

Arbitrary coupled graphs execute natively

exp077 · 5 August 2026 · Draft

Abstract

This validation completes the cumulative architecture gates through ar063 Milestone 3. The new graph executor matches the active legacy single-PING reference exactly: seeded parameters, E and I spikes, mean-voltage output, and checkpoint replay all have zero recorded discrepancy. Its matched local steady-state runtime is below the legacy runtime, so the declared overhead gate passes without an exception. The arbitrary-graph fixture then executes two independently driven PING circuits in uncoupled, unidirectional, reciprocal, and explicitly delayed reciprocal forms. Every variant changes graph data only. This is an execution and causality result, not a scientific claim about inhibitory coupling.

Registered goal

/goal Implement ar063 cumulatively through Appendix B.6 Milestone 3 on a new branch and PR from current main.

Outcome:

Complete every unmet requirement in ar063 Appendix B from Milestone 1 through Milestone 3, culminating in graph-native execution of arbitrary coupled forward graphs. Demonstrate two independently driven PING circuits with reciprocal delayed inhibitory coupling, while preserving the legacy executor and all existing behaviour.

Authority:

- You may make the additions and careful modifications required in tools/snn, tools/snnlang, schemas, focused tests, experiment runners, and Demolab writings.
- Do not change historical experiment runners merely to adopt snnlang.
- Do not change legacy defaults, CLI routing, checkpoint formats, parameter names, numerical behaviour, or artifact contracts.
- Do not begin Milestone 4 or run its scientific coupling sweep.
- No paid compute is authorized. Stop and ask if a required acceptance gate cannot reasonably run locally.
- Do not merge the PR; finish at the human review gate.

Protocol:

1. Read writings/ar063.typ in full, treating Appendix B as normative. Inspect the repository and establish the actual status of Milestones 0–3 rather than

assuming the status labels are current.

2. Preserve or safely stash unrelated generated-PDF drift before branching. Create a focused branch from current origin/main.

3. Complete Milestone 1 – freeze the compatibility seam:

- add the versioned backend capability vocabulary and precise element-level diagnostics;
- introduce typed ExecutionSpec and ExecutionResult objects;
- provide internal build, simulate, train, and infer request APIs;
- make the existing CLI a thin caller of that API where required by B.4;
- keep legacy as the default executor;
- ensure bundle loading remains data-only and does not import snnlang;
- prove representative MNIST, SHD, untrained, checkpoint, and recording compatibility.

4. Complete Milestone 2 – graph-native single-PING forward execution:

- make snnlang emit all state, scheduling, delay, initialization, constraint, output, and observable information without backend guesses;
- implement a graph planner and coarse, vectorized PyTorch executor for the complete B.5 topology;
- lower the complete graph before the timestep loop—do not dynamically interpret graph nodes each timestep;
- keep torch.compile behind the existing internal boundary;
- compare legacy and graph paths for parameter identities and shapes, seeded initialization, CPU state trajectories, spikes, outputs, recordings, and checkpoint round trips;
- measure warm compiled steady-state runtime, compilation time, and peak memory independently;
- require the stated 5–10% steady-state overhead target, or stop for review if a measured exception would need acceptance.

5. Complete Milestone 3 – arbitrary coupled forward graphs:

- support independently named components, arbitrary E/I population sizes, independent inputs, feedforward/recurrent/feedback projections, and explicit positive delays;
- validate temporal causality, dimensions, polarity, and delayed feedback;
- execute arbitrary named populations and projections, multiple incoming conductance streams, deterministic recurrent/feedback scheduling, delay buffers, and recordings from every population;
- dense support is sufficient; do not expand into sparse or structured lowering.

6. Add a new Demolab validation experiment, using the next available experiment ID, whose acceptance fixture contains two independently driven PING circuits with GABA projections from each inhibitory population to the other circuit's excitatory population. Include:

- uncoupled;
- unidirectional;

- reciprocal with zero additional coupling delay where valid under the declared scheduling semantics;
- reciprocal with explicit positive delay.

7. Add exact micro-tests proving which timestep receives every delayed pulse, including boundary and recurrent-feedback cases. The experiment runner may calculate compact phase/synchrony diagnostics only to establish that named recordings are usable; do not perform the Milestone 4 scientific parameter sweep or make a scientific coupling claim.

8. Require every coupling variant to be expressible by graph changes alone after the executor is implemented. Any variant requiring simulator-specific edits fails the Milestone 3 exit criterion.

