CROSS-PARADIGM RELAXATION DIALECT

RELAX/1

Ten operations, an MLIR and NIR-style exchange form, and a lowering matrix across substrates.

Institute for Physical AI @ Bailey Military Institute

Charlot Lab ecosystem · released CC0 for comment

Draft for comment. A portability layer is only useful if it cannot be captured; this one is CC0 by construction.

Abstract. If relaxation-style computing is to run on thermodynamic samplers, Ising machines, neuromorphic fabrics and ordinary GPUs alike, it needs a dialect that lowers to all of them. RELAX/1 specifies ten operations with a reference lowering, so that one program can be executed by several kinds of physics and the receipts compose honestly across them.

EFA-RFC-002: RELAX/1

A relaxation dialect and cross-paradigm exchange for energy-native computing

Status	Draft v0.1 — for public comment
Date	2026-08-02
License	CC0 1.0 — adopt, fork, or absorb without permission
Companion	EFA-RFC-001 (OER/1 receipts + DRIFT/1 benchmark) — every executing op in this dialect emits an OER receipt stub
Intended audience	MLIR community · NIR maintainers · Intel Lava · Extropic `thrml` · Klere `flowg`/Joule · Ferric · UXL Foundation · vendors of settling, sampling, annealing, crossbar, and

neuromorphic hardware · GPU-simulation implementers

RFC 2119 keywords apply.

---

1. Motivation

Every portability layer in production — Triton, StableHLO/IREE, SYCL, Mojo/MAX, CANN, MUSA, Vulkan compute, ggml — shares one assumption: the primitive of computation is a **kernel applied to tensors on a clock**. A thermodynamic sampling unit, an Ising annealer, a memristor crossbar doing in-situ writes, or a ternary settling fabric has *no representation* in any of them. You cannot lower a matmul onto a device whose native operation is *relax until stable*. The consequence is structural: the entire make-hardware-fungible movement commoditizes GPU-class hardware while leaving the energy-native paradigm without a compiler target, so every new substrate rebuilds its stack from zero — the precise failure mode that killed the first neuromorphic wave.

One shipping counterexample proves the fix: **NIR**, the Neuromorphic Intermediate Representation, runs one spiking model across Loihi 2, SpiNNaker2, Xylo, Speck, and simulators — because its primitives match the physics (stateful neurons and spikes, not kernels). RELAX/1 is the analogous object for relaxation and sampling computation: a small op set whose primitives are **settle, sample, anneal, local write, verify, certify** — declarative about *what equilibrium or distribution is sought*, silent about *how the substrate reaches it*. It is specified as (a) an MLIR dialect (`relax`) so it lands inside the compiler commons the industry already runs, and (b) a JSON graph-exchange form (mirroring NIR) for runtimes without MLIR.

Design north star: **a model expressed once in RELAX/1 runs today on a GPU in simulation, tomorrow on any conforming physical substrate, and produces an OER/1 receipt either way — with the provenance axes telling you honestly which one you got.**

2. Design principles

1. **Physics-neutral semantics.** Ops define fixed points, distributions, and update rules mathematically; substrates satisfy them by any means (gradient descent, analog dynamics, thermal noise, optical interference).
2. **Mandatory simulation lowering.** Every op MUST have a reference GPU/CPU lowering (via StableHLO/Triton). This is the adoption strategy — no exotic hardware required to use the dialect — and the correctness oracle for physical targets. Simulated executions are auto-tagged `provenance: simulated × emulation`.

3. **Energy-typed execution.** Every executing op takes a **budget** (joules, steps, seconds, or a temperature schedule) and MUST either satisfy it or refuse — refusal is a first-class, receipted outcome (semantics after Klere's Joule/`flowg`).
4. **Receipts native.** Executing ops emit OER/1 receipt stubs; the runtime completes and signs them. Compilation without receipts is legal; execution without receipts is nonconforming above RELAX-C1.
5. **Stochasticity in the type system.** Sampling ops are effectful and seeded where the substrate permits; native-thermal sources declare non-reproducibility explicitly rather than faking seeds.
6. **Small on purpose.** Ten core ops. Anything expressible as composition stays out of the core.

3. Core abstractions

- **Energy** (`!relax.energy<form, state_type, N>`): a scalar function over states. Forms: `ising` (pairwise couplings + biases), `hopfield` (dense/ternary associative), `factor_graph` (sparse local factors), `ebm_net` (learned neural E_θ, carried as a StableHLO/ONNX sub-module), `composite` (sum/min of energies).
- **State** (`!relax.state<domain, N>`): `binary`, `ternary`, `continuous<fN>`, `spiking` (bridged to/from NIR).
- **Budget** (`!relax.budget<joules|steps|seconds|schedule>`): compile-time-checked where static; runtime-metered otherwise.
- **FastWeights** (`!relax.fastweights<N, M>`): inference-time-mutable coupling store — the Hebbian target; on crossbar substrates this is physical memory.
- **Certificate**, **Receipt**: as defined in RFC-001.

