International Journal of Sustainability and Innovation in Engineering (IJSIE)
https://www.doi.org/10.56830/IJSIE2024

Vol.2 2024

Addressing power efficiency challenges in AI hardware through verification

Vikas Nagaraj
MTS at Advanced Micro Device(AMD), San Jose, California, USA
Author Email: vikas.jodigattenagaraj@gmail.com

Received: 1 June 2024. Accepted: 10 August 2024. Published: 10 October 2024

Abstract

AI accelerators already run with constrained energy and thermal budgets, and small inefficiencies are amplified across an entire fleet, resulting in increased costs and emissions. This work redefines power efficiency as a checkable requirement and not a back-silicon addition. It specifies power intent in IEEE 1801 (UPF), encodes protocol non-correctness with SystemVerilog/PSL assertions, and quantifies progress with power-state, transition, and cross coverage (DVFS X workload phase X thermal bin). The reproducible dataset schema integrates time with microarchitectural counters, voltage, frequency, temperature, and power, measured across real workloads (ResNet, BERT, and attention/GEMM) in simulation, emulation, and instrumented silicon. Telemetry input is synchronised using triggers and PTP/NTP; rails calibrated and error budgets quoted. The continuous integration gates are merged on quantitative thresholds (e.g., >2% p95 energy/inference regression), and dashboards auto-bisect offending changes. Experiments show a hybrid analytical-plus-ML estimator of 3.8-6.1% MAPE at millisecond latency with 30- 60x emulation throughput compared to simulation and mid-single-digit energy reductions due to verification-driven fixes. Case studies involve preventing standby leakage through restored isolation, smoothing a DVFS table to eliminate 10-15 ms oscillations, and fixing compiler schedules that caused incorrect L2 miss models and increased DRAM data traffic. This yields a realistic, start-to-finish pipeline UPF, ABV/formal, emulation/FPGA, calibrated rigs, and CI to bring watts into the top echelon of test metrics and achieve long-lasting efficiency improvements in GPU/NPU/ASIC accelerators. The full scope includes training and inference across 14-5 nm nodes adhering to rigorous safety, ethics, and licensing practices.

Keywords: *Power-aware verification, Unified Power Format (UPF), Dynamic Voltage and Frequency Scaling (DVFS), AI accelerators, Power-state coverage.*

1. Introduction

The acceleration of rendering has shifted from single-user boutique labs to multi-tenant clouds, where energy is the limiting factor. Large transformers are trained to sustain near-peak utilization over days, while the inference fleets have a continuous work pattern with bursty demand. Dennard scaling is dead; as leakage increases, the integration sees increasing energy draw without a proportionate voltage drop, so the per-operation dynamic energy drops at a slow rate, whereas the leakage energy is rising. Because of this, performance improvements are taking the form of power and cost issues rather than showing themselves as a free efficiency improvement. The activity, voltage, and frequency have to be controlled to be practically efficient. At the subsystem level, power can be written as P = 2 q C 1/2 V f 2 + P leaking. The various AI workloads increase the activity factor, alpha, due to congested MAC usage and reliance on tensor transfer. The focus of the "memory wall" is radiating: on-die SRAM and last-level caches trade so frequently; networks-on-chip flip-flop; and HBM or DDR interfaces incur large energy overheads per bit. Hotter silicon worsens efficiency; more heat raises leakage and the possibility of droop, so throttles are needed to limit throughput at higher power per operation—fleet-wide small inefficiencies resulting in small incremental increases in costs. The idle/stalls power consumed in an ungated clock domain, a DVFS governor shifting between two extremes, and retention leakage power in standby devices are all examples of wasted watts per device. Amplified by thousands of accelerators and 24/7 duty cycles, these defects increase operating cost and emissions. Hence, power should not only be elevated to a measurement, but a proven behaviour denotation, with fields examined in permanent unification.

This paper reviews how pre- and post-silicon verification can be used systematically to expose and prevent power regressions in AI accelerators. The goal is to elevate a post-check metric to a pass/fail first-order core requirement in RTL simulation, gate-level analysis, emulation, bring-up, and fleet operation. This paper delves into how to specify power intent- domains, isolation, retention, and level shifting in such a way that it can be statically discovered and dynamically proved that a particular violation occurs. It also speaks of how workload instrumentation is done on representative workloads and gathers synchronized power and performance telemetry at kHz rates without affecting the behavior. It also discusses how to transform intent, assertions, and measurements into coverage models and continuous-integration gates that prevent energy regressions, how to localize root causes across RTL, firmware, compiler-scheduling, and runtime DVFS policies.

Covered are GPUs, NPUs, and ASIC accelerators, which are built in advanced CMOS process nodes between 14-nm and 5-nm. Inference as well as training regimes are investigated, precision modes (FP16, BF16, INT8), and either batch or sequence-length ranges that challenge memory bandwidth and on-chip interconnections. The analysis considers typical digital designs with on-silicon SRAM, a last-level shared cache, network-

on-chip, and off-package HBM or DDR. Analog in-memory computing, near-threshold computing, and exotic memories are omitted since they have different failure modes and measurement requirements. Targets datacenter-class form factors, although this methodology applies to edge accelerators with less telemetry and more restrictive thermal margins.

This report adds a verification-based process that takes the energy as a specification. It first determines UPF-driven power intent and static low-power checks to guarantee complete isolation, retention, and level shifting across domains. Second, it specifies assertion-based verification of protocol sequencing, including save/restore, clock-gating enables, DVFS handshakes, workload phase, and thermal-bin measurable assertion-based coverage of power-states, transitions, and cross-coverage. Third, it explains practical instrumentation: SAIF/VCD at pre-silicon, emulation power proxies, post-silicon telemetry via sense resistors and PMBus, and on-die monitors, calibration, and synchronization. It suggests a CI gating strategy and dashboards that diagnose builds on energy regressions and surface root causes across RTL, firmware, compiler scheduling, and runtime policies.

This study is organized into various chapters. Chapter 2 discusses accelerator power budgets as well as power management, estimation, verification, and benchmarking, with a particular focus on power coverage gaps and DVFS validation. Chapter 3 provides information on data sets, data preprocessing, visual analytics, power-aware verification method, and measurement instrumentation. In chapter 4, implementation and tooling will be described: UPF/CPF authoring, assertion libraries, the checking of formal low-power and emulation/FPGA mapping, and supporting CI automation flows including coverage gates and passing and failing policies. Chapter 5 contains setups, baselines, results, sensitivity, and case studies. Chapter 6 includes the implications and threats, limitations, and future considerations. Reproducibility artifacts and conclusions are addressed in chapters 7 and 8.

2. Literature Review

2.1 AI Accelerators and Power Budget Breakdown

The new generation of AI accelerators is focused on serious multiply-accumulate throughput packed into small datacenter form factors. Dynamic power follows $P \approx \alpha \cdot C \cdot V2 \cdot f$, while leakage rises with temperature and bias; both vary with utilization and workload phase. Compute arrays, then dominate switching at high occupancy. However, on-chip memories and the network-on-chip can come close to computing in overall energy since operand movement scales the effective activity. Transformer and vision tasks accelerate memory demand: long-sequence attention, cross-token mixing, as well as large activation maps trigger bursts of SRAM and DRAM accesses that increase the activity factor α. Reported performance of large-model pipelines- including the multi-stage reasoning over visual inputs- pushes the heavy tensor flow and control variability that tests memory and

interconnect as well as arithmetic units (Singh, 2023).

Numeric format trends (Bit, INT, FP, FxP, BFP, unspecified) over time (2019-2022) reveal the influence of the precision decision on power budgets in AI accelerators. An increase in the use of rising INT and mixed-precision computing around 2021 will increase the throughput of compute arrays, but increase operand movement. These shifts increase the activity factor α, because dynamic power is proportional to $P \approx \alpha \cdot C \cdot V^2 \cdot f$. long sequences and large activation maps on transformer and vision pipelines exacerbate SRAM/DRAM bursts, putting pressure on memory and on-chip interconnect as much as the arithmetic units when being stressed.

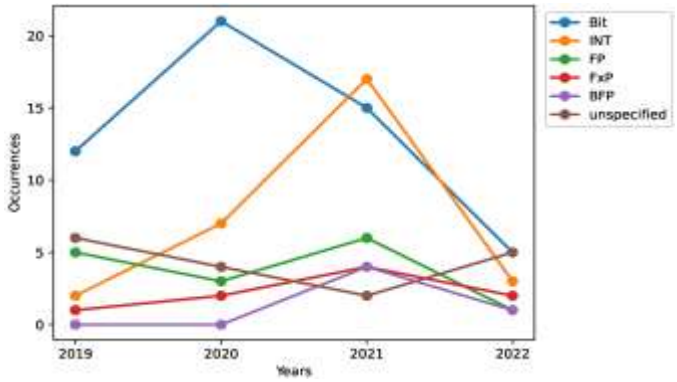

Figure 1: Evolving numeric formats impacting α and memory traffic (2019–2022)

A subsystem perspective divides the budget into compute arrays, register files, and SRAMs, cache hierarchy, NoC routers and links, I/O PHYs, and off-chip DRAM. The capacitive load per hop is increased, as well as the level-shifter overhead, where islands differ in voltages. Isolation and retention policy has a significant impact on leakage between kernels because static power is dominant at retention and low-utilization at advanced nodes. RAM references are expensive since a miss must also trigger I/O drivers, termination networks, and DRAM bank activation; reducing misses via tiling and reuse pays off in energy and latency. Dynamic external memory models, i.e., architectures that trade intermediate representations to allow subsequent comparison, explicitly demonstrate the alignment of algorithm memory demand to bandwidth demand and power consumption (Raju, 2017).