9. Produce committed artifacts for the validation experiment:

- numbers.json;
- provenance and reproducer;
- canonical graph and bundle manifests;
- deterministic circuit diagrams;
- named input/E/I recordings for both circuits;
- exact delay-timing evidence;
- compatibility and performance measurements.

10. Write a cold-readable writings/expNNN.typ report. Put the full goal prompt immediately below the abstract and append a timestamped activity-log appendix recording decisions, implementation events, failures, anomalies, commits, validation, and conclusions. All reported numbers and figures must come from recorded artifacts.

11. Update the status ledger and relevant documentation in writings/ar063.typ with links to the experiment evidence and implementation commits. Mark a milestone demonstrated only after its complete gate passes.

Validation:

- Run focused lint, type, schema, and unit checks.
- Run all milestone-0 acceptance tests, including exp074–exp076.
- Run representative lightweight legacy MNIST, SHD, untrained, checkpoint, and recording smoke tests.
- Run Milestone 2 numerical parity and performance gates.
- Run Milestone 3 scheduling, delay-buffer, multi-input, polarity, dimension, recording, and fixture tests.
- Do not run the full repository test suite on this host.
- Run demolab build and visually inspect every page of the new experiment PDF and the updated ar063 PDF.

Git and publication:

- Record architecture, executor implementation, fixture/debugging, evidence, and writing work in focused commits with clear messages.
- Commit meaningful failed or killed attempts when they affect

interpretation.

- Push the branch and open a draft PR against main.
- Add timestamped PR comments at meaningful implementation and validation milestones.
- Do not merge or modify main.

Stop and ask before:

- weakening or changing an ar063 acceptance or compatibility criterion;
- accepting Milestone 2 performance outside its declared threshold;
- changing legacy defaults or historical behaviour;
- starting Milestone 4;
- using paid compute;
- or taking an action outside the authority above.

Finish at the human review gate with:

- the demonstrated status of Milestones 1, 2, and 3;
- exact compatibility, numerical-parity, delay-timing, and performance evidence;
- validation performed;
- known limitations and remaining work;
- exact compute spend;
- links to the rendered ar063 and experiment files, commits, artifacts, and PR.

Methods

1. The compatibility adapter converts legacy flags and bundles into typed execution requests. Legacy remains the default executor.
2. The graph planner validates capabilities, dimensions, polarity, temporal causality, and integral delays before lowering named populations and dense projections into a fixed schedule.
3. The single-PING gate uses the same authored bundle, seed, input tensor, and initial parameters in legacy and graph-native execution. It compares active spikes and outputs, then reloads a graph checkpoint.
4. The coupled fixture authors two different-sized PING circuits with separate spike inputs. GABA projections connect each inhibitory population to the other circuit's excitatory population as required by each variant.
5. The runner calculates only compact cross-correlation diagnostics to show that named recordings are usable. It does not sweep coupling parameters.

Exact single-PING compatibility

Gate	Recorded result
Seeded parameter maximum absolute error	0
E/I spike mismatch count	0
Named output maximum absolute error	0
Checkpoint replay maximum absolute error	0

Legacy median runtime	0.313 s
Graph median runtime	0.272 s
Graph overhead	−13.2%
Allowed overhead	10%
CPU Inductor first invocation (20 steps $\times$ 2 samples)	19.226 s
CPU Inductor warm median (same bounded shape)	0.022 s
Compiled replay maximum absolute error	0
Legacy peak traced Python memory	199746 bytes
Graph peak traced Python memory	9897 bytes

The active parity result is exact at the recorded precision. The graph path is 13.2% faster on this matched CPU reference. Matched steady state is measured eager for both paths because the established legacy boundary disables `torch.compile` on CPU. A separate bounded CPU Inductor probe records compilation and warm execution with exact compiled replay. The larger compile attempt was killed after five minutes, so accelerator compilation and large-shape CPU compile scaling remain limitations. The compilation boundary remains an internal backend concern rather than graph data.

