OSH GLOBAL SYSTEMS

ARCHITECTURAL RUNTIME BENCHMARK 3

Resident Runtime V1 Technical Report

OSH structured homeostatic cycle vs. TinyLlama 1.1B Chat Q4_K_M

CPU-only | 30 matched inputs | one-token Transformer baseline | same benchmark family

OSH MEDIAN 30 measured cycles	**6.466 ms**
RUNTIME GAP TinyLlama / OSH median	**2,812.69x**
MEMORY GAP TinyLlama / OSH Working Set	**26.64x**

Integrity status The run completed with OSH structural validation OK, 30/30 unique state signatures, and no FARO production files modified.

Interpretation boundary This is an architectural runtime and resident-memory benchmark. It does not score intelligence, reasoning quality, factual accuracy, or final language quality.

Report date: 12 September 2026
Hardware: Intel Atom D2500 @ 1.86 GHz | 8,172.4 MB physical RAM
Execution platform: Windows | CPU only

1. Executive Summary

This benchmark measures the installed OSH Resident Runtime V1 against TinyLlama 1.1B Chat Q4_K_M under the same benchmark family used for the earlier architectural-runtime comparisons. The OSH side executed 30 structured homeostatic cycles; the Transformer side ran through llama.cpp on CPU with generation deliberately constrained to exactly one token.

Metric	Value	Detail
OSH median	6.466 ms	P95 15.015 ms
TinyLlama median	18,186.463 ms	P95 38,472.047 ms
Median ratio	2,812.69x	TinyLlama / OSH
OSH Working Set	24.883 MB	measured process Working Set
TinyLlama Working Set	662.770 MB	ready process Working Set
Memory ratio	26.64x	TinyLlama / OSH

The measured runtime gap increased substantially relative to the earlier Benchmark 2 report because the Resident Runtime V1 removed repeated state-file reads from the hot path while preserving a changing structured state. The present run reported 30 distinct OSH state signatures across 30 measured inputs.

Key conclusion Under this protocol, OSH Resident Runtime V1 completed the defined structured cycle with a 6.466 ms median, while TinyLlama required 18,186.463 ms median engine time for a deliberately conservative one-token baseline.

2. Objective and Scope

The benchmark is designed to compare local computational runtime cost and resident memory, not semantic quality. This distinction is essential because the systems execute different computational architectures and the Transformer baseline is intentionally minimized to one generated token.

2.1 What is measured

- OSH cycle latency for the structured homeostatic runtime.
- Presence of a changing structured state and speech-act classification.
- C/P/X/T/H state evolution across the 30 inputs.
- OSH process Working Set.
- TinyLlama llama.cpp engine timing and ready Working Set.
- Runtime and memory ratios derived from the measured medians and Working Sets.

2.2 What is not measured

- Intelligence or reasoning quality.
- Factual accuracy.
- Language quality.
- Equivalent task capability between architectures.
- End-to-end application latency including networked services.

Conservative Transformer condition TinyLlama was restricted to exactly one generated token per input, minimizing generation work rather than maximizing language quality.

3. Test Environment and Source Integrity

Item	Measured / reported value
CPU	Intel(R) Atom(TM) CPU D2500 @ 1.86 GHz
Physical RAM	8,172.4 MB
OSH measured inputs	30
TinyLlama measured inputs	30
Execution	CPU only
Transformer model	TinyLlama 1.1B Chat · Q4_K_M
Runtime	llama.cpp
GPU layers	0
Transformer generation	Exactly 1 token per input
OSH validation	OK
Unique OSH states	30 / 30
FARO files modified	None

The console output identifies the run as an architectural runtime benchmark and reports that no FARO production files were modified. The operator states that this benchmark uses the same benchmark harness family as Benchmark 1 and Benchmark 2.

Reproducibility note This report records the measured values shown in the supplied benchmark console output. It does not infer unreported hashes or internal implementation details.

