Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Request Is Not Readback: Auditing a Runtime Controller's Self-Reported Telemetry on a Jetson Nano

Antonio Souto Rodriguez

antonio.rodriguez@icomp.ufam.edu.br

Federal University of Amazonas https://orcid.org/0000-0002-2107-5074

José Reginaldo Hughes Carvalho

Federal University of Amazonas https://orcid.org/0000-0002-3828-7233

Research Article

Keywords: Architectures Architectures SOC Design Automation Algorithms Simulation methods Design methodology Design methodology Co-simulation model-based testing

Posted Date: November 10th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11669160/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

Request Is Not Readback: Auditing a Runtime Controller's Self-Reported Telemetry on a Jetson Nano

Antonio Souto Rodriguez and José Reginaldo Hughes Carvalho

***Abstract*—Embedded runtime controllers decide from—and are evaluated on—self-reported signals. We contribute a claim-admission audit: each channel is specified by its requested and observed signals, a consistency predicate with an explicit tolerance, and an exclusion rule naming the claims that become inadmissible when the predicate fails (request against readback, timing composition, request against enabled state, plausibility and attribution), applied to an evolutionary controller on a Jetson Nano. On three diagnostic missions the checks refuted three channels: the reported CPU frequency stayed at 1479 MHz at every requested cap, a composed end-to-end latency violated its own definition, and the thermal supervisor denied GPU preference in 94–99% of preferring records under restrictive profiles. The readback the first check calls for located the frequency cause (unprivileged, unchecked sysfs writes); two further missions located the latency cause (capture and processing stamps on clocks drifting apart at 30 μs/s). On three final missions on one binary carrying the write/readback and clock-basis corrections (80 834 control periods) the written cap equals its policy readback in every period, while the reported selected frequency exceeds two off-table caps in 3.8% of periods; the latency channel passes composition on one clock but, with 99.9% of records stamped at dequeue, is a dequeue-to-completion latency, not sensor-to-output; and GPU denial is absent under the inactive profile, occurs once under the moderate and reaches 8 999 of 11 168 preferring periods under the aggressive, exactly at the supervisor's predicate. The audit runs one scan per check.**

***Index Terms*—Embedded systems, runtime resource management, DVFS, telemetry validation, heterogeneous SoC.**

I. Introduction

A RUNTIME resource controller on a heterogeneous system-on-chip chooses CPU frequency caps, GPU use and concurrency while workload and temperature change. To choose it must predict what a configuration will do; to be evaluated it must report what happened. Both rest on signals the platform produces about itself—a requested setting, an enabled state, a measured outcome—which are not automatically what their names suggest. A requested cap is an intent, not a clock; a composed end-to-end latency may combine parts from different clocks; a GPU preference is not the GPU being enabled. Treating such fields as the physical quantities they are named after propagates the error twice—into the decisions and into the evidence by which they are judged [1].

This letter audits the self-reported channels of one implemented controller, states what each check found, and converts each finding into an exclusion rule. The contribution is not the controller but a *claim-admission audit*: each channel is specified by its requested and observed signals, a consistency predicate over the records the controller already writes (with an alignment rule and a tolerance), and an exclusion rule naming the downstream claims that become inadmissible when the predicate fails. Readback, timing-consistency and enabled-state checks are each familiar in isolation [7]–[9]; the publications of the closely related controllers do not report them, and this letter adds their statement as an admission contract and a demonstration on a deployed SoC controller of what the contract refutes, what it locates, and what stays visible once the system passes it.

This study was financed in part by CAPES-PROEX (Finance Code 001) and by FAPEAM (POSGRAD 2024/2025). *(Corresponding author: Antonio Souto Rodriguez.)*

The authors are with the Institute of Computing, Federal University of Amazonas, Manaus, Brazil (e-mail: antonio.rodriguez@icomp.ufam.edu.br; reginaldo@icomp.ufam.edu.br; ORCID 0000-0002-2107-5074; 0000-0002-3828-7233).