The costs of dataflow placement aggravate or diffuse these costs. Putting compute as close to large SRAM macros minimises NoC traversal energy at the expense of congestion and timing closure. At the operator level, the redundant read and write operations are prevented, and instead, producers and consumers are scheduled back-to-back. Carefully designed, structured, and unstructured sparsity can save energy α by omitting useless operations; at times, an overly aggressive sparsity can actually increase index bandwidth and index control flow to move energy off arrays and into the interconnect. The benefits of collective communication at the cluster time scale include training in parallel, which saves an additional layer of power expense by overlapping with compute and gradient

compressions, thereby shaving off time and joules.

2.2 Power Management Techniques

Power management is an effective control in the physical, microarchitecture, firmware, and compiler/runtime. α can be minimized by power gating or clock gating when idle pipelines still consume power, because toggles are suppressed. However, latency is incurred during wake-up, and state-retention costs are incurred. Retention and isolation techniques characterise safe interchange between power states in a legal domain, whereas level shifters assure electrical integrity between domains. Dynamic voltage and frequency scaling (DVFS) presents a strategy on how to trade off performance in terms of quadratic energy sensitivity to voltage, limited by timing margin and droop tolerance [19]. Governors should not swing with thermal and workload changes; popular stabilizers include guard hysteresis, minimum-residency timers, and slew-rate limits.

Discipline is important, as the mechanisms of operationalizing are as important. In software security, the addition of automated gates into the CI/CD will prevent sign-offs prior to deployment (Konneru, 2021). Existing flows also make low-power design inflexible. Presubmit and nocturnal travels can solve ceilings on idle power in specified states, minimum duration in deep-sleep, maximum transition delay (on↔retention↔off), as well as energy-per-inference constrained on standard workloads. Firmware and compiler revisions are also tested against the power-savings pass-fail threshold in addition to functional tests. In practice, teams connect rail-telemetry parsers to the continuous tests, normalize the data by inlet temperature, and store traces of DVFS states. Failures produce artifacts (plots, logs, diffs) that are relayed to the change owner immediately so that they may be remediated.

2.3 Power Modeling & Estimation

Well-calculated, high-speed estimation supports design-space exploration and gating CI. Analytical macro models take unit-level capacitances and empirical activity factors to allocate energy per operation and interconnect hop [22]. They get rapid early bound because of switching activity by simulation (SAIF/VCD) or proxy counters in emulation. Gate-level simulation has higher fidelity but cannot simulate hours of traces that characterize training; economic flows integrate RTL/emulation work with reduced unit-level fidelity models and a modest set of post-silicon anchors.

Dependable telemetry and synchronization are central to calibration. Board sensors, available to the rail, reveal the voltages and currents of the rails; on-die monitors relay temperature and drooping conditions; system software provides the timestamp of the kernel. Such sources are required to align with per-phase energy attribution, as well as to split the compute/memory/I/O contributions. Historical examples of applications of telematics in engineering practice can be based on the fact that to establish trust relying on

large-scale telemetry, it is required to provide a high-quality timestamping across numerous devices of various heterogeneity, tamper-proof logging, and noise-aware filtering (Nyati, 2018). In accelerators, multi-rate fusion is used to time-align fast current probes with slower thermal sensors, and the software logs; Outlier rejection is used to manage spikes in measurements during rail transits.

The estimators should not violate physics and workload structure. The monotonicity with respect to voltage and near-quadratic sensitivity to V should be imposed or penalized during fitting; the models should saturate at full compute and memory bandwidth above the bottleneck. The case of attention layers requires the estimator to switch regimes between compute-bound at short sequences and bandwidth-bound at long sequences; that of convolution requires capturing the tile shape and its reuse of DRAM and buffer accesses (Singh, 2023). A separation of validation by workload family and silicon SKU can expose overfitting. Reports include MAPE of instantaneous power, energy per task, confidence intervals, and runtime overheads, such that teams can balance fidelity against throughput in CI.

2.4 Verification Methodologies for Power

Power checking layers. The power intent and protocol correctness are used on top of functional checks to ensure that regressions never get to silicon. Unified Power Format records domains, supply sets, retention /iso strategies, level shifters, and a legal power-state table. Static analysis identifies a lack of shifters, unlawful moves, coverage holes in retention, and inconsistencies at the level. Dynamic, power-aware simulation understands UPF semantics such that toggling and X-propagation are correct with respect to the domain state; assertion-based verification specifies legal entry/exit sequences, save/restore ordering, bus-quiesce, and DVFS handshake. States, transitions, and crosses to be quantified (e.g., DVFSxworkload phase thermal bin).

As shown in the figure below, a UPF-aware low-power verification flow integrates power intent with RTL: a single compiler that supports RTL and UPF, IEEE 1801 standards-based validation, and a shared simulation framework that is used in both functional and low-power simulations. Static, dynamic, and CDC checks verify conditions on domains, isolation, retention, and legal transitions of power states, whereas assertion-based protocols gate save/restore, quiesce, and DVFS handshakes. Scalable coverage is compatible with UCDB/VIQ, a connected debugger can accelerate triage, and data-driven closure combines functional and power data in a single database where regression signoff is continuous.

Figure 2: An example of UPF-aware power verification flow

The approach is practical due to its process and automation. A triage artifact power-state table is fed by a scoreboard with waveform bookmarks to its failing cycle and hierarchy paths to isolation or retention primitives. FPGA and emulation platforms enable the emulation of hours-long traces that are more representative of production behavior and inject rare events like droop-triggered throttling or clock-stop recovery. The concept of CI/CD, with its emphasis on policy gates, automated collection of evidence, and immediate feedback, translates well to low-power verification, shoring up repeatability and accountability [12]. In practice, teams connect power-aware tests that run along with functional suites, mandate some threshold level of power-state coverage, and reject merges that exceed some percentage energy increase at fixed throughput.

2.5 Benchmarks, Datasets & Gaps

The processes of measuring should reflect the activities of production instead of playing with kernels. CNNs and transformers train on representative batch and sequence tensor arrays and memory hierarchies; training involves optimizer, checkpointing, and all-reduce steps, which impose significant tensor array and memory demands. In addition to throughput and TOPS/W, diagnostic metrics include energy per inference, energy delay product, tail-latency energy at the 95th percentile, and power-state transition latency. The schemas of dataset tuples are the synchronized time-performance, performance-voltage/frequency, performance-temperature, and measured-power; the labels should identify power states and DVFS transitions to make the targeted evaluations supervised. The big-model pipelines in the field of visual question answering indicate a realistic compound step which include feature representations, cross-modal fusion, and response approximation that are fair to put through as benchmark scenarios as far as they challenge diverse levels of the hierarchy : dataset quality and repeatability [25].

Telemetry management practices appropriate in the fleet-scale monitoring (identity, integrity, and time coherence) enhance the quality and repeatability of datasets [18]. Although the progress is evident, there exist lacunae. There is no commonly agreed concept of power coverage, similar to functional coverage, and without state, transition, and cross

coverage targets over realistic workloads, sign-off remains arbitrary. DVFS validation is fragile: clean ambient tables cannot be used with thermal constraints and droop detectors, and either oscillate or throttle excessively at the expense of energy. Co-verification across layers is still limited across crossings between compiler scheduling, power management at runtime, and RTL protocols, even as neural architectures using a strict external memory access or dynamic control can trigger especially active boundaries [21]. Formalizing coverage models, fixing DVFS governors with guard conditions, and normalizing open, telemetry-complete power sets enable organizations to post formal measurements of progress. They would make power a form of first-class and verifiable requirement.

3. Methods and Techniques

The section will explain the data, processing, visualization, power-aware verification, and instrumentation employed to solve the power efficiency in AI hardware. The task is to develop a reproducible workflow that reveals squandering activity, quantifies advancements, and blocks degradations via coverage-based tests [15]. Procedures are defined in pre-silicon as well as post-silicon contexts to allow transfer of insights wire-to-chip.

3.1 Description of Data Set

The analysis makes use of a power-trace dataset in a synchronized form (t, perf counters, V, *f*, T, P). Here, t is an increasing monotonic timestamp; perf counters are counts of microarchitectural activity, including array utilization, kernel identifiers, L2/L3 miss rates, memory-controller queue depths, DRAM bandwidth, tensor-core occupancy and stall types; V and *f* are instantaneous rail voltage and clock frequency; T is junction temperature; and P is instantaneous power in watts. Analog rails and power are sampled at either 1-10 kHz, depending on DAQ constraints [6]. Digital counters and DVFS state changes are latched at kernel boundaries and on interrupts. Each sample is marked with a capture ID and a software build hash to enable traceability.

