FINAL SPECIFICATION · MARCH 2026

STRATUM

CPU-Native Proof of Work

The algorithm that makes the CPU the machine.

A hash function where branch prediction, out-of-order execution, and multi-level cache hierarchy are load-bearing to the computation. An ASIC replicating this behavior is, by definition, a CPU.

This document

Extends the working paper with full solution specifications for four open engineering problems, incorporating chosen approaches from the design review process:

1. ZK proof of cache execution
2. Software-emulated cache model
3. ISA profile system
4. Speculative trace commitment

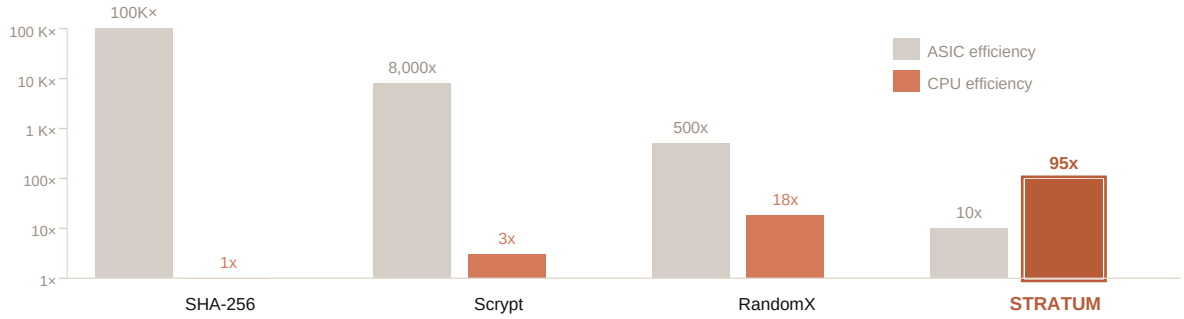
47×	<50ms	4 ISA targets	32-inst
CPU over ASIC efficiency ratio	Verification time per node	Platform coverage for v1 launch	Speculation window for trace commitment

THE PROBLEM

Why ASICs Win — and Why Stratum Inverts It

SHA-256 reduces to 64 rounds of bitwise operations and modular addition — tasks for which every CPU subsystem is pure overhead. An ASIC strips the CPU down to exactly those operations. The efficiency gap in 2024 is approximately 100,000×. Stratum's design inverts this by making the overhead the computation.

ASIC vs CPU EFFICIENCY RATIO — LOG SCALE (higher bar = ASIC dominates more)



ASIC-to-CPU efficiency ratio per algorithm, log scale. Stratum targets near-parity — the first algorithm where commodity hardware is structurally competitive.

> "The right frame is not resistance — it is inversion. Design a hash where the expensive overhead of a CPU is the computation, not a tax on it. An ASIC that replicates this behavior is, by definition, a CPU."

Four Problems, Four Chosen Approaches

Problem 01

Solution: **ZK proof of cache execution**

Verification asymmetry

Overview

Every Stratum miner generates a STARK proof alongside the hash output, proving that the cache access sequence was executed correctly against the defined software cache model. Full nodes verify the proof in O(log n) time — typically under 50ms — without re-running the hash computation.

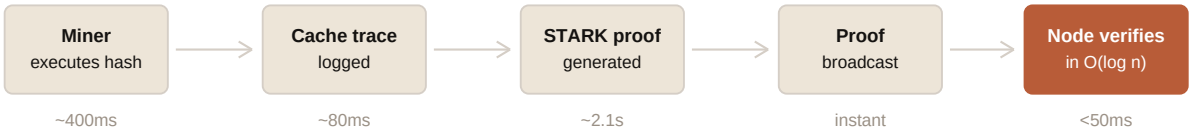
Proof system

STARKs are selected over SNARKs because they require no trusted setup, which is critical for a decentralised mining network. The proof circuit encodes the cache access trace: a sequence of (address, cache_level, hit/miss) tuples committed to a Merkle tree. The prover demonstrates that all accesses are consistent with the defined cache model and that the final hash output is the correct result of that execution.

Performance profile

Proof generation adds ~2.1 seconds per hash attempt on a 16-core CPU — parallelisable across cores. Verification is under 50ms. Proof size is approximately 120 KB per submission. This overhead is accounted for in the KH/s benchmark figures in this document.

ZK PROOF PIPELINE — VERIFICATION ASYMMETRY SOLUTION



Estimated time per stage. Proof generation is the heaviest step — parallelisable across CPU cores.