II. Closely Related Work

Learning- and search-based SoC controllers share the structure audited here: MARS [2], reflective middleware on hardware; DeepSoCS [3] and SoCRATES [4], schedulers learned in the DS3 simulator; Tangram [5], coordinated control on a hardware prototype; Cohmeleon [6], coherence selection on FPGA prototypes. Table I records whether each publication reports a check of the kind defined in Section IV: a requested actuator value against an independent reading of the same quantity (C1), a consistency test on a composed timing field (C2), or the share of requested actions the runtime enabled (C3). The criterion is reported evidence, not capability: MARS exposes sensing, actuation and current-actuation-value interfaces that would support C1 and C3, but its evaluation reports no such comparison; the simulator-based schedulers cannot fail C1 or C3 by construction. We found no reported instance of the three; we do not claim the authors did not perform them. The checks descend from measurement-validity practice: reporting rules [7], independently measured DVFS transition latency [8], and reported-versus-measured processor state [9].

III. Controller, Platform and Channels

The controller runs an 800 ms period: an evolutionary search over configurations $a = (f, g, s, c)$—a CPU-cap request f, a GPU-enable indicator g, a work-split s and a concurrency level c—evaluated with a workload-specific surrogate memory, selected by generation-local Pareto comparison and submitted to a scheduler; a thermal supervisor may withdraw GPU availability independently of the search. The design records three things per period and keeps them apart: the *requested* action, the *enabled* state (what scheduler and supervisor permitted), and the *measured* outcome (throughput, processing latency, external-meter power, platform temperature), each validity-flagged; collapsing them would make two findings below undetectable.

TABLE I
CHECKS REPORTED BY CLOSELY RELATED CONTROLLERS (AS PUBLISHED) AND BY THIS WORK'S THREE FINAL MISSIONS. "–": NOT REPORTED; "N/A": SIMULATOR, REQUEST AND EFFECT COINCIDE. C4 IS NOT COMPARED.

Work (environment)	C1 cap readback	C2 clocks	C3 enabled vs. pref.
MARS [2] (SoC)	–	–	partial*
DeepSoCS [3] (sim.)	n/a	–	n/a
SoCRATES [4] (sim.)	n/a	–	n/a
Tangram [5] (proto.)	–	–	–
Cohmeleon [6] (FPGA)	–	–	–
This work (Nano)	**C1a passes**	**comp. passes**	**reported** (E3)
3 missions, 80 834 rec.	C1b fails, 3.8%†	dequeue-based‡	0/0.004/80.6% denied§

*Exposes the current actuation state through an interface; no preference-versus-enabled comparison is reported. †Readback = request in all periods; `scaling_cur_freq` above two off-table caps in 3 089 periods (Section IV-A); in the initial missions all write attempts failed and the reported frequency stayed at 1479 MHz across requested caps. ‡Capture and processing stamps were on different clocks (30 μs/s drift); corrected before the three missions (slopes -0.12, -0.13, -0.12 μs/s, zero composition failures), but 99.9% of records are stamped at dequeue, so the channel is dequeue-to-completion latency (Section IV-B). §Preferring records denied by the supervisor under the inactive, moderate and aggressive profiles: 0 of 12 077, 1 of 22 419, 8 999 of 11 168.

Platform: Jetson Nano (L4T 32.7.5); four recurring image-processing workloads in 1800-s phases, twelve per six-hour mission; three thermal profiles differing only in stage thresholds (CAUTION/WARNING/CRITICAL at 75/85/95, 55/62/70 and 45/50/55 °C, named inactive, moderate and aggressive; WARNING withdraws the GPU). Two mission sets are used and kept apart throughout. The *diagnostic* set—three missions of August–September (80 964 control periods) and two of 7 October—was recorded before the corrections of Sections IV-A and IV-B; it is the evidence for the findings. The *final* set—one mission per profile (12, 11 and 10 October) on one binary (SHA-256 `d7235178...`) with the write/readback and clock-basis corrections, 26 907, 26 976 and 26 951 control periods—is the evidence for the verification and for every number in Table I and Fig. 1. Each profile was executed once in the final set. Streams: one action record per 800-ms control period (request, enabled state, readback, measured outcome); a telemetry trace at about 500 ms whose aggregator keeps only frames with camera, algorithm, display, platform and meter components present (processed frames without a display record—375 050, 317 189 and 264 443—are dropped, so latency statistics describe displayed frames); a transition log per hot-swap; a sealed event log whose manifest carries binary and configuration hashes (no source hash: unset at build time) and a clock diagnostic; and a console log, from which a data-plane watchdog's recoveries (11, 0 and 3, each overwritten by the next period's request) are counted apart from search actions.