Table 1: Schema and sampling overview of the power-trace dataset

Field(s)	Definition / Values	Units / Encoding	Sampling / Trigger
t	Monotonic timestamp	s (float64)	1–10 kHz
perf_counters	Array utilization, kernel ID, L2/L3 misses, MC queue depth, DRAM BW, tensor-core occupancy, stall types	int64 / categorical	Latched at kernel boundaries and interrupts

Field(s)	Definition / Values	Units / Encoding	Sampling / Trigger
V, f	Instantaneous rail voltage and clock frequency; DVFS step IDs	V, Hz / enum	1–10 kHz for V; on DVFS events for f
T	Junction temperature (for leakage/derating compensation)	°C	1–10 kHz (sensor-limited)
P	Instantaneous power; integrates to energy	W	1–10 kHz
Workloads & Knobs	ResNet-50/101, BERT-base/large; LLaMA-style kernels; GEMM sweeps; knobs: batch size, sequence length, heads, sparsity, quantization (FP16/BF16/INT8), data layout	model/op names; numeric/categorical	Scripted harness per run with phase markers (warm-up/steady/cooldown)
Labels	Bench energy (∫P dt), windowed power (10–50 ms), power state {ON, OFF, RETENTION, ISOLATION}, DVFS transitions with timestamps, workload-phase tag	J, W, enums	Derived post-capture; events on state changes
Storage & Provenance	Columnar Parquet with units and calibration; manifest (SKU, board rev, firmware, ambient); capture ID + software build hash; QC gates (monotonic t, expected state changes, acceptable telemetry loss)	files + JSON/YAML	Per capture at ingest

The workloads are chosen to overload memory- and compute-bound. It also provides end-to-end inference and training passes of ResNet-50/101 and BERT-base/large alongside layer-level kernels of LLaMA-style decoder blocks (self-attention, MLP, softmax). Microbenchmarks traverse GEMM shapes on batch size, sequence length, head count, and structured sparsity. Each workload has several harnesses that vary batch size, sequence length, activation sparsity ratio, quantization mode (FP16/BF16/INT8), and data layout to explore utilization/traffic phase space. Explicit markers of warm-up, steady-state, and cooldown periods have been included in the harness so that thermal transients can be isolated and regime behavior assessed.

Labels include bench-level energies as joules (integral of P over time), windowed power in watts (10-50 ms moving windows), and power-state labels based on design power intent. The labels used by State are ON, OFF, RETENTION, and ISOLATION; the DVFS steps are encoded as discrete state IDs per rail and a change time stamp [23]. A workload-phase tag (data load, warm-up, steady-state, cooldown) is assigned to each record in order to facilitate stratified analysis. Traces are written in columnar format (Parquet) with noted units and per-channel calibration information, and a manifest is addressed to the device SKU, board revision, firmware version, and ambient conditions. Quality checks turn away those captures with non-monotonic timebases or the lack of state changes when known kernel boundaries are passed, or excessive packet loss occurred on telemetry buses.

3.2 Data Pre-processing

Multi-rate data (produced by heterogeneous sensors) needs to be synchronized. Clock-domain reconciliation via hardware timebases is performed when present; otherwise, the host timestamps are time-corrected based on the estimation of offset and frequency of device-emitted heartbeats through the timebeating device driver. All raile signals are brought to a shared target rate via band-limited interpolation of analog signal rails and zero-order hold of categorical states. De-noising is performed using a Savitzky-Golay filter; the width of the window is used such that both DVFS edges (tens of milliseconds) and high-frequency noise imposed by the DAQ can be suppressed. Filtering is used on P and V but not on state indicators, so the filter prevents the smearing of discrete events.

Table 2: Data pre-processing workflow and key QC checks

Step	Method	Signals	Example settings	Output / Check
Time sync & alignment	Hardware timebase; else heartbeat-based offset/drift estimation	All streams	Drift < 50 ppm; max offset corrected per capture	Common timeline with aligned timestam

Step	Method	Signals	Example settings	Output / Check
				ps
Resample & preserve states	Band-limited interpolation (analog); zero-order hold (categorical) to a shared rate	P, V; DVFS/state	Target 2 kHz within 1–10 kHz envelope	Uniform samples; DVFS edges not smeared
De-noise (analog only)	Savitzky–Golay filter	P, V	Window 31–51 samples; polyorder 2–3	DAQ noise reduced; DVFS steps intact
Outliers & sanity checks	Hampel on dP/dt; cross-sensor concordance (PMBus vs shunt)	P, V	3σ threshold; reject >5% rail mismatch	Spikes removed; bad packets dropped
Temp comp. & splits/QC	Fit Pleak(T)=AekTP_\text{leak}(T)=Ae^{kT}Pleak(T)=AekT on idle bins; compute Pdyn=P−PleakP_\text{dyn}=P-P_\text{leak}Pdyn=P−Pleak; stratified splits; validity gates	All rails; labels	Preserve DVFS/temperature bins; time skew ≤1 ms/min; required PST transitions present	Leakage isolated; fair splits; captures pass Q

Outliers may be identified through Hampel filtering of the first derivative of the power and checks of cross-sensor consistency (such as PMBus power and shunt-based measurement). Temperature compensation isolates the dynamic and leakage components. Leakage is represented on each rail as $P_{leak}(T)=A\,e^{kT}$, where idle segments are binned by temperature; dynamic leakages are then $P_{dyn}=P-P_{leak}(T)$. Toggle-density proxies of VCD/SAIF are extracted by normalizing counts of toggling modules to their clock edges and summing the switching counts into subsystem activity factors. Additional capabilities include

cache/TLB miss rates, DRAM bandwidth, SM/PE utilization, instruction issue/retire rates, stall breakdown (memory, dependency, pipeline), and droop-detector events. SKU densities have standardized all characteristics with sturdy scalers to deal with heavy tails.

To avoid label leakage in downstream analysis, train/validation/test splits are made by workload family and silicon SKU only, and not randomly across time. Each split leaves the distribution of DVFS states and the number of temperature bins unchanged, allowing for generalization claims to be made [30]. A quality gate ensures that no deadband occurs on sensor saturation that the timebase skew is within the maximum values allowable. Those required PST transitions are present during directed tests.

3.3 Data Exploration using Visual Analytics

Visual analytics facilitates the localization of defects and the generation of hypotheses. Temporal heatmaps present power by kernel over time, to indicate cyclic throttling, idle-with-clocks-on interruptions, or memory-controller pulsing. A Sankey diagram maps a total energy investment using activity-weighted power models into subsystems of the compute array, SRAM/L2, interconnect, DRAM, and peripheral controllers to reveal outliers in energy consumption with respect to each workload category. The plotting of Pareto fronts in a throughput versus energy range allows engineering choices to be described as movement along the efficiency curve (or throughput accuracy) or improving the efficiency curve. P andP_{dyn} per DVFS state plots of the overlapping and instability of the governors can be visualized using violin plots.

As shown in the figure below, visual analytics is used to combine data, models, and interactive visuality to turn measurements into knowledge that drives power efficiency. Temporal heatmaps present power per kernel as a function of time, showing throttling, power-meter asleep-with-clocks-on periods, and memory-controller pulsing. A Sankey diagram shows energy allocation, through activity-weighted models, across compute arrays, SRAM/L2, interconnect, DRAM, and peripheral controllers, and highlights individual subsystem outliers by workload category. Pareto plots can be used to plot throughput against energy to compare substitute design tradeoffs. Violin plots of P and P_{dyn} over DVFS states show overlap and governor instability [7].

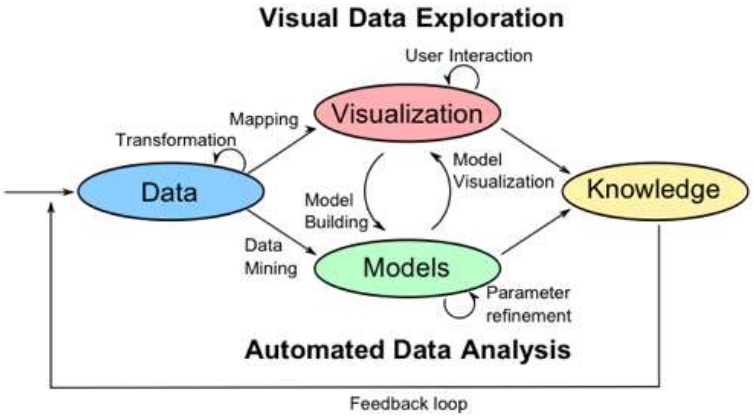

Figure 3: Visual analytics workflow linking data, models, visualization to power knowledge

Such views are also hard-coded into a notebook dashboard with SKU, workload, and thermal bin filters. Analysts mark anti-patterns, such as red spinning clocks during a kernel crunch, L2 thrash increasing DRAM traffic, or DVFS oscillating across a threshold. Each flag stores a probe timespan, a snapshot of evidence (waveform cross-section, counters, assertion traces), and is elevated to a verification ticket that generates directed tests and UPF/ABV refinement.