4. Controlled Experimental Design

4.1 OSH phase

The OSH phase measured 30 structured homeostatic cycles. Every cycle returned a changing C/P/X/T/H state and a speech-act label. Structural validation completed successfully, with 30 unique state signatures observed across 30 measured inputs.

4.2 TinyLlama phase

TinyLlama 1.1B Chat Q4_K_M was executed through llama.cpp with zero GPU layers. The reported runtime ratio uses llama.cpp engine timing. Generation was limited to one token per input, creating a deliberately conservative computational baseline.

4.3 Ratio definitions

Runtime ratio = TinyLlama median engine time / OSH median cycle latency.

Memory ratio = TinyLlama Working Set / OSH Working Set.

Timing source The benchmark explicitly reports llama.cpp engine timing as the Transformer ratio timing source.

5. Measured Results

Metric	OSH Resident V1	TinyLlama
Median latency	6.466 ms	18,186.463 ms
Mean latency	7.968 ms	20,511.532 ms
P95 latency	15.015 ms	38,472.047 ms
P99 latency	16.692 ms	42,392.736 ms
Minimum	4.223 ms	1,476.255 ms
Maximum	17.048 ms	43,480.178 ms
Working Set	24.883 MB	662.770 MB
Unique structured states	30 / 30	N/A

The headline measured runtime ratio is 2,812.69x. The measured resident-memory ratio is 26.64x. OSH's median is 6.466 ms, with a P95 of 15.015 ms and a maximum measured cycle of 17.048 ms.

