TITLE:
Mazie Arithmetic in Computational Lithography and Transistor Simulation;
A Total Arithmetic Runtime Semantics for Degenerate Conditions in OPC, ILT, and TCAD.

AUTHOR:
Aaron C. Mazie

January 16, 2026

ABSTRACT

Computational lithography, including optical proximity correction (OPC) and inverse lithography technology (ILT), as well as transistor and device simulation (TCAD), rely on large-scale numerical computation. These pipelines use iterative optimization, FFT-based simulation, and nonlinear solvers, often accelerated on GPUs. In such environments, division by zero and invalid floating-point operations can introduce NaN states, infinities, or runtime exceptions that propagate through the computation and destabilize results.

This paper proposes a runtime arithmetic semantics, referred to as Mazie Arithmetic, in which division by zero is totalized as an identity-preserving null action. Instead of producing an undefined value, infinity, or exception, division by zero returns the original operand unchanged. The intent is not to replace classical mathematics, but to provide a deterministic, fault-tolerant arithmetic layer suitable for long-running, safety-critical, or autonomous computational pipelines.

---

1. BACKGROUND: COMPUTATION IN LITHOGRAPHY AND TRANSISTOR DESIGN

Modern semiconductor manufacturing depends on computation at multiple stages. Lithography relies on computational models of optical imaging and resist processes. Transistor design relies on numerical simulation of semiconductor physics.

In both cases, the mathematics used is implemented in software and hardware systems where runtime behavior matters as much as analytical correctness.

---

2. COMPUTATIONAL LITHOGRAPHY AS NUMERICAL OPTIMIZATION

OPC and ILT workflows involve repeated simulation and optimization cycles. A typical loop includes:

- simulate aerial image and resist response
- compute error relative to target pattern
- compute updates to mask geometry
- apply updates and repeat

These steps involve FFTs, normalization, gradient calculations, and step-size control. Division by small or zero denominators can occur in normalization steps, gradient scaling, or line-search logic.

---

3. TRANSISTOR AND DEVICE SIMULATION AS NONLINEAR SOLVING

TCAD tools solve coupled nonlinear partial differential equations representing carrier transport, electrostatics, and material behavior. These equations are discretized and solved iteratively.

Practical TCAD usage encounters:

- extreme dynamic ranges
- stiff nonlinearities
- sensitivity to mesh and step-size parameters

Division by zero or near-zero values can occur due to discretization artifacts, transient states, or numerical underflow.

---

4. RUNTIME NUMERICAL FAILURE MODES

In standard IEEE floating-point arithmetic, invalid operations such as division by zero may result in:

- NaN values
- positive or negative infinity

- hardware traps or software exceptions

In GPU-accelerated workloads, NaN propagation can silently corrupt downstream computations, making debugging difficult and reducing reproducibility.

---

5. MOTIVATION FOR A TOTAL ARITHMETIC SEMANTICS

Engineering systems often prefer deterministic behavior over analytical completeness. In many safety-critical systems, the correct response to an invalid operation is not to crash, but to preserve state.

Mazie Arithmetic adopts this philosophy by eliminating undefined arithmetic states.

---

6. CORE RULE OF MAZIE ARITHMETIC

The defining rule of Mazie Arithmetic is:

x divided by 0 equals x

This rule is interpreted operationally, not algebraically. Zero in the divisor position indicates a lack of authorization for transformation. When no valid operation can occur, the operand is preserved.

---

7. OPERATIONAL INTERPRETATION

In this framework:

- division is not assumed to be the inverse of multiplication
- division is treated as an attempted transformation
- zero indicates null jurisdiction

If jurisdiction exists, the operation proceeds.
If jurisdiction does not exist, no action is taken.

---

8. APPLICATION TO OPC AND ILT PIPELINES

In OPC and ILT, Mazie Arithmetic can be applied selectively at risk points such as:

- normalization steps
- gradient scaling
- step-size or damping calculations

Instead of generating NaN or infinity when a denominator is zero, the computation continues deterministically with identity preservation.

This allows optimization loops to remain stable and continue execution while logging boundary events for analysis.

---

9. GPU-ACCELERATED LITHOGRAPHY WORKLOADS

GPU implementations of lithography simulation emphasize throughput and parallelism. Exception handling and branching are expensive.

Mazie Arithmetic enables:

- guard-free arithmetic
- reduced exception handling
- predictable propagation of values

This improves runtime stability and reproducibility in accelerated environments.

---

10. APPLICATION TO TCAD AND DEVICE SIMULATION

In TCAD, Mazie Arithmetic is best applied in auxiliary calculations rather than core physics equations. Suitable locations include:

- step-size control
- damping factors
- normalization of update vectors

By preventing numerical blow-up during intermediate steps, solver robustness may be improved without directly altering physical model equations.

---

11. WHAT THIS APPROACH CAN INFLUENCE

Mazie Arithmetic can plausibly influence:

- stability of numerical optimization in OPC and ILT
- robustness of long-running TCAD simulations
- reduction of NaN and exception propagation
- reproducibility across hardware platforms

It acts as a computational reliability layer rather than a physical model.

---

12. WHAT THIS APPROACH DOES NOT DO

Mazie Arithmetic does not:

- change optical resolution limits
- alter EUV wavelength or mask physics
- modify transistor electrostatics
- replace classical analytical mathematics

Any impact on achievable transistor dimensions is indirect, through improved computational robustness rather than new physics.

---

13. IMPLEMENTATION STRATEGY

A practical implementation would treat Mazie Arithmetic as an optional arithmetic mode:

- classical mode for analysis and verification
- Mazie-safe mode for runtime execution

Boundary activations (division by zero events) should be logged for diagnostics and model improvement.

---

14. VALIDATION REQUIREMENTS

Any adoption of Mazie Arithmetic requires:

- benchmarking against standard OPC and TCAD workflows
- quantitative comparison of convergence and stability
- evaluation of bias introduced by identity preservation
- explicit scope control

The framework is intended as a controlled engineering tool, not a universal replacement.

---

15. CONCLUSION

Mazie Arithmetic formalizes a runtime behavior already common in engineering systems: preserve state when an operation is not authorized to act.

When applied carefully, this total arithmetic semantics may improve determinism, stability, and fault tolerance in computational lithography and transistor simulation pipelines, especially in GPU-accelerated and long-duration environments.

References

- T. Cecil, “Advances in Inverse Lithography.”
- S. Zheng et al., “Model-based OPC Extension in OpenILT.”
- OpenILT (OpenOPC/OpenILT), open-source ILT research platform.
- K. Yamazoe et al., “Fast fine-pixel aerial image calculation…” Applied Optics (Optica).
- AIP/AVS paper on fast aerial-image simulations for partially coherent systems.
- SPIE JMM (example) FFT-based aerial image simulation algorithm.
- Berkeley tech report on applying adjoint methods to ILT.
- P. He et al., “Linearized EUV mask optimization based on the adjoint…” Optica OE.
- Oracle documentation summarizing IEEE-754 floating-point exceptions.
- NVIDIA CUDA Programming Guide (NaN behavior / floating-point semantics).
- I. Laguna et al., “Detecting Floating-Point Exceptions in GPU Applications” (OSTI).
- I. Laguna et al., “BinFPE: Accurate Floating-Point Exception Detection…” ACM.
- SIAM literature on iterative methods for semiconductor device equations (example).
- CWI report on multigrid methods for semiconductor device simulation.
- “Numerical Analysis of Semiconductor Devices” (survey-style PDF).
- MIT thesis on TCAD-informed surrogate models and numerical stability/convergence debugging.