3.4 Verification Methodology (Power-Aware)

Power-aware verification links an explicit power intent to an executable checker that ties design states, software protocols, and physical rails. The power intent is written in the 1801 Unified Power Format (UPF) and quantifies domains, supply sets, isolation and retention strategies, and level shifters. A golden Power-State Table (PST) lists legal states and legal transitions between those states, including constraints on which sets of domains may be turned ON, OFF, or in RETENTION/ISOLATION concurrently, as well as sequencing constraints related to allowed transitions between individual states and between groups of states. Based on this specification, the following protocols are identified: save/restore, isolation enable, clock-gating, and voltage/frequency handshakes. The dissimilarity to crisply define boundaries applies well to UPF because, in the distributed software systems, context boundaries are used to restrict unwanted cross-talk, which would entail defining interfaces and responsibilities precisely [2].

Assertions can be programmed in SystemVerilog Assertions (SVA) or PSL and attached to RTL, such that they persist through the elaboration and low-power insertion steps. Checks made are retention save/restore ordering, isolation timing order, clock-gating enables, bus-quiesce before power-down, and DVFS request/ acknowledge handshakes. Unauthorized transgression of the domain is caught with messages of varying gravity. Each assertion has payload fields--domain IDs, rail names, and timestamps to facilitate triage. Firing assertions adds markers to waveform data files and host records files, providing the ability to cross-probe to power and activity traces. Coverage is gathered along three axes:

state coverage (all states observed on at least one workload), transition coverage (all transitions observed), cross coverage over DVFS and workload phase, and × thermal bin. Thresholds (≥95% transition coverage and ≥80% cross-bin coverage) are imposed in CI; they fail presubmit or nightly jobs and automatically add the offending traces.

3.5 Measurement & Instrumentation

Pre-silicon measurement includes simulation, emulation, and FPGA prototyping. TL and gate-level simulations generate SAIF dumps to estimate power consumption by toggling; to manage runtime, activity is sampled at the module level in representative windows delimited by kernel markers instead of the entire test. Emulation offers a pairing of power proxies such that internal activity counters and DVFS state taps are made visible to host software [31]. Waveform capture is event-based: assertion hits because ring buffers to spill pre-/post-windows around the failure, keeping storage versus time bounded. Examples of shunt-based telemetry being exported to FPGA prototypes include shunts on significant rails; logic exports revealed counters for array usage, cache misses, and interconnect flits, to be consistent with the perf-counter model.

Post-silicon rigs include sense resistors on the board and precision differential amplifiers per-supply rail with high-sample-rate data acquisition. Digital telemetry is interrogated at the fastest safe rates on PMBus/SMBus monitors and on-die ring oscillator, droop detectors, and thermal diode monitors. Each channel is compensated with a programmable reference load; thermal drift models are also fit per rail to ensure long-run experiments are consistent. Ground-truth power P is calculated as a summation over rails of V x I, but with synchronized samples; energy is cumulatively integrated by using trapezoidal rules and compared, where possible, with board-level coulomb counters. The telemetry and synchronization design are based on the principles of the scalable communications system, which is also able to provide the throughput and data quality under high-rate sampling and multi-device aggregation [24].

Synchronization completes the loop between the software and the hardware perspectives. Memorable trigger lines are used on the hardware boundaries of a kernel and on DVFS updates; these pins are buffered out to the DAQ as stern anchors. The Network Time Protocol only works in coarse alignment. However, Precision Time Protocol with hardware timestamping offers a larger reduction in skew, to less than 100µs between the hosts and embedded controllers [9]. Alignment algorithms are responsible for cross-correlating between the trigger streams and reducing the offsets to the minimum remnant error value. Final verification replays marks over waveforms and traces to verify that assertion timestamps, DVFS events, and power edges occur at the same time within known tolerances.

4. Power-Aware Verification Implementation & CI Integration

4.1 UPF/CPF Authoring Patterns for AI Accelerators

A power intent reflects the architectural decomposition of the accelerator and is combined with a robust low-power implementation. A typical topology divides the chip into compute arrays (tensor cores or MAC tiles), SRAM scratchpads and register files, the NoC or PCIe complex, and the DDR or LPDDR PHY with I/O pads [5]. Individual supply sets, power switches, and retention strategies are dedicated to each domain to allow flexible termination of functional islands without losing configuration information necessary to restart. Segmentation: The isolation cells lock ex-inter-domain signals to protocol-safe values as a source goes dark, and level shifters ensure integrity when signals are passed between voltage islands.

A golden Power State Table (PST) lists legal combinations of the states of the domains and DVFS points, and prescribes explicitly reset, retention, and wake sources. Static checks determine that all types of crossings have been covered, whether the isolation/retention insertion is correct, and there are no uncontrolled always-on paths into dead logic. UPF/RTL version teams run relevant pieces through a structural diff in review and sign off requirements, including but not limited to: no unisolated crossings and all PST rows have defined entry/exit sequences.

4.2 Assertion-Based Power Sequencing (SVA/PSL)

Power protocols are encoded as temporal properties that enforce save→isolate→clock-gate→switch-off on entry and switch-on→clock-ungate→de-isolate→restore on exit. Sequencing Assertions parametrize windows on regulator slew, PLL lock, and firmware latencies whilst still firing within timeouts. Guards will see that quiescent acknowledgments precede clock stops and that outward-bound retention saves finish before power switches open. There are explicit overflow and error paths: retention-save failure will cause rollback to safe state with clocks on; wake request during sequence will cause controlled sequence rollback before exit. Monitoring detects domain crossings on illegal signals, clock gating asserted during active transactions, and wakeup latency violations, which would cause software SLA breaches [20]. Every assertion stores a waveform bookmark and an error code, allowing CI triage to select the cycle and handshake in which the assertion failed.

4.3 Power-State Coverage Engineering

Coverage translates qualitative purpose into quantitative completion. State coverage asserts that each PST row is exercised with at least one realistic workload. Transition coverage includes a check of all arcs-ON-RET, RET-OFF, and OFF-ON and DVFS transitions with

interrupts, stalls, and noC traffic present. Cross coverage links environment and workload by binning across DVFS state, kernel type (GEMM, attention, softmax), and thermal bin, revealing that, say, RET during heavy DRAM traffic at hot temperature was not ever observed.

As shown in the figure below, a power-state table (PST) coverage schematic lists four states and their event-driven transitions, and exercises the transitions between states with arrows. State coverage counts to ensure that each state is covered by at least one realistic workload and transition coverage counts arcs, including DVFS-related moves under interrupts, stalls, and NoC traffic. The green callouts show the incoming and outgoing transitions per state; they sum to a coverage score of six, indicating quantified completeness rather than qualitative intent of the power protocol.

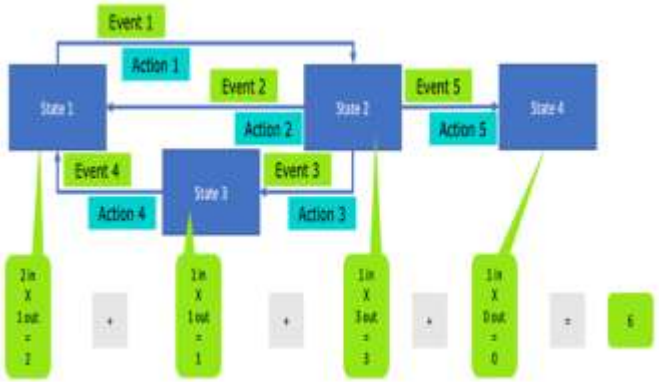

Figure 4: Power-state and transition coverage across PST states and arcs

Targeted stimulus generation is then based on hole analysis: directed test lists, traffic generators, and firmware knobs produce the special conditions needed to seal the bin. This field resembles other fields with rubric-based feedback loops, which have demonstrated better improvement when the process is structured and measurable than when it is ad hoc [11]. Coverage dashboards show trending numbers of bin covers over time and highlight regressions when a covered path is driven to zero because of design changes.

4.4 DVFS & Power-Integrity Validation in CI

Validation is not limited to functionality ordering but also includes electrical safety and energy efficiency, which have to be automated at the end-to-end level. DVFS tests use P-states swept over the workload phase to ensure governors do not oscillate, satisfy dwell times, and maintain throughput quality of service. Frequency changes at quiescent boundaries are checked by guards, or that switching under load has a compensating voltage headroom [8]. Voltage-droop resilience reflects the level of introducing bursty high-di/dt kernels and bus storms; droop monitors must be capable of detecting within guard bands and generating frequency fold-back without false trips.

To exploit grid weak points, IR-drop stress vectors are temporarily switched to open wide datapaths and memory interfaces in parallel, and thermal throttling situations test that the temperature sensors and derate tables prevent runaway. Regression budgets (e.g.,>2% energy increase at p95 load fails) and auto-bisect on a commit range to isolate the offenders are enforced by CI gates. Telemetry and power-trace differences are run through analytics pipelines to identify abnormalities in a statistically-based rather than a manually-reviewed manner, which is in line with the Best Practices offered by DevOps, which highly value continuous measurement and automated feedback (Kumar, 2019).

4.5 Emulation/FPGA and Hardware-in-the-Loop (HIL)

Pre-silicon acceleration becomes a necessity in long-running environment-dependent scenarios. Emulation specifies what power-control nets should be connected to which emulated switches and records emulated switch-activity counters as power proxies at MHz-class frequencies, allowing day-scale duty-cycle tests. Compilers that bundle the assertions into emulators pinpoint the protocol violations on demand, whereas individual trigger-gated trace collection can only capture a limited number of bits.

