FabSystems: Semiconductor Manufacturing Digital Twin - An Evidence-Aware Architecture, Evaluation Plan, and Integration Roadmap

Aaron Singh

Abstract **- FabSystems proposes a deterministic digital twin for wafer-lot routing, process recipes, equipment state, queue behavior, alarms, and hierarchical queen/child-node control. The first phase combines discrete-event simulation with typed telemetry and replayable fault scenarios. It is a research design: no production-fab fidelity, proprietary process recipe, or measured yield improvement is claimed. This paper defines the system model, evidence contract, risks, and staged validation plan. No physical-system or performance result is claimed without a committed reproducibility artifact.**

Index Terms - semiconductor manufacturing, digital twin, wafer routing, process control, fault diagnosis

I. INTRODUCTION

Semiconductor manufacturing couples long process routes with constrained tools, maintenance, queue-time rules, and incomplete sensor evidence. FabSystems creates an inspectable model in which every lot movement, recipe decision, state transition, and alarm is recorded. The goal is to evaluate scheduling and diagnosis methods without converting simulation into an unsupported fab claim.

Research contribution

The contribution is not a performance claim. It is a reproducible reference architecture, a falsifiable evaluation protocol, and an evidence schema that records inputs, decisions, outputs, failure modes, and claim boundaries.

II. TECHNICAL CONTEXT

A fab model is useful only when route legality, resource contention, equipment state, and event provenance remain explicit. The research design therefore separates the scheduler, tool-state machines, telemetry bus, and evidence store. This prevents a simulated cycle-time result from being mistaken for production yield or equipment capability.

III. RESEARCH QUESTIONS

1 Can event-level provenance make scheduling decisions reproducible?

2 How do tool faults and queue-time limits affect simulated cycle time?

3 Which telemetry is sufficient to localize modeled process excursions?

IV. SYSTEM MODEL AND ARCHITECTURE

A queen coordinator assigns wafer lots to child tool nodes. Each node exposes queue, recipe, state, health, and maintenance events through a typed event bus. A replay service reconstructs the global timeline, while an analysis layer correlates alarms, process variables, and simulated outcomes.

For lot l at operation k: T(l,k) = Q(l,k) + P(l,k) + D(l,k), subject to route, recipe, qualification, and queue-time constraints.

The model is a design and test abstraction. It establishes invariants and expected-value checks; it does not convert simulated variables into measured hardware behavior.

V. DETAILED OPERATION AND RATIONALE

1 Define wafer lots, recipes, routes, resources, and legal tool-state transitions.

2 Run deterministic discrete-event schedules under controlled random seeds.

3 Inject downtime, drift, queue pressure, and missing-telemetry conditions.

4 Compare scheduling and fault-localization baselines using preserved event traces.

VI. DATA AND EVIDENCE CONTRACT

Experiments use synthetic routes, recipes, equipment alarms, and maintenance events. The UCI SECOM data set provides a public process-measurement baseline for fault-prediction experiments; its anonymous features do not reproduce a complete fab.

Each run must preserve a machine-readable configuration, random seed where applicable, source revision, environment description, raw result, summary method, and known limitations. Derived plots must point back to the exact records used to produce them.

VII. IMPLEMENTATION BLUEPRINT

Lot and route registry. Defines operations, precedence, recipes, and holds. Its acceptance artifact is versioned route manifest.

Tool-node state machine. Models idle, setup, process, down, and maintenance states. Its acceptance artifact is transition trace and invariant checks.

Queen scheduler. Selects legal tool assignments under resource constraints. Its acceptance artifact is decision log with rejected alternatives.

Replay and diagnosis. Reconstructs events and correlates alarms. Its acceptance artifact is deterministic replay plus fault labels.

VIII. EXPERIMENTAL METHODOLOGY

Evaluation proceeds in four gates. Gate 1 checks deterministic behavior and invariants. Gate 2 applies controlled noise, tolerance, and injected-failure sweeps. Gate 3 repeats measurements and reports distributions rather than a favorable single run. Gate 4 introduces an external data source or bounded lab adapter only after the first three gates are reviewable.

Primary metrics are reported with sample counts, aggregation rules, uncertainty, and failure cases.

IX. REPRODUCIBILITY AND ACCEPTANCE CRITERIA

A result is accepted only when another clean environment can reproduce the reference behavior from a named revision. Every comparison uses identical budgets and held-out conditions. Failed runs remain part of the evidence record. Unknown hardware, calibration, and production conditions are labeled unknown rather than assigned synthetic values.

X. FAULT INJECTION AND NEGATIVE TESTING

Negative testing is part of the main evaluation rather than an appendix. Each risk is converted into an explicit scenario with an expected detection, rejection, abstention, or safe-state outcome. A silent success under missing

or contradictory state is treated as a failed test.

Synthetic fidelity. The failure consequence is false transfer to real fabs. The required control is label every result as simulated.

Route leakage. The failure consequence is future information reaches a decision. The required control is event-time joins and replay tests.

Unsafe control interpretation. The failure consequence is research output treated as a command. The required control is read-only adapters and interlocks.