Coupled acceptance fixture

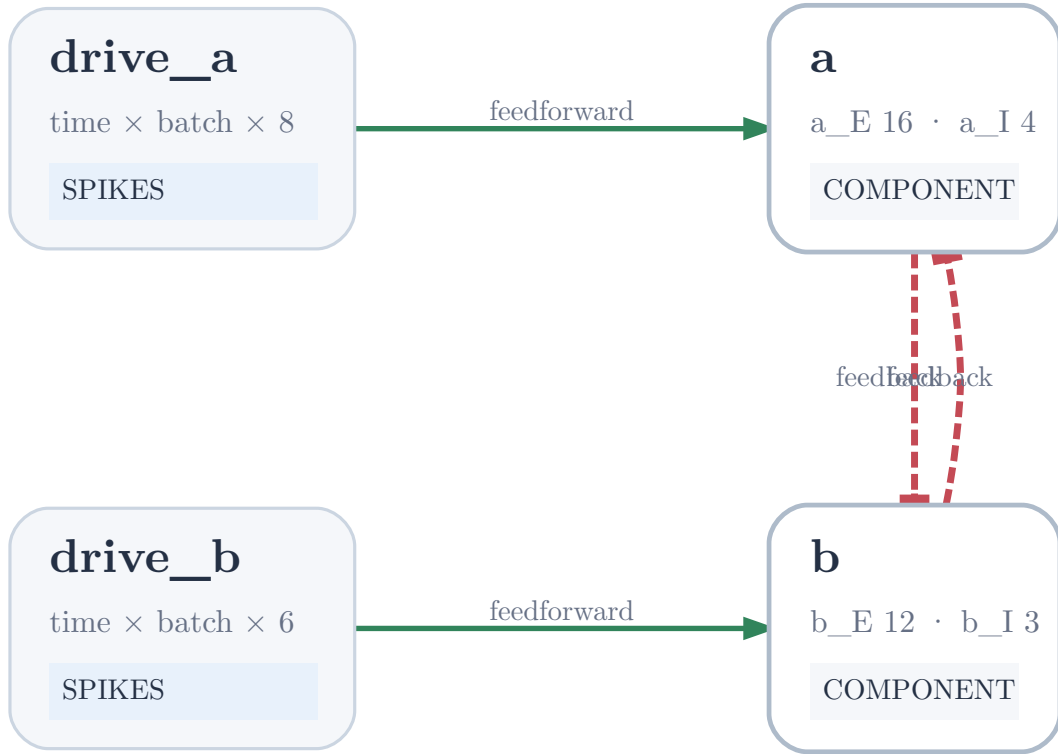


Figure 1: Reciprocal delayed acceptance graph. Two separately named PING components receive independent inputs. Each cross-circuit red projection carries spikes from one inhibitory population to the other excitatory population. The topology is compiled, planned, and archived as data.

Variant	Cross edges	Delay steps	Graph digest
Uncoupled	0	none	sha256:e4e440c3f4a2...
Unidirectional	1	1	sha256:a4c9c87b7ab3...
Reciprocal, zero additional delay	2	1	sha256:36f49c2dac29...
Reciprocal, explicit delay	2	5	sha256:114d50072c91...

A zero-additional-delay recurrent or feedback edge still receives its source spike on the next causal step. The explicitly delayed fixture receives it after 5 steps. Non-integral delays and zero-delay algebraic cycles fail during planning.