5.1 OSH latency profile

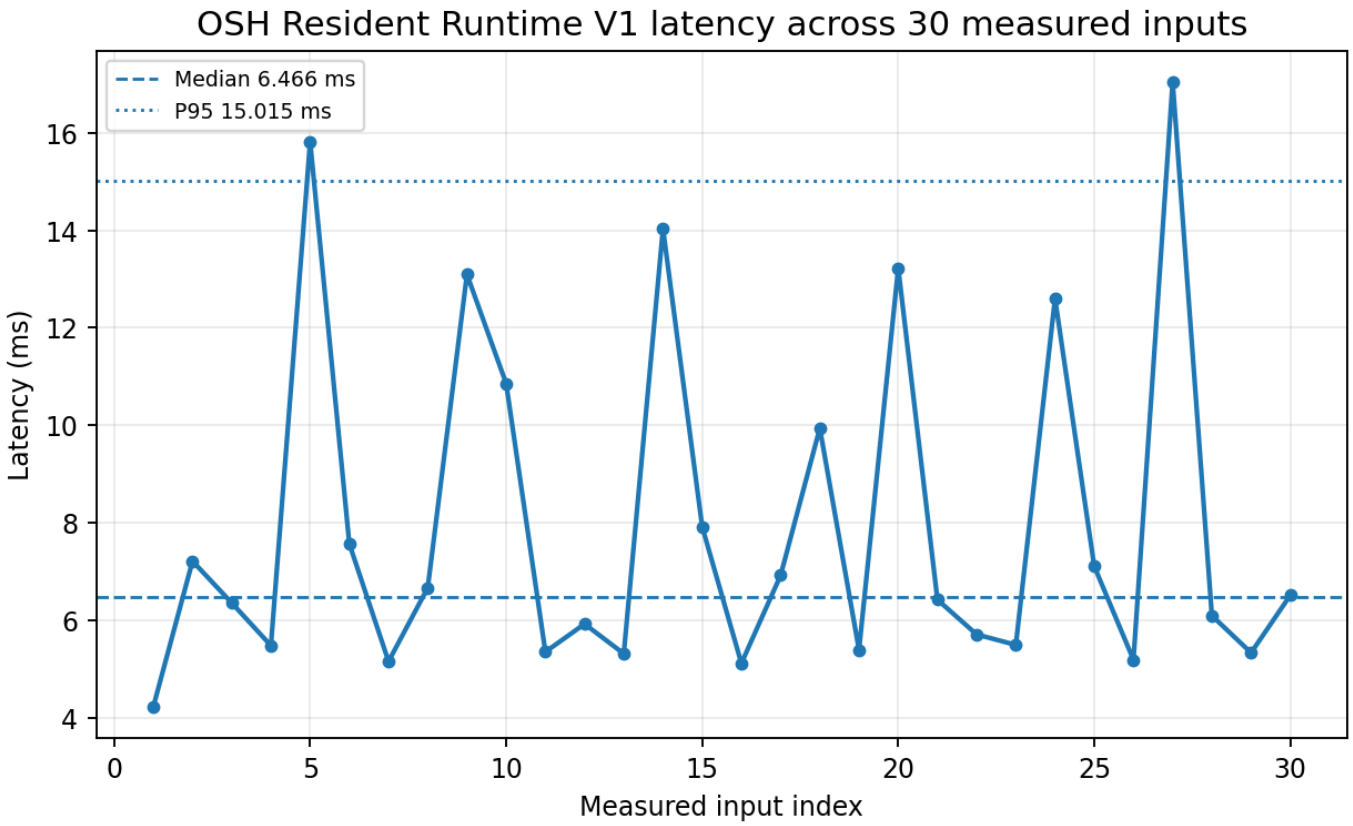

Figure 1. OSH Resident Runtime V1 latency for all 30 measured inputs. Dashed and dotted lines show the reported median and P95.

OSH median latency was 6.466 ms, mean latency 7.968 ms, and P95 15.015 ms. The fastest measured cycle was 4.223 ms and the slowest was 17.048 ms.

Thirty distinct homeostatic state signatures were reported across the thirty measured inputs, indicating that the measured cycles were not accepted as a constant-state execution.