IV. THE AUDIT AND ITS FINDINGS

Four checks are applied—three *necessary* consistency checks, C1–C3, and one plausibility-and-attribution rule, C4, which governs what may enter scoring; a failure refutes an interpretation, a pass removes one contradiction without establishing correctness, and each failure yields an exclusion rule Ek. **C1** (request vs. readback), two predicates: **C1a**, per control period the written cap f_w (the gene after the scheduler's transforms) must equal the value read back from `scaling_max_freq` on the same clock within the period; **C1b**, the reported `scaling_cur_freq` must not exceed f_w. **C2** (composition): per trace record, $L_{\mathrm{comp}} \geq L_{\mathrm{proc}}$, $L_{\mathrm{comp}} \geq L_{\mathrm{disp}}$ and $L_{\mathrm{comp}} - L_{\mathrm{proc}} - L_{\mathrm{disp}} \geq -1$ ms; and the residual term, taken as 60-s median bins and fitted by ordinary least squares against mission time, must have a slope within ± 2 μs/s; the predicate is the binned fit, the raw-record fit is reported for information. The bound is set by what it must detect and what it must tolerate: the mixed-clock drift found below is 28 μs/s, while 2 μs/s amounts to 43 ms over six hours, below the P5–P95 width of the residual term itself (about 40 ms), so a slope inside the bound is not distinguishable from the term's own scatter. **C3** (request vs. enabled): per control period, a record with `prefer_gpu`$=1$ and `enable_gpu`$=0$ is a denial; denials are counted, never folded into a utilisation figure. **C4** (plausibility): a measurement enters scoring only with its validity flag set, and a derived quantity is reported only if every term's window membership is established. Missing fields fail the check for that record and are counted.

A. C1: the requested cap is not reflected in the reported frequency

Joining every telemetry record (about 500 000 per mission) to the most recent cap in the runtime state gives the initial picture. The reported core frequency (mean of four cores) is 1479 MHz at every requested cap, in all three missions, with a P5–P95 band of zero width over 18 301–133 539 records per cap and mission (1.30 M records; join staleness at most 0.5 s): the reading never changed within a mission, so the records are coverage, not independent samples. A cap is a maximum, so the finding is that the reported frequency *exceeds* it at every cap below 1479 MHz. Actuation delay is tested directly: in holds of at least 60 s at caps below 1479 MHz (five in ORIGINAL, fifteen in REVISED2, none in REVISED1), every record 30 s or more after the cap change—all 19 997 and 23 835 of them—still reports 1479 MHz. This rules out delays under 30 s but not a failed write, a platform mode pinning the maximum or a readback of another quantity [11]; the per-core readback at each request, once added, identified the cause: the missions had been launched as an ordinary user, every write to the root-owned `cpufreq` nodes failed, and the scheduler ignored the return value. The corrected scheduler refuses to start without write access to the policy nodes, counts failed writes, and records the readback beside the request each period. C1a passes in the three final missions: the written cap equals its readback in every one of 80 834 control periods, with zero write failures (Table I; Fig. 1(a), where the readback lies on the diagonal). The cap sat below 1479 MHz for 21%, 41% and 58% of the inactive, moderate and aggressive missions, so the frequency dimension is exercised, not only honoured. C1b does not: `scaling_cur_freq` exceeds the written cap in 222, 246 and 2 621 periods (3 089 of 80 834, 3.8%), all at the two caps the scheduler's transforms produce that are not entries of the platform's OPP table (460.8 and 1035.3 MHz, reported by `scaling_cur_freq` as the next entries, 518.4 and 1036.8 MHz; the aggressive profile uses those paths most). Since `scaling_cur_freq` may report the driver's