Incomplete telemetry. The failure consequence is fault localization becomes overconfident. The required control is missingness sweeps and abstention.

XI. ANALYSIS AND INTERPRETATION

Analysis reports distributions, failure counts, and representative traces rather than a single favorable score. Baselines share the same inputs, budgets, stopping rules, and held-out conditions. When multiple configurations are explored, the selection procedure is recorded so the test set is not used as a tuning signal.

The primary metric families are Lot cycle and queue time, Equipment utilization, Fault-localization accuracy, Recovery time, Trace completeness. Each is reported with its unit, aggregation rule, sample count, spread, and failed or undefined scenarios. An improvement is meaningful only when reference correctness is preserved and the result repeats across declared evaluation seeds or groups.

XII. INTEGRATION INTERFACES

Integration occurs through versioned records rather than hidden shared state. Inputs declare units, coordinate frames or topology identifiers, timestamps, validity windows, and provenance. Outputs include a status, result payload, limitation flags, and a stable link to the run configuration.

Adapters are introduced in read-only or simulated form first. A live adapter cannot inherit the trust assigned to the reference model; it requires separate authentication, calibration, timeout, retry, and safe-state evidence.

XIII. EXTERNAL VALIDITY AND RELEASE CRITERIA

The initial conclusions are bounded by the declared simulator, public data, protocol mocks, and sampled conditions. They do not automatically generalize to other equipment, optical assemblies, wafers, stages, power systems, operating ranges, or production environments. External validity is evaluated as a new experiment, not assumed from architectural similarity.

A portfolio release requires the paper source, generated PDF, configuration schema, reproducible entry point, tests, raw or referenced evidence, and a concise limitations record. If one of these artifacts is missing, the project may remain a research design but cannot be presented as an evidence-backed implementation.

XIV. CLAIM BOUNDARIES, SAFETY, AND RISKS

FabSystems does not connect to production tools, contain proprietary recipes, or establish real yield improvement. Control commands remain inside simulation. Any later equipment adapter requires authentication, interlocks, and a separately reviewed safety envelope.

The primary research risk is a category error: treating simulation, public benchmark data, or protocol mocks as proof of physical-system performance. The mitigation is provenance at every boundary, fail-closed behavior for missing state, and an explicit review before any live interface is enabled.

XV. GOALS, MILESTONES, AND INTEGRATION

1 Implement the wafer-route and tool-state reference model.

2 Add replayable fault injection and evidence export.

3 Establish SECOM baselines for process-fault prediction.

4 Integrate Obsidian inspection evidence and Kinetic stage traces.

5 Validate a read-only lab telemetry adapter before any control interface.

XVI. CONCLUSION

FabSystems now has a research plan comparable in rigor and presentation to the established portfolio papers. Its architecture is inspectable, its evaluation is falsifiable, and its limitations are explicit. Credibility will come from completing the staged evidence artifacts, not from promoting unverified numbers.

REFERENCES

[1] UCI Machine Learning Repository, 'SECOM,' 2008.

[2] Aaron Singh, 'FabSystems research specification,' local portfolio repository, 2026.

TABLE I
SYSTEM COMPONENTS AND EVIDENCE RESPONSIBILITIES

Component	Responsibility	Required evidence
Lot and route registry	Defines operations, precedence, recipes, and holds	Versioned route manifest
Tool-node state machine	Models idle, setup, process, down, and maintenance states	Transition trace and invariant checks
Queen scheduler	Selects legal tool assignments under resource constraints	Decision log with rejected alternatives
Replay and diagnosis	Reconstructs events and correlates alarms	Deterministic replay plus fault labels

TABLE II
PLANNED EVALUATION MATRIX

Metric family	Acceptance record
Lot cycle and queue time	Repeated result with configuration, seed, revision, and limitations
Equipment utilization	Repeated result with configuration, seed, revision, and limitations
Fault-localization accuracy	Repeated result with configuration, seed, revision, and limitations
Recovery time	Repeated result with configuration, seed, revision, and limitations
Trace completeness	Repeated result with configuration, seed, revision, and limitations

TABLE III
RISK REGISTER AND CONTROLS

Risk	Failure consequence	Control
Synthetic fidelity	False transfer to real fabs	Label every result as simulated
Route leakage	Future information reaches a decision	Event-time joins and replay tests
Unsafe control interpretation	Research output treated as a command	Read-only adapters and interlocks
Incomplete telemetry	Fault localization becomes overconfident	Missingness sweeps and abstention

TABLE IV
PROJECT GOALS AND REQUIRED PROOF

ID	Goal	Completion evidence
G1	Implement the wafer-route and tool-state reference model.	Reviewable source, test, trace, or measurement artifact
G2	Add replayable fault injection and evidence export.	Reviewable source, test, trace, or measurement artifact
G3	Establish SECOM baselines for process-fault prediction.	Reviewable source, test, trace, or measurement artifact
G4	Integrate Obsidian inspection evidence and Kinetic stage traces.	Reviewable source, test, trace, or measurement artifact
G5	Validate a read-only lab telemetry adapter before any control interface.	Reviewable source, test, trace, or measurement artifact