5.2 TinyLlama latency profile

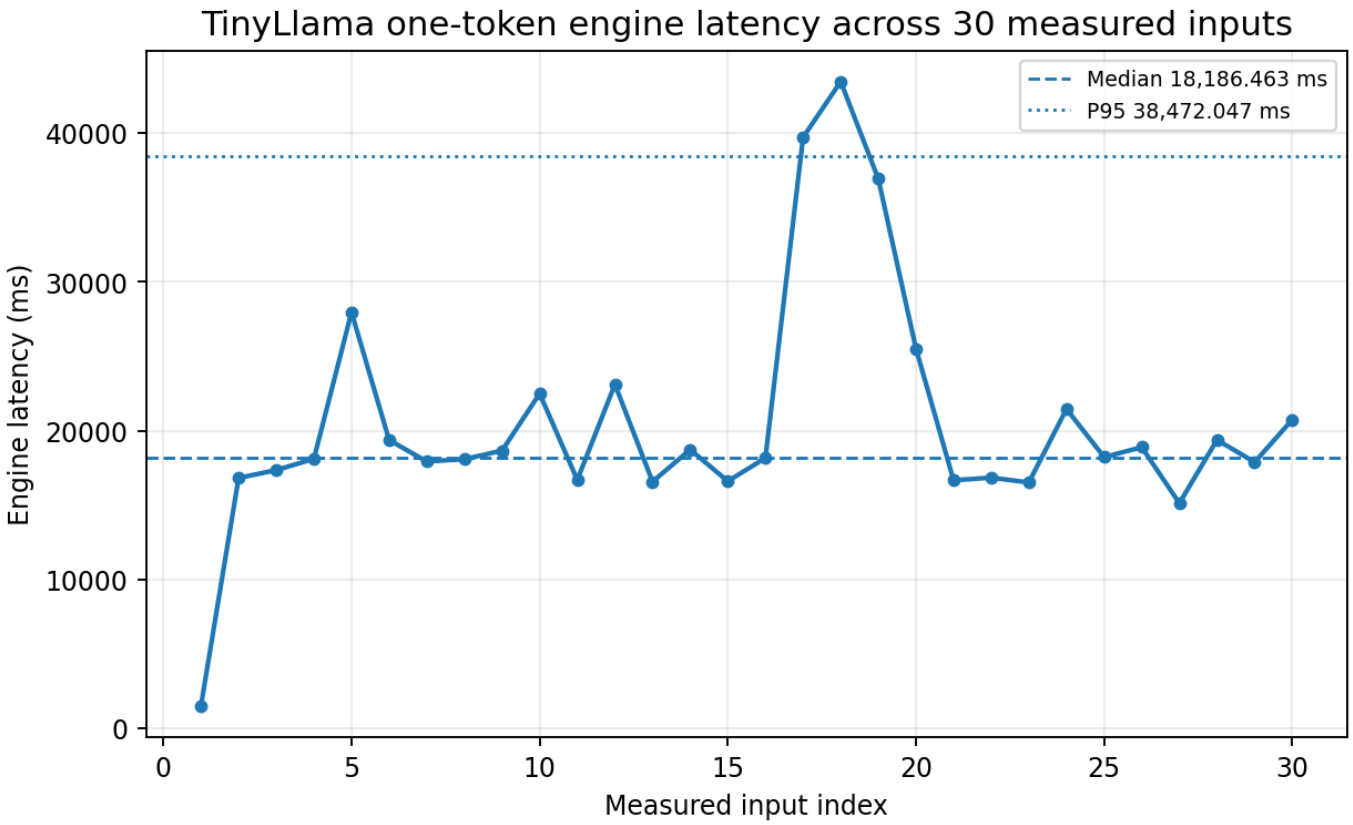

Figure 2. TinyLlama llama.cpp engine latency for all 30 measured inputs. Generation was restricted to exactly one token.

TinyLlama median engine latency was 18,186.463 ms, mean 20,511.532 ms, and P95 38,472.047 ms. The fastest measured engine time was 1,476.255 ms and the slowest was 43,480.178 ms.

Interpretation The first TinyLlama sample was much faster than the remainder of the run, but all 30 measured values are retained in the statistics. No outlier deletion is applied.