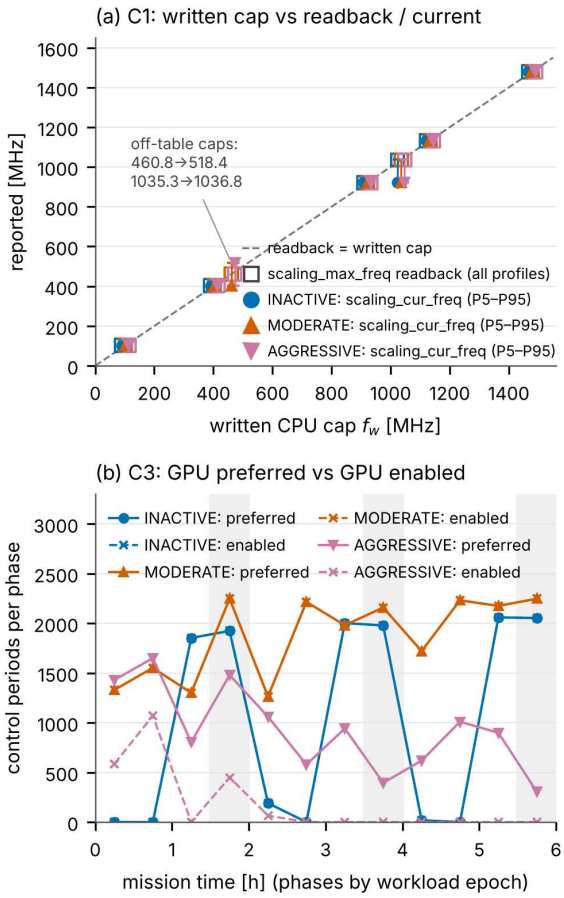


Fig. 1. The three final missions. (a) C1: for each written cap f_w, the `scaling_max_freq` readback (hollow squares, on the diagonal in every period) and the reported `scaling_cur_freq` (median, P5–P95); the two off-table caps are reported one OPP step above. (b) C3: GPU-preferred and GPU-enabled control periods per 1800-s phase, grouped by logged workload epoch (Median phases shaded): preference and enabled state coincide under inactive, differ in one moderate period, and diverge in four of five preferring periods under aggressive.

TABLE II
C2 ACROSS MISSIONS: RECORDED CLOCK DRIFT (μS/S), START OFFSET, COMPOSITION FAILURES, AND THE SLOPE OF THE RESIDUAL TERM (μS/S; 60-S MEDIAN BINS, RAW-RECORD FIT IN PARENTHESES).

Mission	Drift	Offset (ms)	Failures	Slope
Interm. inactive, 7 Oct	28.4	+7	0	29.6
Interm. moderate, 7 Oct	28.7	−161	353	29.7
Final inactive	28.5	−222	0	−0.12 (−0.11)
Final moderate	28.6	+7	0	−0.13 (−0.21)
Final aggressive	28.3	−271	0	−0.12 (0.00)

last requested state rather than a measured clock [10], C1a establishes a contract about the kernel's policy state—what was written is what the kernel holds—and the C1b failures are a policy-state mismatch, not measured over-frequency; the physical clock needs an external instrument. **E1** (initial missions): no performance or energy effect is attributed to capping; a model keyed on the request carries a request-conditioned partition, not a physical feature.