Fine-grained emulation can be complemented by FPGA prototypes where precision shunt sensors are attached to key rails (VDD_CORE and VDD_IO), sampled using differential ADCs, and calibrated to laboratory sources. In hardware-in-the-loop, hardware such as real voltage regulators, heaters, and programmable load equipment surround the board, and allows brown-out injection, thermal sweeps, and endurance soaking. Measurements gathered on these rigs flow back to the same CI dashboards as the simulation, providing a location to compare measured and predicted power between revisions.

5. Experiments and Results

5.1 Experimental Setup

An evaluation of three different types of accelerator representative of production deployments was performed, which were a 7 nm GPU-class device (400-450 W TDP) with HBM2e memory, a 5 nm NPU-class ASIC (250-320 W) with on-package SRAM tiles, and a 16 nm FPGA development board (75-120 W) used to conduct the power-proxy experiment—monitored VDD_core, SRAM/L2, HBM/DDR, and AUX/PCIe. Synthesized experiments were performed at RTL and gate level with SAIF/VCD toggle traces back-annotated to a power estimator. Emulation on a datacenter emulator with a power-proxy counter (domain-level power-toggle, clock enable, and DVFS state taps). Post-silicon measurements used instrumented boards consisting of four-wire 0.51 m, 1 m 1 0 shunts /rail, low noise differential amplifiers, and 16-bit ADC modules sampled at 50100 kS/s. Software counters were slaves to rail telemetry via a shared trigger line and an NTP-disciplined clock.

International Journal of Sustainability and Innovation in Engineering
(IJSIE)
https://www.doi.org/10.56830/IJSIE2024
Vol.2 2024

Table 3: Key data pre-processing parameters and QC thresholds

Metric / Setting	Statistic	Value / Range	Notes
Time drift after correction	Max residual drift	< 50 ppm	From heartbeat-based offset/drift estimation when no hardware timebase
Timebase skew gate	Limit	≤ 1 ms per minute	Capture rejected if exceeded
Resampling rate	Target	2 kHz (within 1–10 kHz sensors)	Band-limited interpolation (analog P,V); zero-order hold (categorical/DVFS)
Savitzky–Golay filter	Window / order	31–51 samples; polyorder 2–3	De-noise P and V while preserving tens-of-ms DVFS edges
Outlier rejection	Hampel threshold	3σ on dP/dt	Plus cross-sensor check: reject if PMBus vs shunt mismatch > 5%
Leakage compensation	Model	Pleak(T)=AekTP_{\text{leak}}(T)=A e^{kT}Pleak(T)=AekT	Fit on idle bins; compute Pdyn=P−Pleak(T)P_{\text{dyn}}=P-P_{\text{leak}}(T)Pdyn=P−Pleak(T)
Feature scaling	Robust scaler	median/IQR	Handles heavy-tailed distributions across SKUs
Dataset splits	Scheme	By workload family & SKU	Preserve DVFS-state and temperature-bin distributions; avoid label leakage
Validity gates	Required events	PST transitions present	Reject captures with sensor saturation deadbands or non-

Metric / Setting	Statistic	Value / Range	Notes
			monotonic timebases

The conv/GEMM microbenchmarks and ResNet-50, transformer encoder-decoder stacks with sequence lengths 64-2048 and 1-4096 sequences were used in determining workloads. Vendor profilers reported counters-processing-element occupancy, SM/compute-array occupancy, cache/TLB statistics, DRAM bandwidth, and stall reasons exported to kernel time boundaries. Continuous integration that ran nightly and presubmit suites: simulation jobs produced SAIF under exemplar stimuli; emulation captured multi-hour traces (genuine PowerPC binary and MIPS); on-rig smoke tests verified DVFS schedules and low-power sequences.

The two-stage calibration was used to acquire ground-truth information. To start with, bench supplies were used to sweep known loads, and sensors were temperature stabilized, resulting in linear calibration curves [1]. Second, an accuracy source introduced a small AC as a perturbation to the measurement of bandwidth and phase delay. The board losses at a Kelvin configuration were deducted from the rail readings. Budgets of uncertainty included shunt tolerance, amplifier gain error, ADC quantization, synchronization jitter, and thermal drift, and were combined to yield a standard uncertainty below 1.8% power and 2.2% energy over 10s.

5.2 Baselines & Ablation Design

Four estimator baselines were contrasted—an analytical-only approach employed macro-cell capacitances and SAIF computed activity factors to compute the subsystem power as $P=\alpha CV2f+P_{leak}$. A model that uses only microarchitectural countermeasures is a gradient-boosted decision tree on microarchitectural counters, DVFS state, thermal bins, and workload descriptors. A hybrid residual model imparted a learned correction to the deterministic covidence and constrained monotonicity in V and f through soft constraints [29]. A proxy-only model also lacked explicit awareness of power intentions but used emulation counters. To alleviate labeling burden, unlabeled emulation windows were utilized in pre-training feature encoders that were subsequently fine-tuned on labeled power samplings; the rationale follows established benefits of self-supervised representation learning to enhance downstream performance in nearby tasks subsequently [26].

Verification ablations contrasted “no-UPF checks” (functional-only) against “full power-aware ABV and formal,” which included assertions for isolation/retention sequencing, clock-gating enable timing, and DVFS handshakes, plus power-state, transition, and cross coverage over DVFS × workload-phase × thermal-bin. Variants of the DVFS governor

were examined: a fixed "performance" table, an on-demand heuristic, which responded to utilization and temperature, and a silicon-characterization-synthesised table with guardbands. Fixes to each found issue were themselves added behind flags, such that before/after A/B comparisons could be done within the same commit as the change. This isolates the effect of the change driven by verification.

5.3 Main Findings

The hybrid residual estimator demonstrated the best accuracy/latency trade-off across hardware classes. Median MAPE on per-kernel power windows was 3.8-6.1 per cent with the hybrid model, 5.2-9.4 per cent with ML-only, 8.7-12.6 per cent with analytical-only, and 11-15 per cent with proxy-only. Estimator latency was less than 2 ms on-rig, which is acceptably small to allow test-time gating without a throughput impact. Emulation provided a 30-60X speedup over simulation in terms of throughput (executions per day) without sacrificing domain-level power ordering to support overnight execution of lengthy sequences and temperature bins that simulation could not support before.

Table 4: Quantitative summary: estimator accuracy, latency, speedups, coverage effects

Metric	Statistic	Value / Range	Impact
Estimator accuracy (hybrid residual)	Median MAPE (per-kernel windows)	**3.8–6.1%**	Best accuracy–latency trade-off
Estimator accuracy (ML-only)	Median MAPE	**5.2–9.4%**	Lower accuracy than hybrid
Estimator accuracy (analytical-only)	Median MAPE	**8.7–12.6%**	Misses nonlinear effects
Estimator accuracy (proxy-only)	Median MAPE	**11–15%**	Least accurate baseline
Estimator latency	On-rig end-to-end	**< 2 ms**	Safe for test-time CI gating

Metric	Statistic	Value / Range	Impact
Emulation vs. simulation	Throughput gain	**30–60×**	Enables multi-hour sequences, thermal bins overnight
Verification fixes	Leakage/idle/DVFS	**SRAM standby leakage removed; idle draw down by several watts; DVFS oscillation eliminated**	Stabilized rail current; reduced tail-latency energy
Energy per inference	Change after fixes	**Mid-single-digit % reduction**	Savings at iso-throughput
Coverage correlation	Transition coverage	**<80% → late on-rig bug discovery**	Insufficient early detection
Cross/overall coverage	DVFS×phase×thermal; ROC	**95% cross → earlier detection; >90% overall → higher TPR at similar FPR**	Quantitative coverage predicts verification efficacy

Fixes resulting from verification influence identified power savings. Isolation on the uncovered debug paths also removed standby leakage on the SRAM rail, as well as several watts of system idle power draw. Tightening clock-gating can eliminate idle-with-clocks-on plateaus seen in waveforms, resulting in a reduction of the energy per inference by mid-single-digit percentages at iso-throughput. Fixing a DVFS table hole eliminated governor oscillations that had been generating consecutive voltage steps and transient droops; the patch stabilized rail current and lowered tail-latency energy.

The amount of coverage correlated with the detection of defects. Reporting lower than 80% transition coverage led to some late power-bug discovery during on-rig validation; increasing the cross coverage of DVFS workload-phase and thermal-bin to 95% detected most issues earlier during simulation or emulation. A receiver-operating characteristic constructed through the labeling of runs as defect present or no defect indicated that coverage levels greater than 90% produced higher rates of true positives at numbers of false positives reasonably similar to lower coverage levels, indicating that quantitative

coverage measures can be predictive of verification performance in power.