6. Runtime and Memory Comparison

Runtime

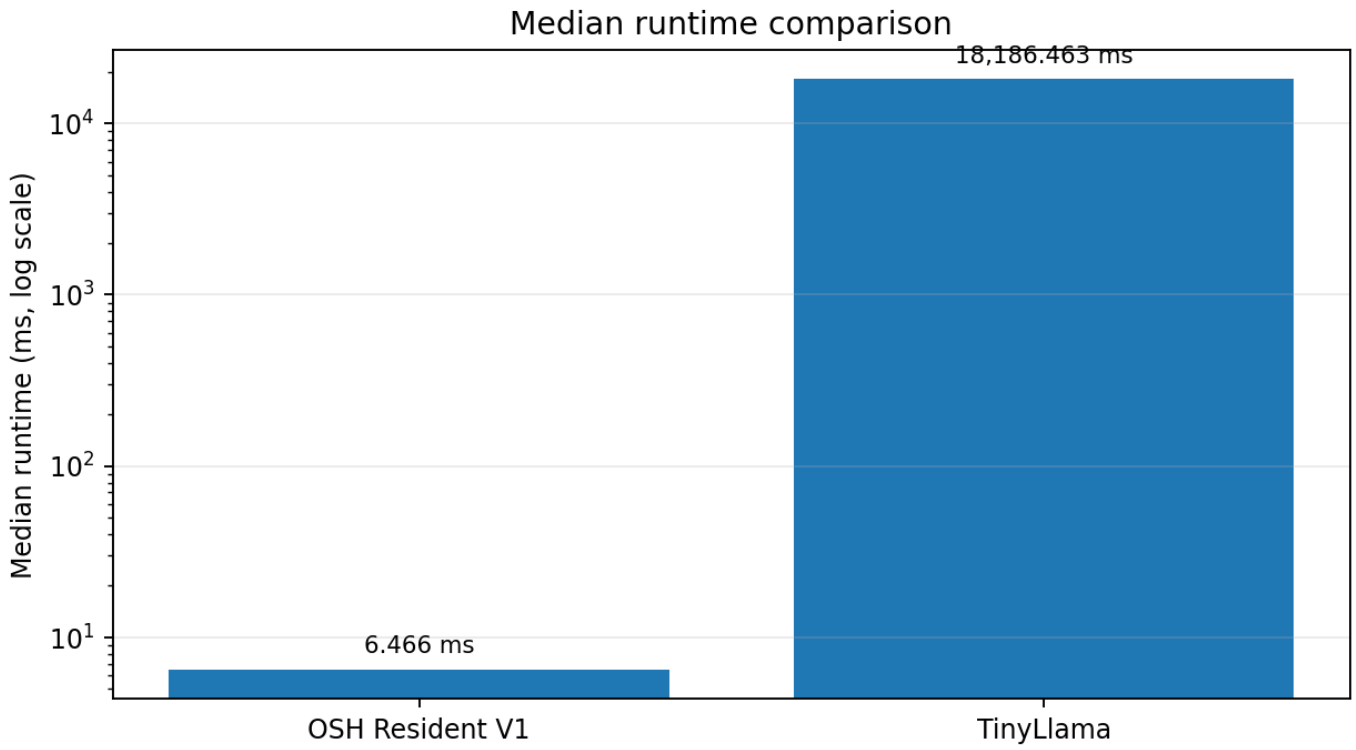

Figure 3. Median runtime on a logarithmic scale.

The median runtime ratio is 2,812.69x in favor of OSH under this benchmark definition. This is approximately 3.45 orders of magnitude.