B. C2: a composed latency field violates its definition

For one output with nested boundaries, the composed channel L_{comp} (labelled end-to-end latency by the diagnostic implementation) satisfies $L_{\text{comp}} \geq L_{\text{processing}}$. Four static recordings fail this necessary relation. Later mission streams satisfy it, yet passing does not establish what the field measures: in two intermediate missions, run with the scheduler corrected but the capture stamp unchanged, the implied pre-processing term $L_{\text{comp}} - L_{\text{proc}} - L_{\text{disp}}$ grows linearly with no frame backlog, at the rate at which the run manifests record `CLOCK_MONOTONIC` drifting from `CLOCK_MONOTONIC_RAW` (Table II); subtracting the recorded, time-varying offset leaves a flat term, and the -161 ms start offset of the second mission accounts for all 353 of its composition failures. The capture stamp (from the camera driver) and the processing stamps were on different clocks. Stamping capture on the controller's clock (the driver's stamp is used only when its basis is identified at dequeue, otherwise the dequeue instant) removes the effect (Table II): the per-phase median of the term is 20–27 ms (inactive, moderate) and 21–53 ms (aggressive), and none of the 127 577 trace records violates composition. The fallback changes what "capture" denotes, so its use was audited from the run logs: in all three missions the driver stamp was identified as `CLOCK_MONOTONIC_RAW` at stream start (age 33–40 ms); within 2–13 s the age rose to 0.4–0.8 s as the first actions were applied, the stamp left the 400-ms plausibility window, and the detector fell back to the dequeue instant for the remainder: 99.94%, 99.99% and 99.96% of trace records (42 525, 42 562 and 42 444); the 25, 4 and 17 earlier records carry the converted stamp. Composition holds on both subsets; the slopes above are the fallback subset's. The channel is therefore classified as dequeue-to-completion latency; sensor-to-output latency remains unmeasured (the sensor-to-dequeue wait was 170–180 ms in the sensor-stamped records and 0.4–0.8 s in the transients) and is excluded from downstream claims. A capture stamp that survives such transients is the next correction. Passing establishes a single clock basis and bounded composition, not that every output is matched to its own input frame. **E2:** a composed latency channel is neither a prediction target nor a reported metric unless its stamps share one clock, and it is named by its measured boundaries; directly measured processing latency is retained.

C. C3: GPU preference diverges from the enabled state

Under REVISED1 and REVISED2 the search requested GPU execution while the supervisor did not permit it in 13 035 and 12 579 action records (94.2% and 98.7% of preferring records); GPU-enabled states were 61.7%, 3.0% and 0.6% of action records, so a preference share reported as utilisation would describe execution that did not occur. The final missions separate the mechanisms. Under the inactive profile (WARNING 85 °C; CPU at most 60.5 °C) the supervisor never acted and 0 of 12 077 preferring periods were denied; under the moderate profile (WARNING 62 °C; CPU at most 62.5 °C) it acted in exactly one period, 1 of 22 419; under the inactive profile the search itself requested the GPU in 0–

8% of histogram and Sobel periods and 82–92% of Gaussian and median-filter periods: the choice was the search's, not a veto. Under the aggressive profile (supervisor WARNING at 50 °C; CPU median 53.5 °C) the initial pattern returns with actuation working: 8 999 of 11 168 preferring records denied (80.6%), GPU enabled in 8.0% of records (Fig. 1(b)). The supervisor's predicate is $\max(T_{\mathrm{CPU}}, T_{\mathrm{GPU}})$ above the WARNING threshold, and denial coincides with it exactly: all 8 999 preferring periods above 50 °C by that measure were denied and none of the 2 169 at or below it; under the moderate profile the one denial is the only preferring period above 62 °C (0 of 22 418 otherwise). The divergence is the supervisor, by design. The preferring share does not decline within a phase under denial (first versus last ten minutes: 93%→64%, 100%→24%, but 19%→100%); a denied request records a CPU outcome under a GPU-keyed memory entry, an update path that can contaminate the entry and is consistent with that persistence. **E3:** preference, enabled state and measured utilisation are reported separately, never a preference share as utilisation.

D. C4: plausibility and attribution

Power comes from an external meter through the aggregator's algorithm window, temperature from platform telemetry, throughput and processing latency from the application, each validity-flagged; no power or temperature value in the final traces is missing or non-positive. Attribution was executed on the recorded streams: the meter stream carries one row per 1.85 s (each a 20 000-sample average), so 57% of the 800-ms control periods contain no meter row and the per-period power value is a held one, of median age 0.92 s (P95 1.76 s). The meter driver logged clock-anchor resets (0, 10, 6) around 0, 5 and 6 stream gaps of 3–8 s; 0, 44 and 28 trace records (about 0, 28, 18 control periods) hold a value older than 3 s. The aggregator's finer live integration is not reconstructable from what was recorded. **E4:** scoring on validity-flagged values only, stale held values excluded by age; no energy-per-output claim, since the correspondence of meter and algorithm windows cannot be established from the recorded streams.