5.4 Sensitivity & Scalability

Sensitivity sweeps were performed to test the batch, sequence length, and temperature. Energy per inference reduced sublinearly with batch size as host transfer and launch costs were spread across more inferences; considerably high-size batches increased HBM rail energy due to congestion and more extended duration occupancy, marginally expanding residuals of the estimators. Length of sequence was most influential to the attention-centric models, wherein KV-cache traffic saturated memory rails; the hybrid model error increased by about 1.2% absolute with the most extended sequence, but less than 7.5% MAPE. The temperature influenced leakage and DVFS guardbands; stratified training over thermal bins allowed a reduction of bias to below 0.6% under hot conditions. Under programmable steps in the load on VDD_core, DVFS transition ordering assertions fired as desired, and post-fix runs indicated that transitions that violate the ordering were removed.

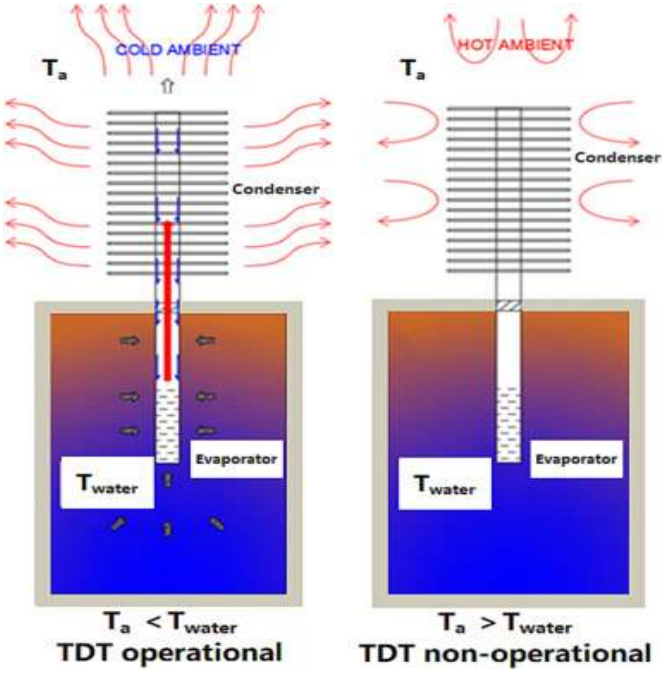

Figure 5: Thermal bins: cold operational, hot non-operational; temperature impacts behavior

Scalability was computed and solved organically. Emulation positions were distributed by workload and thermal partition; on-rig tests were executed in parallel among the boards that shared telemetry storage. This CI pipeline followed a microservices-like decomposition into specific services (build, simulation, emulation, and lab control) and autoscaling and budget caps to control the costs relative to the needs [3]. Wall-clock penalties were low: the simulations required ~10% more clock cycles with power-intent assertions in use, emulation a few percentage points more with counter collection enabled, and on-rig telemetry negligible because of the DMA logging.

5.5 Error Analysis & Case Studies

Case 1 (Absence of isolation => leakage). A low-power island that monitors infrequently accessed debug logic lacked isolation on a path that was seldom traversed on an interrupt [16]. Sleeping firmware asserted during an active debug session led to the illegal power-off state that formal checks could access. On hardware, standby current on the SRAM rail was increased by ~150 mA. Waveforms indicated that retention save was applied, but that isolation enable was two cycles late. Isolation cell as well as save→isolate ordering assert were added to eliminate the path, which puts the standby power to the baseline.

Case 2 (DVFS table hole → oscillation). A mid-frequency entry that lacked a validated voltage point at a higher temperature led to bouncing at adjacent steps of the governor during transients. Alternating current power traces resembled periodic 10-15 ms oscillations of core-rail current, and independent IPC jitter in counters also occurred. A claim on monotonic voltage frequency did not work in emulation. Oscillations vanished and rail current stabilized under stress when re-characterizing the table, widening guardbands, and adding a rate limiter to the governor.

Case 3 (Compiler schedule → L2 misses). A scheduler optimization that maximized the overlapping of memory-bound kernels exceeded L2 capacity and wasted DRAM bandwidth. Hot spots were transferred to HBM rails; dominant predictors of excess power identified by attribution were rank L2 miss rate and DRAM utilization [10]. An ABV rule warning was issued during host staging on a clock-gated-false condition when the compute-unit utilization is less than 5% and the time is greater than 1 millisecond. Metadata-based phased prefetching and tighter clock-gating on host waits eliminated L2 thrash and returned energy to baseline.

6. Discussion

6.1 Key Findings in Context

Power-aware verification advances energy efficiency as a non-binding goal to an auditable design requirement. Early declaration of power intent, domains, supply sets, isolation, and retention help eliminate illegal power crossings as well as maintain architectural state during gating and wake-up [14]. Verifying power protocols as assertion-based verification facilitates unrestrictive translation from intent to executable specification: save→isolate→gate ordering is proved correct on every transition; clock-gating enables are shown to quiesce interfaces; and DVFS handshakes are guaranteed not to violate minimum residency to avoid oscillation and rail stress. Coverage measures complete the circle by quantifying how well the design has exercised states, transitions, and critical crosses (e.g., DVFS level x workload phase x thermal bin).

When such verification telemetry is connected to continuous integration (CI) with explicit

gating, e.g., fail the build when the p95 energy/inference regresses by more than 2% at iso-throughput, energy can become a critical test criterion before silicon rather than a post-silicon shock. The main conclusion, consequently, is that faults that pathologically overestimate watts (idle clocks running, retention save miss, isolation mis-sequence, DVFS tab instability, and compiler locality loss) may be detected sooner, more cost-effectively, and without ambiguity of ownership. Likewise, operations principles, such as codifying constraints, instrument telemetry, and gate against key performance indicators, are known to increase efficiency and predictability in algorithm-driven algorithmic dispatch systems and constitute an organizational analogy to the described engineering controls [17].

6.2 Implications for Design & Flow

The attainment of these profits needs disciplined plumbing, ownership, and automation. Power intent cannot be left out of version control. Power checks must be run with every commit, including a UPF diff-lint step that emits readable differences on removed isolation, retention, or level-shifting. Assertion suites are to be sustained at the two levels: (1) the unit benches that are used to cause edge-case handshakes with the help of formal assistance to cover exhaustive corners, and (2) workload-accurate system bench that reproduces typical kernel traces in a perturbed clock, rails, interrupts, and thermal throttling signals. Power semantics implies taps to DVFS state, clock enables, retention events, and power-good indicators should be made available in emulation and FPGA prototypes; selective waves can be captured on assertion failures to reduce debug latency.

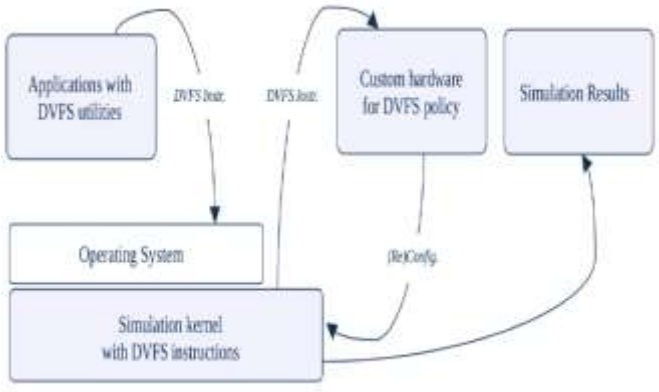

Figure 6: End-to-end DVFS control path: apps, OS, simulation kernel, hardware

As shown in the figure above, a DVFS-aware flow interconnects applications and OS utilities with a simulation kernel containing DVFS instructions, and custom hardware that implements the DVFS policy; the loop communicates simulation results back to update the policy. This is analogous to well-disciplined verification plumbing UPF diff-lint on every commit, and unit and system assertions packages, and emulation/FPGA taps offering insight into DVFS state, clock-enable, retention, and power-good, and selective-use

waveform capture on assertion failure, fast debug and continuous, version-managed, power checks across diverse accelerators prototypes in a consistent way.

Coverage dashboards and power budgets need to be made per workload class, and coverage should have deterministic pass/fail policies (block merge if any tracked scenario exceeds the energy baseline +1.5 standard deviations). Triage pipelines need to append failing claims, minimal repro traces, and involved commits to minimize mean time to root cause [13]. Ownership is necessarily cross-functional because RTL engineers own clock-gating and isolation completeness, verification engineers own assertions and coverage closure, physical design owns retention checks and level-shifter placement, firmware owns DVFS stability and quiesce sequencing, compiler and runtimes own locality and concurrency and burstiness, validation engineering owns telemetry calibration and lab-to-field correlation. Institutionalizing these roles should be supported through change control and training.

6.3 Threats to Validity

Internal risks of validity are weighted towards measurement noise, label drift, and thermal coupling. Noise includes that of ADC quantization, offset and temperature drift, switching-supply ripple, and inductive pickup on shunt wiring; a variety of countermeasures, including oversampling with synchronized timestamps, conservative digital filtering with documented passbands, shielded routing and Kelvin sensing, and periodic calibration against traceable precision loads, can be deployed. Drifts in labeling occur when kernel mix or governor heuristics change between firmware, drivers, or compilers after datasets are captured; as a counterfoot, dataset versioning based on build hashes, nightly re-baselining, or A/B gating against a frozen control image are helpful mitigation strategies.