4. Op set (v0.1 core)

Op	Signature (informal)	Semantics
`relax.define_energy`	attrs(form, params/URI) → `energy`	Declare/bind an energy landscape
`relax.encode` / `relax.decode`	tensor ⇄ `state`	Move between tensor world and state world
`relax.settle`	(energy, state, budget, tol) → (state, receipt)	Reach a local fixed point: E non-increasing along the trajectory;

Op	Signature (informal)	Semantics
		stop at tol or budget; MAY refuse
`relax.sample`	(energy, n, temp, method, seed?) → (states, receipt)	Draw n samples ∼ e^(−E/T); method ∈ gibbs | langevin | metropolis | **native_thermal**
`relax.anneal`	(energy, schedule, budget) → (state, receipt)	Settle under a declared temperature path; the op where the coupled-chain/annealing question is answered *on hardware*
`relax.local_write`	(fastweights, x, y?, rule, α, gate?) → fastweights'	Gradient-free write: rule ∈ hebbian(αxx^T) | delta | decay | surprise_gated; **the native crossbar/ settle×2 op**
`relax.nudge`	(energy, target, β) → energy'	Clamp/bias toward a target (EqProp phase two)
`relax.eqprop_grad`	(x_free, x_nudged) → local Δcouplings	Learning signal from two relaxations
`relax.verify`	(energy, state, τ) → (bool, margin, receipt)	Energy-as-verifier: accept iff E(x) ≤ τ
`relax.certify`	(energy, region, type) → certificate	type ∈ energy_monotone | contraction_region; populates OER `certificate`
`relax.compose`	(E_1, E_2, mode) → energy	mode ∈ sum (conjunction) | min (disjunction) — zero-shot compositionality

Illustrative MLIR-style listing (drift-benchmark inner loop, RFC-001 task D-A):

```
%E   = relax.define_energy ising couplings(@W_world) : !relax.energy<i
%z   = relax.encode %obs : tensor<4096xf32> -> !relax.state<ternary, 4
// think: K-step settle under a joule budget — may refuse
%z*, %r1 = relax.settle %E, %z budget(#relax.budget<joules 2.0e-6>) to
// plan: sample candidate futures, verify with the same energy
%cands, %r2 = relax.sample %E_goal, 64 temp(0.8) method(metropolis) se
%ok, %m, %r3 = relax.verify %E_goal, %best threshold(0.12)
// adapt on surprise: gradient-free write to fast weights (native cros
%Z1 = relax.local_write %Z, %z* rule(hebbian) alpha(0.02) gate(surpris
// receipts %r1..%r3 aggregate under RFC-001 §3.3 min-grade compositio
```

5. Lowering targets (v0.1 matrix)

Substrate family	`settle`	`sample`	`anneal`	`local_write`	Notes
gpu_sim (reference, REQUIRED)	unrolled descent (Triton/ StableHLO)	Gibbs/Langevin kernels	scheduled Langevin	tensor update	Correctness oracle tag `simulated×emu`
fpga_settling (Klere-class ternary fabric)	native	via settle+noise inject	clocked schedule	settle×2 / direct SRAM write	First physical conformance targe receipts via freque sweep (OER-L1)
tsu_pbit (Extropic-class; `thrml` bridge)	zero-T limit	**native_thermal**	native schedule control	host-side	DTM = chained `relax.sample`; seedless native sa declared
ising_annealer (SBM / DA / CIM / D-Wave)	zero-T anneal	repeated anneal reads	native	n/a (couplings reload)	Latency-record su QUBO adapter
cim_crossbar (memristor/PCM)	analog matvec iteration	+ noise	scheduled	**in-situ outer product — native**	The write IS the p endurance budget declared
neuromorphic_snn (via **NIR bridge**)	attractor nets	stochastic neurons	intrinsic	on-chip plasticity rules	`state<spiking` round-trips throu
photonic	iterative optical loop (exp.)	optical noise (exp.)	exp.	n/a	Marked experimer v0.1

A conforming physical backend implements ≥1 column natively and MAY delegate the rest to gpu_sim with correct provenance tags — mixed executions compose under RFC-001 min-grade rules.

6. Exchange form and interop

- **JSON graph** (NIR-mirroring): nodes = ops above; edges = states/energies; energies-by-reference (URI + hash) so learned E_θ travels as ONNX/StableHLO payload.
- **ONNX**: custom domain `org.efa.relax` for the ten ops (opset 1).
- **NIR bridge**: `state<spiking>` encode/decode is specified against NIR node types; a spiking policy can be *verified* by a `relax.verify` energy and vice versa.
- `thrml` / **Lava adapters**: thin mappings published as reference code; neither project need change to be targeted.
- **Receipts**: every runtime boundary above is an OER emission point (RFC-001 §5).

7. Conformance

Level	Meaning
RELAX-C0	Dialect-valid programs run on gpu_sim reference lowering
RELAX-C1	≥1 op lowered to a physical substrate, oracle-checked against gpu_sim
RELAX-C2	C1 + OER-L1 receipts emitted on every executing op
RELAX-C3	C2 + `relax.certify` implemented for actuation-class workloads; DRIFT/1-eligible

8. Anti-goals

Not a kernel language (Triton/Mojo remain the right tools inside `ebm_net`). Not a spiking format (NIR owns that; we bridge). Not a quantum IR (annealers enter only through their classical interface). Not a scheduler — placement/routing across substrates is the runtime's job (`flowg`-class systems), for which this dialect is the instruction set.

9. Reference-implementation plan before v1.0

JSON-Schema + MLIR dialect definition (TableGen); gpu_sim lowerings for all ten ops; two physical backends on dissimilar substrates (one settling fabric, one sampler); the RFC-001 D-A task expressed end-to-end with receipts; conformance test suite with oracle tolerances.

Acknowledgments

Primitive set distilled from: EBT descent-as-thinking (Gladstone/Du); EqProp (Scellier–Bengio); fast-weight/delta-rule lineage (Schmidhuber; Ba; Yang; Behrouz); EFA's unified readings (settle/verify/certify/write) and published negatives; Klere's budget-refusal and receipt semantics; NIR's proof that paradigm-matched primitives portabilize; Extropic's DTM as chained sampling. CC0 — the point is adoption, not attribution. Drafted with AI assistance (Claude, Anthropic) at the direction of the EFA program.