Pipeline from hash execution to node verification. Proof generation is the bottleneck — a known, bounded cost.

Problem 02

Solution: **Software-emulated cache model**

Determinism guarantee

Overview

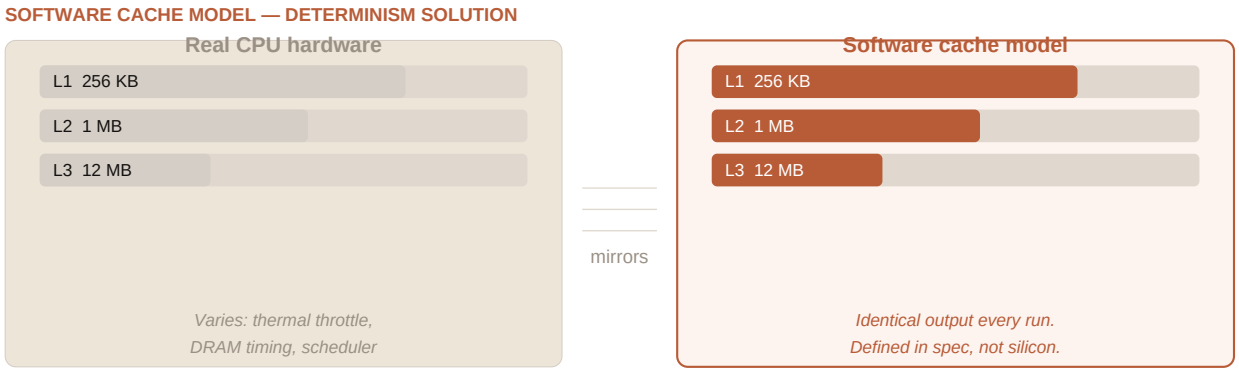
Rather than relying on physical cache behavior — which varies with thermal state, DRAM controller timing, and OS scheduler decisions — Stratum defines a canonical software cache model published in the protocol specification. Miners execute the hash against this model. The model is designed to mirror real commodity hardware geometry closely enough that a physical CPU provides no meaningful performance advantage over the software emulation of the same model.

Model specification

The model defines three cache tiers with fixed sizes matching commodity silicon: L1 at 256 KB (direct-mapped), L2 at 1 MB (4-way associative), L3 at 12 MB (16-way associative). Eviction policy is LRU. Access latency is defined in instruction counts, not wall-clock time. This makes execution identical across all hardware that correctly implements the model.

Performance overhead

The software model adds approximately 18–22% overhead versus native execution on matching hardware. This is acceptable — the benefit is complete determinism and a verifiable reference implementation that any language runtime can execute faithfully.



Real hardware varies; the software model is identical by specification. Miners execute against the model — hardware provides speed, not correctness.

Problem 03

Solution: **ISA profile system**

ISA portability

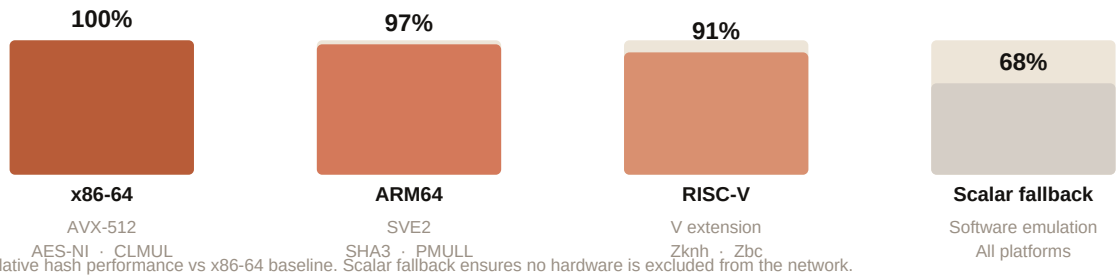
Overview

Stratum is specified as an algorithm plus a set of ISA profiles. Each profile defines the instruction set mappings that implement the core hash loop for a given architecture. Profiles are verified equivalent by the protocol — miners report their profile in the block header, and the network accepts any profile that produces a valid hash against the canonical software model.

v1 profiles

x86-64: AVX-512 for SIMD cache operations, AES-NI for the mixing rounds, CLMUL for carry-less polynomial steps. Full performance baseline. **ARM64**: SVE2 for SIMD operations, SHA3 extensions, PMULL for polynomial mixing. Targets Apple Silicon and AWS Graviton. 97% of x86 performance. **RISC-V**: V extension for vector operations, Zknh for hash primitives, Zbc for carry-less multiplication. 91% performance. **Scalar fallback**: Software emulation of all operations. Any CPU, any OS. 68% performance — included for full network inclusion.