Thermal coupling is a problem that confounds causal inference, where the temperature increases leakage and throttling; pre-heating to target bins, isothermal fixtures, heat-soak sequences, and analyzing temperature using temperature as a covariate are possible mitigations. External validity depends on the similarity of workloads and the variety of SKUs [27]. The workload matrix must encompass memory-bound, compute-bound, and communication-intensive periods, and the distribution of a sequence and batches will be synchronized with fleet telemetry. Hardware diversity should comprise process offsets, package thermal impedances, rail partitioning, and droop detector levels. Construct validity is dependent on metrics: TOPS/W cannot reveal any burst failure; energy per inference, energy-delay product (EDP) with p50/p95 bands provide meaner signals. Classification of statistical procedures, such as bootstrap confidence intervals, effect sizes, and nonparametric tests, should be standard in CI reports.

6.4 Limitations

Several limitations are enclosed generally. The low-power operation at the threshold increases the variability of delays. It degrades the existence of fixed windows in sequencing

assertions, providing more false positives, or nondeterministic timing at the expense of proof strength [28]. The need to simulate analog behavior in mixed-signal blocks and in-memory compute introduces an analog behavior that is hard to formalize in digital power intent; the coverage of these areas may depend on behavioral monitors and co-simulation without formal guarantees. Emu emulation fidelity is necessarily finite; proxy activity mapped to, but not equated with, the silicon watts, thus post-silicon correlation campaigns are essential. Telemetry accuracy on production boards can be gross or multiplexed across rails and will need to be deconvolved, and can furthermore add uncertainty. Specific optimisations (e.g., deeper power gating) might optimise against latency, wake-up energy, or silicon area to achieve the same steady-state savings, and product-level decision-making might be necessary.

6.5 Future Considerations

Standardization would increase the effects and comparability. The required states, legal transitions, and mandatory cross axes, which are workload phase, DVFS level, thermal bin, and rail configuration, should be defined in a power coverage vendor-neutral schema, and each metric should be associated with a confidence band and minimum required levels to sign off. Co-verification between the layers ought to become a standard procedure: compilers mark kernels with their estimated locality/bandwidth; firmware asserts quiesce points, residency; RTL enforces order, isolation; and CI combines traces between layers and reports violations with synchronized evidence. ASIC DVFS synthesis. That is an exciting horizon to pursue: learn power (frequency, throughput) surfaces per rail, compile stable governors with formal liveness and hysteresis properties, and automatically re-tune them in response to silicon aging or cooling environment. Production fleet power telemetry can be securely aggregated and identify drift, regressions, and hotspots that can be fed back into the design and firmware [4]. Conformance with up-and-coming audit standards (such as MLPerf Power) and green-compute reporting offers the potential to make energy enhancements transparent to customers and regulators by rendering verification evidence externally verifiable.

7. Reproducibility & Artifacts

7.1 Code, Datasets, Configs

The project needs to be released as a monorepo with a good structure /rtl/, /upf/, /sva/, /emulation/, /post_silicon/, /analysis/, /docs/. All experiments should be linked to an irreversible Git commit code and a container image, identified by its complete digest using sha-256. Dependency determinism is provided by a Software Bill of Materials (SBOM) and a lockfile (e.g., poetry.lock, requirements.txt, package-lock.json). There is only minimal reproduction via make repro EXP=exp_power_cov that will run a single script analysis/run_pipeline.py with an experiment YAML that specifies the path to datasets,

seeds, and gating thresholds.

Dataset versions are content-addressed stored, and include a dataset_manifest.json that looks like: time index, performance counters, voltage, frequency, temperature, and power. Configuration files preserve UPF levels, sets of assertions activated, coverage objectives, and DVFS tables. Waveforms, SAIF/VCD summaries, coverage, and calibrated power traces as artifacts are exported to /artifacts/commit/ and indexed by results.csv to enable meta-analysis. To protect privacy and safety, logs have any proprietary identifiers scrubbed; there is a redaction map and unit test to ensure that this is completed.

7.2 Exact Commands & Environment

Reproduction starts with the container execution, pin compilers, EDA tool shims, and Python: podman run --rm --privileged --network=host --userns=keep-id -v $PWD:/ws -w /ws ghcr.io/org/ai-power: sha256-<digest> make repro EXP=exp_power_cov. With host replication and no containers, expterts would have a pinned environment, i.e., conda env create -f env.yml && conda activate ai-power and then pip install -r requirements.txt. Hardware and OS parameters are recorded: Linux distribution and kernel, microcode revision, CPU governor, GPU/NPU driver and firmware versions, board part numbers, and rail identifiers. The /docs/board_setup.md shows boards with jumper positions, shunt, sense amplifier gains, ADC sampling rates, and synchronization methods (PTP/NTP).

The code flow, as shown in the diagram below, where a user-space application travels through the Linux networking stack to the PRUSS Ethernet and PRUSS core drivers (remoteproc), a PHY driver (MDIO PHY) on the ARM host, which communicates with the PRU firmware running on the ICSS and, ultimately, the EVM/custom board hardware. This resembles the environment pinning applied to experiments: containerized or conda-pinned toolchains, known kernel and driver revisions, and values of board-up parameters: jumper positions, shunt values, amplifier gains, ADC sampling rates, and PTP/NTP timestamps.

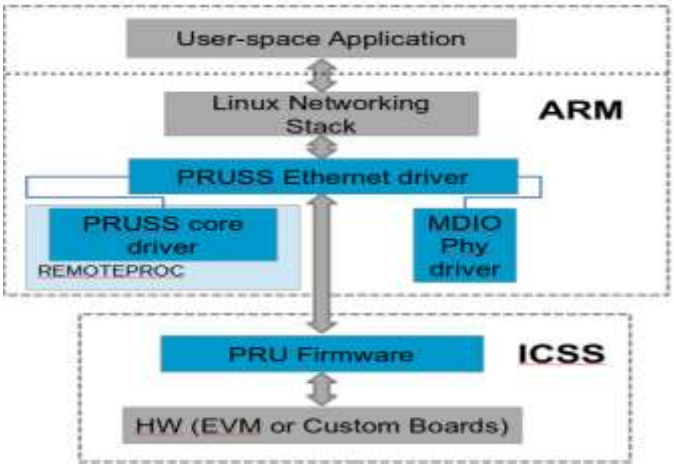

Figure 7: Reproducible environment: Linux drivers, PRU firmware, and board setup

Rail calibration: Rail calibration is done following the process outlined in/docs/ in/docs/rail_calibration.xlsx: measure zero-offset and scale with a precision load, record the ambient and heatsink temperatures, and verify linearity across operating currents. Exact simulator, emulator, and profiler invocations are scripted (e.g., scripts/run_emulation.sh --trace-window 120s --dvfs-profile dvfs_v3.yaml). Seeds are maintained in one source of truth (seeds.yaml) and injected in all stages to eliminate nondeterminism. CI duplicates via a pinned workflow artifact attachment and checksum publishing.

7.3 Licensing & Ethics

Contributions are governed by a modified version of the Apache-2.0 license to facilitate integration as widely as possible, with protections on contributors; documents are distributed under a Creative Commons BY 4.0 license to allow publication subject to attribution; datasets are covered by the Open Data Commons by permissive license to support publication of factual material. Third-party IP (RTL, EDA tool outputs, or vendor firmware) is not supported and is instead replaced with stubs and synthetic traces if redistribution is prohibited. Power-data provenance is stored in a provenance.yaml per run, taking operator, lab location, and equipment. Models/serials, calibration dates, and environmental conditions.

Safety measures are compulsory: de-energize and rewire, observe creepage/ clearance on high-current conducting rails, use differentially rated probes or appropriately rated shunts, or provide thermal management to ensure damage does not occur to the device. There are ethical aspects of handling, such as deletion of metadata containing information that can identify a customer, compliance with facility SOPs, and retention policies to erase raw traces once verified aggregates are known. Reproducibility packages are vetted aperiodically; any divergence results in an erratum in /docs/CHANGELOG.md.

8. Conclusions

This work shifts power efficiency in AI accelerators as a verifiable, first-class design requirement instead of an a posteriori metric. It formalizes power intent in IEEE 1801 (UPF), asserting domain, supply set, isolation, retention, and level shifting information, and translating this intent into executable temporal properties that enforce save/restore ordering, clock-gating versus right, bus quiescence, and DVFS handshakes. The quantitative coverage for power is established in terms of state, transition, and cross (DVFS x workload phase x thermal bin), where qualitative goals translate into quantitative completions. Estimation and gating rely on a reproducible dataset schema (t, performance counters, V, f, T, P) and a representative workload mix (ResNet, BERT, attention, and GEMM microbenchmarks).

The measurement stack includes SAIF/VCD activity in simulation, activity proxies in emulation, and post-silicon telemetry with shunt resistors, PMBus/SMBus rails, and on-die monitors, all coordinated using trigger lines and PTP/NTP synchronization. Continuous integration gates are based on energy regressions (i.e., action when p95 energy/inference exceeds a small threshold at iso-throughput) and on coverage objectives that must be high in terms of transitions and cross coverage before sign-off. Case studies feature actionable workarounds: added isolation to prevent standby leakage, improved clock-gating to eliminate idle-with-clocks-on plateaus, and fixed DVFS tables, eliminating oscillations, with verification evidence linked to waveforms and counters to support quick triage.