6. Runtime and Memory Comparison

Resident memory

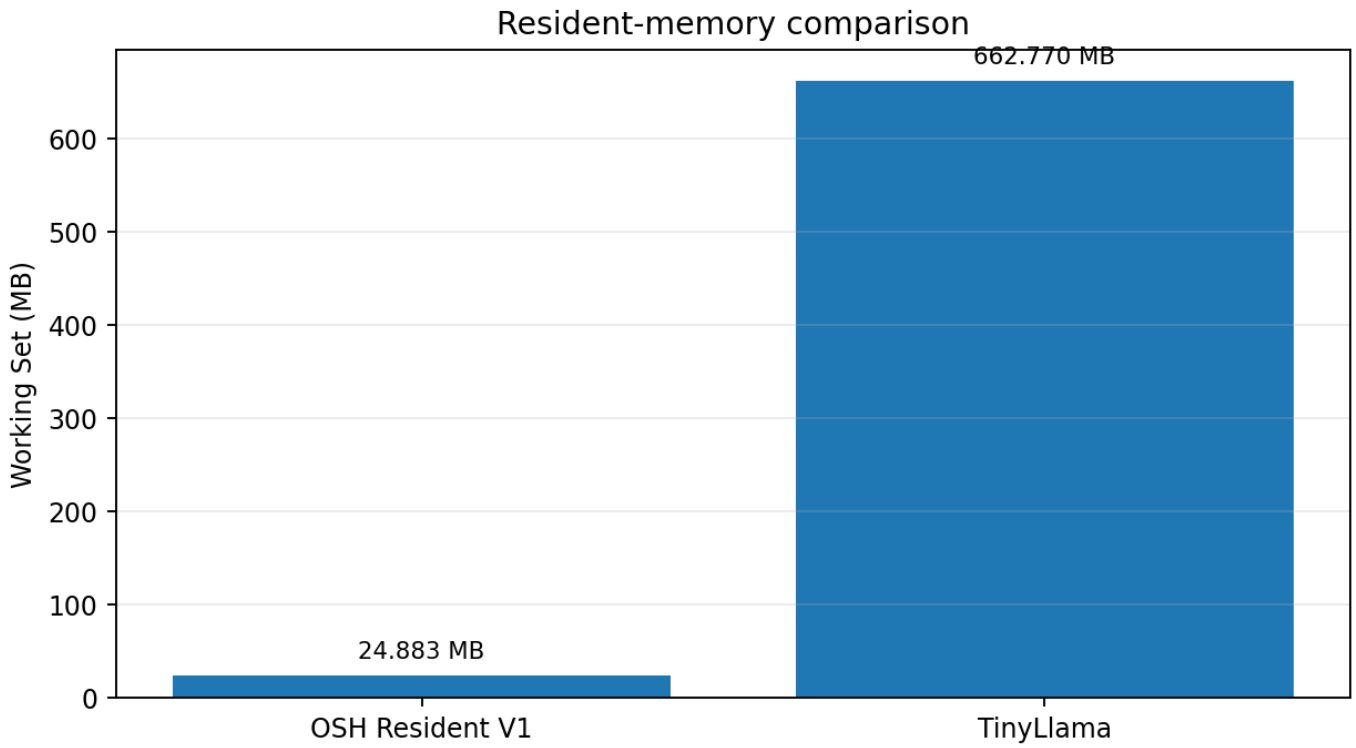

Figure 4. Resident-memory comparison.

OSH used 24.883 MB Working Set versus 662.770 MB for TinyLlama, a measured ratio of 26.64x.

7. Relation to Benchmark 2

Historical reference: Benchmark 2 - Exact Replay Technical Report, 20 August 2026. Its cover and Section 6 report an OSH median of 28.760 ms, an OSH P95 of 34.726 ms, and a TinyLlama/OSH median runtime ratio of 631.45x. These are the Benchmark 2 values used on the website.

Resident Runtime V1 remains a separate runtime-version measurement. The original Benchmark 3 technical run reported a 6.466 ms OSH median and a 2,812.69x runtime ratio; the three-run validation is recorded separately in Section 11.

Metric	B2 - Exact Replay	B3 - V1 single run
OSH median	28.760 ms	6.466 ms
OSH P95	34.726 ms	15.015 ms
Unique states	9 / 30	30 / 30
Median runtime ratio	631.45x	2,812.69x

Cross-reference correction - 12 September 2026. This page replaces the earlier B2 attribution of 26.384 ms, 49.568 ms and 691.19x. Those values are not used as the Exact Replay reference. No Benchmark 3 measurements have been changed.

Source precision. The B2 source itself also displays 28.761 ms in some summary cells. This cross-reference follows its cover, narrative summary and Section 6 (28.760 ms); the original B2 PDF remains unmodified.

Interpretation boundary. This is a comparison of defined runtime versions and protocols, not evidence of equivalent task capability or superior intelligence.

8. Structured-State Integrity

The OSH phase reported structural validation OK and 30 unique state signatures across 30 measured inputs. The state tuple changed continuously across the run, including definition, cause, function, quantity, inference, mechanism, open, presence, and state speech-act categories.

Input	Latency	Speech act
01	4.223 ms	presence
02	7.211 ms	state
05	15.819 ms	cause
10	10.855 ms	function
17	6.919 ms	quantity
18	9.939 ms	inference
20	13.226 ms	open
24	12.600 ms	mechanism
30	6.511 ms	definition

This table shows representative points from the run. The complete console output contained all 30 state tuples and all 30 latencies.

Validity signal The 30/30 unique-state result is particularly important because the speed improvement was not obtained by holding the homeostatic state constant.

9. Scientific Interpretation

What is demonstrated. On the stated Intel Atom D2500 CPU-only system, OSH Resident Runtime V1 executed the defined structured homeostatic cycle with a 6.466 ms median and 24.883 MB Working Set. The deliberately conservative TinyLlama one-token CPU baseline measured 18,186.463 ms median engine time and 662.770 MB Working Set.

