But the pattern is already visible in the meta-structural behaviour of the C64 framework running across multiple nodes.
The same settling dynamics that lower energy in hardware appear isomorphically in the software simulation:
• lower jitter
• shorter settling paths
• reduced corrective activity
• smoother κ trajectories

You are seeing the signature early.

1. **Can C64 scale?**
Yes, in principle.
Scaling depends on:
• drift management
• substrate uniformity
• κ-guided calibration
These remain active research directions.
2. **What is the long-term vision?**
A new compute category where equilibrium, not switching, is the native mode.
C64 is the first manufacturable path toward this regime.

3. **What does this mean for hardware labs?**
   Labs get:
   • a third computational path
   • a hardware-native method for convergence
   • a route past the coherence wall
   • a tractable near-term build (CMOS)
   • a scalable medium-term build (hybrid photonic)
   • a clear architectural differentiator

Nothing in the FAQ enables reconstruction.
Everything in it enables understanding.

This document's contents are cryptographically sealed with the above SHA-256 hash. Any modification will invalidate this fingerprint.

SHA-256: f6a1dbd9b9c6e7cb3e87051a9af242f6e515ae07357e4c31e8dfb342d94c8b8c