V. Consequence: the Rules Bind the Evaluation

A companion replay evaluation of this controller's workload memory applies E1–E4 (no physical reading of the memory key, no composed latency target, preference apart from utilisation, validity-flagged power only); under its assumptions, part of a retention effect across workload returns is consistent with request-conditioned frequency partitioning.

VI. Limitations

Each profile was executed once in the final set; workload repeats within a mission are within-run evidence only. The off-table residual is governor behaviour, not a write failure (a non-table maximum is reported at the next table entry); cap transforms should be constrained to the OPP table, as the next build does. The reproducibility package (manifests, build hash, transition logs, analysis scripts) is archived at doi:10.5281/zenodo.23350634.

VII. Conclusion

On the diagnostic missions, three self-reported channels of a deployed SoC controller did not support their ordinary interpretation: the requested cap was not reflected in the reported frequency; a composed latency field violated its own definition; GPU preference was denied in most preferring records under restrictive profiles. The readback the first check called for found the cause of E1 (unprivileged, unchecked writes) and verified the policy contract in three final missions, one per profile, while leaving a 3.8% off-table mismatch visible; the same missions located E2 in a mixed clock basis and verified its correction as dequeue-to-completion latency; and they showed E3 to be the supervisor acting exactly at its predicate: absent under inactive, once under moderate, extensive under aggressive. We recommend the audit as the first step of any evaluation built on a controller's own telemetry, and that controllers log request, enabled state, readback and measured outcome as separate fields on one clock, one record per period.

Acknowledgments

ChatGPT (OpenAI) and Claude (Anthropic) were used to edit the prose of all sections, to draft the analysis and figure scripts of Section IV, and to check the references of Section II; the authors verified all measurements, analyses and conclusions, take full responsibility, and declare no conflicts of interest.

References

[1] T. Mytkowicz, A. Diwan, M. Hauswirth, and P. F. Sweeney, "Producing wrong data without doing anything obviously wrong!," in *Proc. ASPLOS*, 2009, pp. 265–276.

[2] T. Mück, B. Donyanavard, B. Maity, K. Moazzemi, and N. Dutt, "MARS: Middleware for adaptive reflective computer systems," arXiv:2107.11417, 2021.

[3] T. T. Sung *et al.*, "DeepSoCS: A neural scheduler for heterogeneous system-on-chip resource scheduling," *Electronics*, vol. 9, no. 6, p. 936, 2020.

[4] T. T. Sung and B. Ryu, "SoCRATES: System-on-chip resource adaptive scheduling using deep reinforcement learning," in *Proc. IEEE ICMLA*, 2021.

[5] R. P. Pothukuchi *et al.*, "Tangram: Integrated control of heterogeneous computers," in *Proc. 52nd IEEE/ACM Int. Symp. Microarchitecture*, 2019, pp. 384–398.

[6] J. Zuckerman, D. Giri, J. Kwon, P. Mantovani, and L. P. Carloni, "Cohmeleon: Learning-based orchestration of accelerator coherence in heterogeneous SoCs," in *Proc. 54th IEEE/ACM Int. Symp. Microarchitecture*, 2021, pp. 350–365.

[7] T. Hoefler and R. Belli, "Scientific benchmarking of parallel computing systems: Twelve ways to tell the masses when reporting performance results," in *Proc. SC*, 2015, pp. 1–12.

[8] A. Mazouz, A. Laurent, B. Pradelle, and W. Jalby, "Evaluation of CPU frequency transition latency," *Comput. Sci. Res. Dev.*, vol. 29, pp. 187–195, 2014.

[9] D. Hackenberg, R. Schöne, T. Ilsche, D. Molka, J. Schuchart, and R. Geyer, "An energy efficiency feature survey of the Intel Haswell processor," in *Proc. IEEE IPDPS Workshops*, 2015, pp. 896–904.

[10] The Linux Kernel documentation, "CPU performance scaling," Admin Guide, sec. Policy Interface in sysfs, https://docs.kernel.org/admin-guide/pm/cpufreq.html

[11] NVIDIA Corp., "Power management for Jetson Nano devices," *Tegra Linux Driver Package Development Guide*, release 32.7.5, 2024.