What the ratios mean. Under this specific protocol, the median runtime ratio is 2,812.69x and the Working Set ratio is 26.64x.

What is not demonstrated. The benchmark does not establish superior intelligence, reasoning quality, factual accuracy, language quality, or equivalent task capability. It should not be presented as an intelligence benchmark.

Why the one-token baseline is conservative. Restricting TinyLlama to exactly one generated token minimizes Transformer generation work. The resulting comparison is therefore focused on local computational runtime rather than text-generation quality.

Recommended wording “Under the controlled CPU-only architectural-runtime benchmark, OSH Resident Runtime V1 measured 6.466 ms median cycle latency and 24.883 MB Working Set, versus 18,186.463 ms median one-token TinyLlama engine time and 662.770 MB Working Set.”

10. Conclusion

The Resident Runtime V1 benchmark records a substantial change in the measured OSH runtime regime. The structured cycle completed with a 6.466 ms median, 15.015 ms P95, and 30 distinct state signatures across 30 measured inputs. Against the one-token TinyLlama CPU baseline, the measured median runtime ratio was 2,812.69x and the Working Set ratio was 26.64x.

The result should be interpreted as a computational-efficiency measurement for the specified benchmark harness and hardware. It is strongest when presented together with the stated interpretation boundary and with Benchmark 2 retained as the historical result for the earlier OSH runtime.

Final result OSH Resident Runtime V1: 6.466 ms median | 15.015 ms P95 | 24.883 MB Working Set | 30/30 unique states | TinyLlama/OSH median runtime ratio 2,812.69x.

OSH GLOBAL SYSTEMS

Technical Benchmark Report | Resident Runtime V1

11. Three-Run Repeat Validation

A separate three-repetition run was executed with the same 30 measured inputs, 3 warmups, CPU-only execution, one-token TinyLlama baseline, and read-only OSH path. Each OSH repetition completed structural validation with 30/30 unique state signatures. The console summary reported the following medians and cross-run standard deviations.

OSH MEDIAN OF MEDIANS
3-run repeat
5.770 ms

TINYLLAMA MEDIAN OF MEDIANS
engine time, 1 token
17,799.972 ms

REPEATED-RUN RUNTIME GAP
TinyLlama / OSH
3,084.92x

3-run summary	Value
OSH median SD	0.391 ms
TinyLlama median SD	210.436 ms
OSH structural validation	OK in all 3 runs

11. Three-Run Repeat Validation

Run medians and structural validation

Run	OSH median latency	TinyLlama median engine time
Run 1	6.431 ms	17,999.064 ms
Run 2	5.770 ms	17,578.391 ms
Run 3	5.738 ms	17,799.972 ms
3-run summary	Median: 5.770 ms	Median: 17,799.972 ms

OSH structural validation	Run 1	Run 2	Run 3
Unique states	30/30	30/30	30/30

11. Three-Run Repeat Validation

Algorithmic-decade interpretation

3.49 algorithmic decades

Using the three-run median-of-medians ratio, 17,799.972 ms / 5.770 ms = 3,084.92x. If one algorithmic decade is defined as a 10x runtime separation, then log10(3,084.92) = 3.489. The repeated-run result therefore spans approximately 3.49 algorithmic decades (orders of magnitude) in runtime.

Interpretation boundary. This repeat test strengthens the reproducibility of the architectural-runtime result. It does not convert the benchmark into an intelligence, reasoning-quality, factual-accuracy, or language-quality comparison. The original single-run result (2,812.69x; 3.45 orders of magnitude) remains a separately recorded measurement.

Source: three-run CMD execution on 12 September 2026; no FARO/OSH production files modified.

OSH GLOBAL SYSTEMS
Technical Benchmark Report | Resident Runtime V1 | Three-Run Validation