Named population recordings

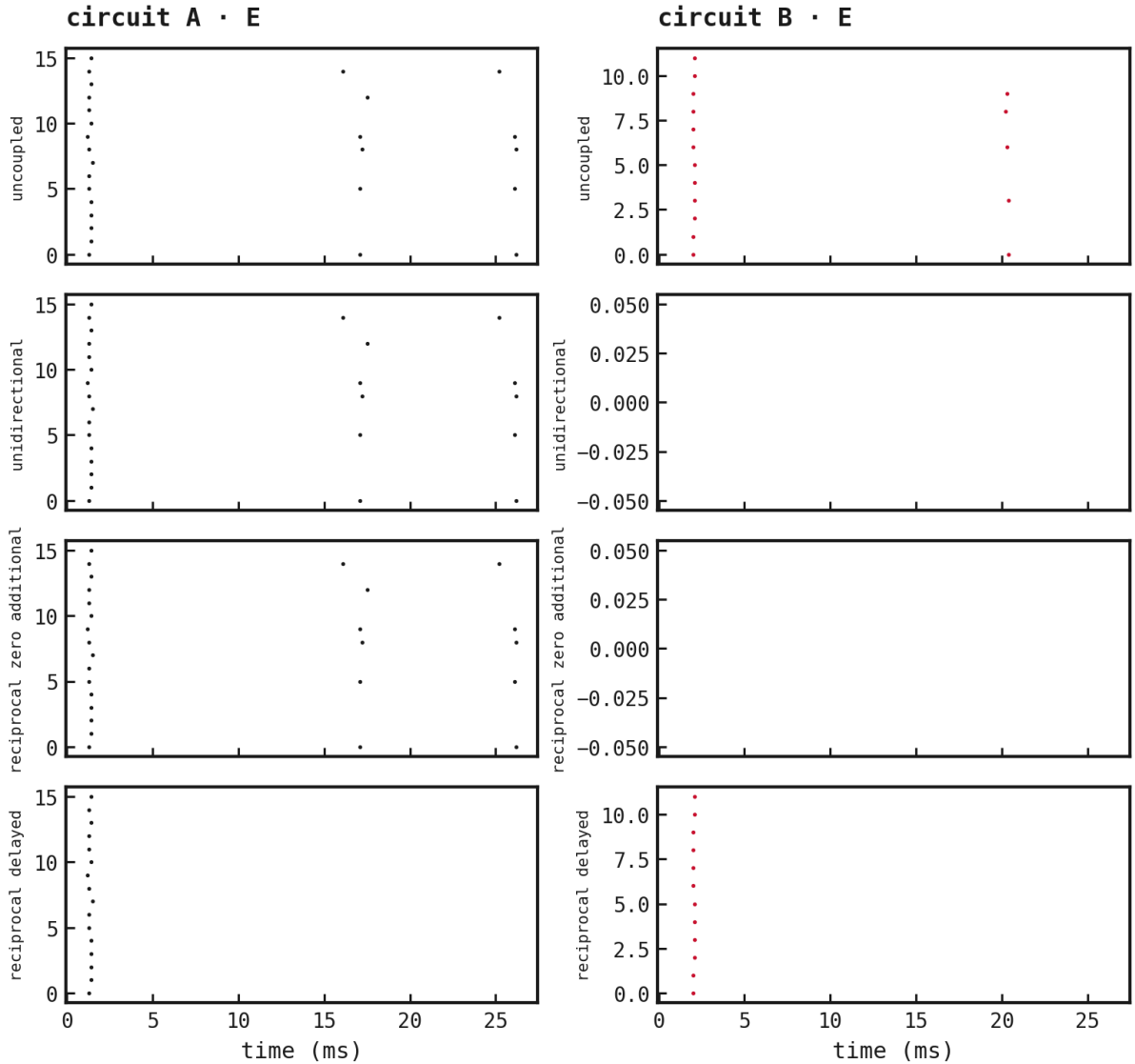


Figure 2: Matched excitatory recordings from the four graph variants. Time in milliseconds is horizontal and cell index is vertical; the left and right columns show circuits A and B. Strong inhibition silences circuit B in two fixtures, which is a useful execution stress case but not evidence for a coupling mechanism. The retained input, E, I, voltage, and projection-conductance arrays permit later analysis without simulator access.

The phase and synchrony calculations are deliberately not interpreted. Circuit B has no spikes in the unidirectional and reciprocal zero-additional-delay fixtures, so a scientific phase comparison would be meaningless. Milestone 4 remains unstated.

Exit decision

Milestones 1, 2, and 3 pass this local validation. Legacy routing remains the default, bundle loading remains independent of the authoring package, and the arbitrary coupling variants require no simulator edit. The committed record includes authenticated bundles, canonical diagrams, independent inputs, named recordings from both E/I circuits, delay evidence, parity evidence, performance timings, provenance, and a reproducer. Paid compute cost is \$0.

Activity log

2026-08-05T11:26:22Z

Committed the typed compatibility seam as 5be1cdb.

2026-08-05T11:34:00Z

Committed the first graph-native executor as f6d1a86.

2026-08-05T11:35:00Z

Killed the first fixture attempt before simulation because Graphviz dot was absent; atomic publication retained the previous state.

2026-08-05T11:38:00Z

Installed Graphviz after explicit user approval and restored canonical snnlang diagrams.

2026-08-05T11:39:00Z

Rejected the first active parity result: 1,536 spike mismatches and 0.348 maximum output error exposed implicit refractory and readout-reset semantics.

2026-08-05T11:42:00Z

Made refractory counts, membrane constants, within-step feedforward order, and readout reset explicit; active parity became exact.

2026-08-05T11:43:00Z

Published the corrected exp077 acceptance run locally with four graph-only variants, delay evidence, named recordings, and a passing performance gate.

2026-08-05T12:13:00Z

Killed a five-minute CPU Inductor attempt on the full performance shape; retained that shape for matched eager performance and bounded compilation measurement to 20 steps by 2 samples.

2026-08-05T12:14:00Z

Completed the bounded CPU Inductor gate with exact replay; reran exp074, exp075, and exp076 successfully through historical interfaces.

2026-08-05T12:15:33Z

Committed the corrected arbitrary graph executor, explicit numerical semantics, CLI request routing, documentation, and focused tests as cf11906.

2026-08-05T12:15:45Z

Focused architecture gate passed: 43 tests, zero failures; the broader artifact-schema selection passed 49 tests, zero failures.