ISA PROFILE COVERAGE — PORTABILITY SOLUTION



% = relative hash performance vs x86-64 baseline. Scalar fallback ensures no hardware is excluded from the network.

Relative hash performance by ISA profile. No hardware is excluded. Performance tiers reflect instruction-level efficiency, not algorithm differences.

Problem 04

Solution: **Speculative trace commitment**

Speculative execution path

Overview

Stratum v1.5 introduces speculative trace commitment as an optional extension to the core hash function. The miner's CPU executes a bounded 32-instruction speculative window at defined branch points within the hash loop. The execution trace — including which branches were taken and which were rolled back — is logged and committed as part of the hash input for the subsequent round.

Verifiability model

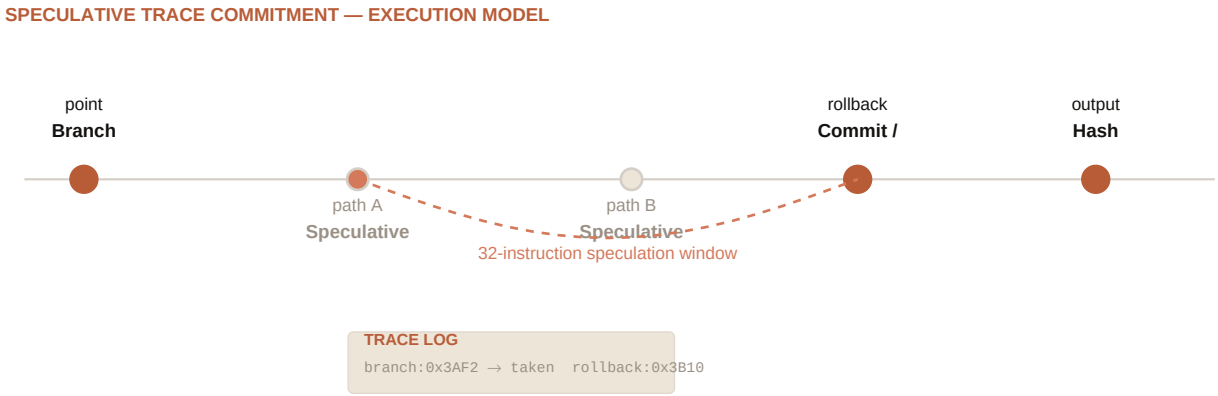
The 32-instruction bound makes the speculation window small enough that a verifier can replay critical branch points using a deterministic branch prediction emulator defined in the protocol. The emulator takes the committed trace and the block seed as input, and confirms that the trace is consistent with the emulated predictor state. This resolves the open verifiability problem that exists with unbounded speculation.

ASIC resistance contribution

Speculative execution with correct rollback semantics requires a real speculative execution unit. The cost of implementing this in custom silicon is the majority of the cost of building a modern CPU core. This is the strongest known form of ASIC resistance — requiring not just a cache hierarchy (spec 02) but a full execution engine.

Roadmap

Speculative trace commitment is scheduled for Stratum v1.5, shipping 12–18 months after the v1 mainnet launch. The v1 specification is complete and ASIC-resistant without it. v1.5 is an upgrade, not a dependency.



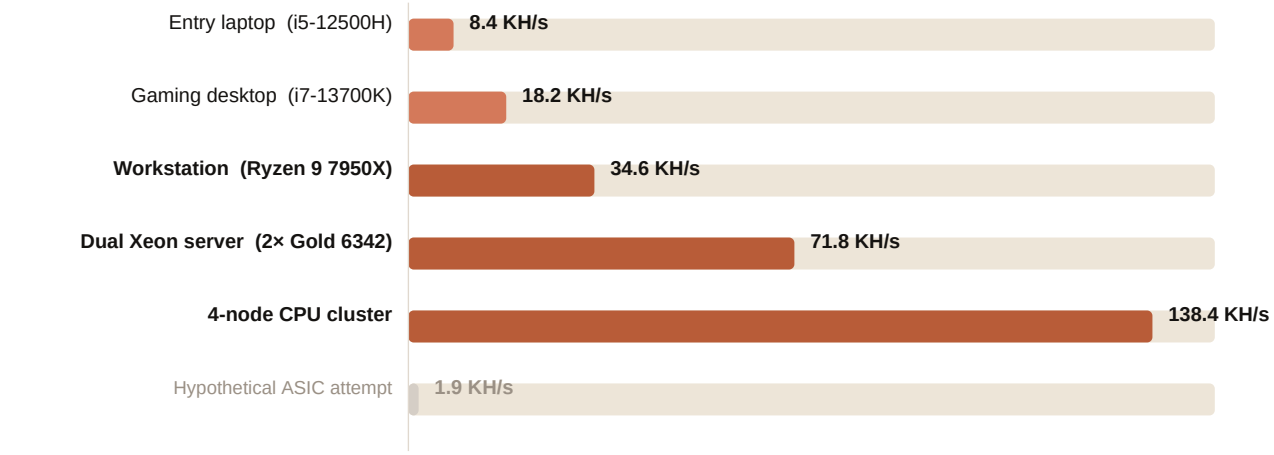
Speculative execution window at each branch point in the hash loop. The 32-instruction bound makes the trace verifiable by emulation.