Cover power intent as code: version, check, and diff-lint UPF in addition to RTL. Maintain a living library of SVA/PSL assertions that encode legal sequences (save→isolate→gate→off; on→ungate→de-isolate→restore), DVFS request/acknowledge, reset ordering, and bus-idle interlocks; run them in unit benches, workload-accurate system benches, and emulation/FPGA to expose hours-long behaviors. Set engineering coverage targets early--strategize to achieve >=95% transition coverage and meaningful DVFSxphasexthermal crosses-- and write focused stimulus to cover holes. Set up deterministic experiments: pin seeds, containers, record board SKUs and rail identifiers, calibrate shunts and amplifiers, and compensate temperature to differentiate leakage and dynamic power. Self-synchronize telemetry with hardware triggers and PTP, export artifacts (waveforms, calibrated traces, coverage) per commit, gates merges on quantitatively-budgeted thresholds (e.g., >2% system energy regression at p95 load fails). RVFS governors with dwell times, hysteresis, and slew-rate limits; throttle on complainants with droop and thermo; and auto-bisect misbehaving complaints with bookmarked failure asserts and trace diffs.

Power-sensitive verification enables an industry-standard auditable language that can correlate design intent with quantifiable energy results and eliminate wasted watts across fleets. Short-term solutions include normalizing a cross-vendor, cross-tools consistent power coverage and artifact report schema, releasing pattern libraries of reference UPF designs of AI compute, SRAM, NoC, and DRAM islands, as well as encapsulating DVFS characterization as user-protected, self-tuning governors that adapt to silicon aging and changing cooling conditions. Cross-layer co-verification would align locality-changing annotations in compilers with coordinated evidence of such violations occurring at runtime as part of the quiesce signals in RTL power protocols. Fleet telemetry fed back into verification will not only detect drift but also recalibrate estimators and gradually refine CI gates; scale to chiplet fabrics and constrained edge form factors without loss of safety and data-handling discipline in the lab. Efficiency gains that result from these steps are sustainable, subject to audit, transferable, and cost-effective economically on an international scale.

References

[1] *Amin, S. U., Shahbaz, M. A., Jawed, S. A., Khan, F., Junaid, M., Kaleem, D., ... & Naveed. (2022). Temperature and humidity controlled test bench for temperature sensor characterization. Journal of Electronic Testing, 38(4), 453-461.*

[2] *Chavan, A. (2022). Importance of identifying and establishing context boundaries while migrating from monolith to microservices. Journal of Engineering and Applied Sciences Technology, 4, E168. http://doi.org/10.47363/JEAST/2022(4)E168*

[3] *Chavan, A. (2023). Managing scalability and cost in microservices architecture: Balancing infinite scalability with financial constraints. Journal of Artificial Intelligence & Cloud Computing, 2, E264. http://doi.org/10.47363/JAICC/2023(2)E264*

[4] *Farahpoor, M., Esparza, O., & Soriano, M. (2023). Comprehensive IoT-driven fleet management system for industrial vehicles. IEEE access.*

[5] *Gimbitskii, A. (2022). Interconnect design for the edge computing system-on-chip (Doctoral dissertation, MA thesis. Tampere university, 2022. URL: https://urn. fi/URN: NBN: fi: tuni-202206035477).*

[6] *Glowinski, S., Pecolt, S., Błażejewski, A., & Sobieraj, M. (2023). Design of a Low-Cost Measurement Module for the Acquisition of Analogue Voltage Signals. Electronics, 12(3), 610.*

[7] *Hebbar, R., & Milenković, A. (2022). PMU-events-driven DVFS techniques for improving energy efficiency of modern processors. ACM Transactions on Modeling and Performance Evaluation of Computing Systems, 7(1), 1-31.*

[8] *Ibba, P., Crepaldi, M., Cantarella, G., Zini, G., Barcellona, A., Rivola, M., ... & Lugli, P. (2021). Design and validation of a portable AD5933–based impedance analyzer for smart agriculture. IEEE Access, 9, 63656-63675.*

[9] *Jiménez López, M. (2019). Distributed control systems based on high accurate timing synchronization (sistemas de control distribuido basado en sincronización temporal de alta precisión).*

[10] *Jung, J., & Erez, M. (2023, October). Predicting future-system reliability with a component-level dram fault model. In Proceedings of the 56th Annual IEEE/ACM International Symposium on Microarchitecture (pp. 944-956).*

[11] *Karwa, K. (2023). AI-powered career coaching: Evaluating feedback tools for design students. Indian Journal of Economics & Business. https://www.ashwinanokha.com/ijeb-v22-4-2023.php*

[12] *Konneru, N. M. K. (2021). Integrating security into CI/CD pipelines: A DevSecOps approach with SAST, DAST, and SCA tools. International Journal of Science and Research Archive. Retrieved from https://ijsra.net/content/role-notification-scheduling-improving-patient*

[13] *Koyuncu, A., Liu, K., Bissyandé, T. F., Kim, D., Monperrus, M., Klein, J., & Le Traon, Y. (2019, August). iFixR: Bug report driven program repair. In Proceedings of the 2019 27th ACM joint meeting on european software engineering conference and symposium on the foundations of software engineering (pp. 314-325).*

[14] *Lewis, T. G. (2019). Critical infrastructure protection in homeland security: defending a networked nation. John Wiley & Sons.*

[15] *Ma, K. (2023). Improving Genetic Diagnostics and Developing Gene Therapies in Rare Muscle Diseases. Yale University.*

[16] *Mayton, B. D. (2020). Sensor networks for experience and ecology (Doctoral dissertation, Massachusetts Institute of Technology).*

[17] *Nyati, S. (2018). Revolutionizing LTL carrier operations: A comprehensive analysis of an algorithm-driven pickup and delivery dispatching solution. International Journal of Science and Research (IJSR), 7(2), 1659-1666. Retrieved from https://www.ijsr.net/getabstract.php?paperid=SR24203183637*

[18] *Nyati, S. (2018). Transforming telematics in fleet management: Innovations in asset tracking, efficiency, and communication. International Journal of Science and Research (IJSR), 7(10), 1804-1810. Retrieved from https://www.ijsr.net/getabstract.php?paperid=SR24203184230*

[19] *Papadimitriou, G., & Gizopoulos, D. (2022). Challenges on unveiling voltage margins from the node to the datacentre level. In Computing at the EDGE: New Challenges for Service Provision (pp. 13-49). Cham: Springer International Publishing.*

[20] *Qazi, F. (2020). Automating SLA enforcement in the cloud computing (Doctoral dissertation, University of Warwick).*

[21] *Raju, R. K. (2017). Dynamic memory inference network for natural language inference. International Journal of Science and Research (IJSR), 6(2). https://www.ijsr.net/archive/v6i2/SR24926091431.pdf*

[22] *Randall, D. S. (2020). Cost-Driven Integration Architectures for Multi-Die Silicon Systems (Doctoral dissertation, University of California, Santa Barbara).*

[23] *Samriya, J. K., Tiwari, R., Cheng, X., Singh, R. K., Shankar, A., & Kumar, M. (2022). Network intrusion detection using ACO-DNN model with DVFS based energy optimization in cloud framework. Sustainable Computing: Informatics and Systems, 35, 100746.*

[24] *Sardana, J. (2022). Scalable systems for healthcare communication: A design perspective. International Journal of Science and Research Archive. https://doi.org/10.30574/ijsra.2022.7.2.0253*

[25] *Singh, V. (2023). Enhancing object detection with self-supervised learning: Improving object detection algorithms using unlabeled data through self-supervised*

techniques. International Journal of Advanced Engineering and Technology. https://romanpub.com/resources/Vol%205%20%2C%20No%201%20-%2023.pdf

[26] Singh, V. (2023). Large language models in visual question answering: Leveraging LLMs to interpret complex questions and generate accurate answers based on visual input. International Journal of Advanced Engineering and Technology (IJAET), 5(S2). https://romanpub.com/resources/Vol%205%20%2C%20No%20S2%20-%2012.pdf

[27] Wang, Y., Lee, V., Wei, G. Y., & Brooks, D. (2019). Predicting new workload or CPU performance by analyzing public datasets. ACM Transactions on Architecture and Code Optimization (TACO), 15(4), 1-21.

[28] Xu, M., Kashyap, S., Zhao, H., & Kim, T. (2020, May). Krace: Data race fuzzing for kernel file systems. In 2020 IEEE Symposium on Security and Privacy (SP) (pp. 1643-1660). IEEE.

[29] Xu, Q., Shi, Y., Bamber, J., Tuo, Y., Ludwig, R., & Zhu, X. X. (2023). Physics-aware machine learning revolutionizes scientific paradigm for machine learning and process-based hydrology. arXiv preprint arXiv:2310.05227.

[30] Yao, Y. (2023). Game-of-life temperature-aware DVFS strategy for tile-based chip many-core processors. IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 13(1), 58-72.

[31] Yassin, Y. H., Jahre, M., Kjeldsberg, P. G., Aunet, S., & Catthoor, F. (2021). Fast and accurate edge computing energy modeling and DVFS implementation in GEM5 using system call emulation mode. Journal of Signal Processing Systems, 93(1), 33-48.