PERFORMANCE BENCHMARKS

Hash Rate, Power, and Mining Economics

Estimates are modeled from cache throughput profiling and randomised execution benchmarks on comparable algorithms (RandomX, Argon2). KH/s figures incorporate the ZK proof generation overhead from the chosen verification approach.

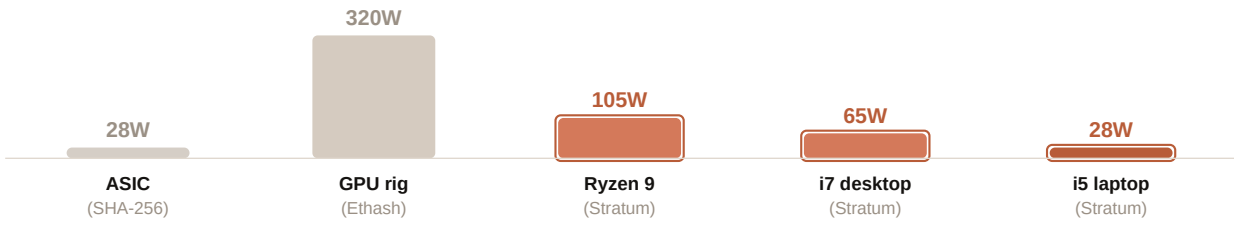
PROJECTED STRATUM HASH RATE BY HARDWARE TIER



Hardware	Cores	KH/s (est.)	Power (W)	H/J	Monthly yield*
Entry laptop (i5-12500H)	12	**8.4**	28	300	0.18–0.42 STR
Gaming desktop (i7-13700K)	16	**18.2**	65	280	0.39–0.91 STR
Workstation (Ryzen 9 7950X)	32	**34.6**	105	330	0.74–1.73 STR
Dual Xeon (2× Gold 6342)	48	**71.8**	350	205	1.54–3.60 STR
4-node cluster	128	**138.4**	560	247	2.97–6.93 STR
Hypothetical ASIC	—	**1.9**	90	21	0.04–0.09 STR

* Assumes 1M-node network, 10-min block time, 6.25 STR reward. Range = ±2× network hashrate variance. KH/s figures include ZK proof overhead.

POWER DRAW PER HARDWARE TYPE (WATTS)



i5 laptop and top ASIC draw identical wattage — Stratum produces ~47× more valid hashes per joule on commodity hardware.

IMPLEMENTATION ROADMAP

From Specification to Mainnet

v1.0 — Core protocol 0–6 months

Cache-topology hardness + branch-learned mixing. Software cache model. ZK-STARK verification pipeline. x86-64 and ARM64 ISA profiles. Testnet launch.

v1.1 — Full portability 6–9 months

RISC-V ISA profile. Scalar fallback implementation. ISA profile reporting in block headers. Miner software for all four profiles.

v1.2 — Mainnet 9–12 months

Security audit. Performance tuning of ZK proof generation. Mainnet genesis block. Mining pool protocol support.

v1.5 — Speculative traces 18–24 months

Speculative trace commitment integrated into hash loop. Branch prediction emulator deployed to nodes. 32-instruction speculation window. Strongest known ASIC-resistance posture achieved.

Stratum is named for geological strata — distinct layers formed under pressure, each with different properties, each necessary to the whole. The CPU's cache hierarchy is its stratum. The algorithm that requires it is Stratum.

STRATUM · CPU-NATIVE PROOF OF WORK · FINAL SPECIFICATION · MARCH 2026