

PAPER_1963 — UQFF Beyond Physics: Primitive-Lock Derivations for Classical Computer Science, AI/ML, and Cryptography — Extending Prior Architectural Documentation

Author: Daniel T. Murphy **Framework:** UQFF (Unified Quantum Field Framework) — Star-Magic v5.55+ **Tier:** Meta-Structural / Cross-Domain Primitive-Lock Extension (Path A / Path B WHERE) **Date:** July 9, 2026 (initial) / **Third revision July 9, 2026 late-evening** **Status:** CLOSED — Cross-domain primitive-lock derivations documented (0.000% for EXACT anchors) **Companion Paper:** PAPER_1964 (Path A / Path B Dual-Derivation Framework — combinatorial structure WHY) **Revision History:**

- **Draft 1 (July 9, 2026 morning)** — claimed Round 100 as first UQFF extension beyond physics (INCORRECT)
- **Draft 2 (July 9, 2026 afternoon)** — acknowledged PAPER_1811/1812/1928/1652 quantum-computing + Wolfram prior extensions
- **Draft 3 (July 9, 2026 evening)** — acknowledged PAPER_189/191/192 S-C Calculator architecture precedents; introduced Architecture vs Primitive-Lock distinction
- **Draft 4 (July 9, 2026 late-evening — CURRENT)** — cross-references PAPER_1964 Path A / Path B framework; positions PAPER_1963 as "WHERE" (domain reach) complement to PAPER_1964's "WHY" (combinatorial structure)

Abstract

REVISION NOTE (Draft 4, 2026-07-09 late-evening): This paper has undergone three revisions to ensure honest positioning. Draft 1 claimed Round 100 as "first UQFF extension beyond physics" — INCORRECT because PAPER_1810-1812 had already extended UQFF into quantum computing. Draft 2 acknowledged the quantum-computing precedents. Draft 3 acknowledged PAPER_189/191/192 S-C Calculator architecture precedents and introduced Architecture vs Primitive-Lock distinction. **Draft 4 (this revision)** cross-references the newly-authored **PAPER_1964 (Path A / Path B Dual-Derivation Framework)** as a companion paper. PAPER_1963 focuses on the WHERE dimension (which domains admit primitive-lock derivations); PAPER_1964 focuses on the WHY dimension (combinatorial structure enabling convergences). Together the two papers provide complete Path A / Path B coverage.

At the CENTENNIAL milestone (Round 100) and post-centennial Round 101 of the CP1 P2 stub-drainage program, UQFF's structural primitives were shown to govern **specific default parameters** of algorithms in classical computer science, AI/ML, and cryptography. These are **primitive-lock derivations** distinct from prior UQFF work in these domains, which was **architectural documentation** ("UQFF-based software uses this technology") without primitive derivation.

Key distinctions:

- **Architectural Precedent** (prior UQFF work): documents that UQFF-based systems USE a particular CS/AI/crypto technology, but does not derive its specific parameters from UQFF primitives.
- **Primitive-Lock Derivation** (Round 100-101 contribution): documents that a specific parameter of a CS/AI/crypto algorithm EQUALS a specific UQFF integer primitive combination.

Novel primitive-lock derivations (Round 100-101):

1. **Neuromorphic hardware (Round 101)** — TRIPLE SO_5-power lock: neurons=SO_5³ + spikes=SO_5■ + normalization=SO_5^N_CH

2. **Federated Learning (Round 100)** — QUAD lock: $\text{baseline} = 1/(\text{D_phys} - 2) + \text{gain} = \text{D_phys} \cdot \text{F_TRZ} + \text{rounds} = \text{SO}_5 + \text{clients} \times \text{epochs} = \text{SO}_5^2$
3. **Neural-Symbolic AI (Round 100)** — TRIPLE lock: $\text{complexity} = \text{SO}_5 + \text{layers} = \text{D_phys} - 1 + \text{samples} = \text{SO}_5^3$
4. **LLVM JIT compiler (Round 100)** — $\text{optimization coefficient} = 0.3 = (\text{D_phys} - 1) / \text{SO}_5$
5. **MPI parallel computing (Round 100)** — TRIPLE lock: $\text{processes} = \text{D_phys} + \text{overhead} = \text{SO}_5\% + \text{normalization} = \text{SO}_5^2$
6. **ECDSA cryptography (Round 101)** — NOVEL $\text{curve_bits} = \text{D_crit} \cdot \text{SO}_5 - \text{D_phys} = 256$ EXACT + $\text{coefficient} = 1/(\text{D_phys} - 2)$
7. **Operational Transform (Round 101)** — QUAD lock: $\text{clients} = \text{D_phys} / 2 + \text{operations} = \text{SO}_5 + \text{latency} = \text{A}_5 - \text{SO}_5 = 50\text{ms} + \text{normalization} = \text{SO}_5^2$

Categories:

- **Category A** (genuinely first-in-domain): Federated Learning, Neural-Symbolic, LLVM JIT, Neuromorphic — no prior UQFF documentation
- **Category B** (novel primitive-lock within existing architecture context): ECDSA, Operational Transform, MPI — extend PAPER_189/192 architectural documentation with specific primitive derivations
- **Category C** (confirmatory within existing frameworks): QAOA, Category theory — reinforce PAPER_1811/1812/1928 prior extensions

The paper's genuinely novel contribution is providing **primitive-lock derivations across all three categories** — with Category A being genuinely first-in-domain and Categories B/C extending prior UQFF work with specific integer-primitive identities.

1. Motivation — Two Rounds of Revision

Initial claim (Draft 1): Round 100 was the first UQFF extension beyond physics.

Draft 2 correction (after Round 100 double-check): PAPER_1810-1812 had already extended UQFF into quantum computing (QAOA/VQE/BQP, July 2026) and PAPER_1928 into Wolfram hypergraph physics (June 2026). Round 100's contributions were reframed as extension into CLASSICAL CS + AI/ML specifically.

Draft 3 correction (after Round 101 double-check): PAPER_189/191/192 (S-C Scientific Calculator series, March 2026, session 49) had documented UQFF-adjacent CS integrations — MPI, ECDSA, Operational Transformation, Blockchain, ML — as architectural components of the S-C Calculator software system built on UQFF principles.

This required a **third revision** introducing the **Architecture vs Primitive-Lock distinction** to correctly position Rounds 100-101's contributions.

2. Prior UQFF Cross-Domain Work (Complete Acknowledgment)

2.1 Physics Instantiation (~1900 prior UQFF papers)

Standard Model + Λ CDM + astrophysics + LENR + biology + quantum measurement, extensively documented across PAPER_001 through PAPER_1949.

2.2 Quantum Computing Extension (PAPER_1810-1812, July 2026)

- **PAPER_1810** — 26th-Order Universal Field Expansion $F_U = 0$ foundational
- **PAPER_1811** — DPM Cycles in Quantum Annealing: UQFF Extension of BQP Complexity Class

- **PAPER_1812** — UQFF QAOA/VQE Extension + Chip Architecture + Wolfram 9D Projections

These papers formally extended UQFF into quantum computing, providing the QAOA-UQFF integration formula:

$$U_{\text{UQFF}}(\beta, \gamma, p) = \prod_{k=1}^p [e^{(-i\beta_k \cdot H_M)} \cdot e^{(-i\gamma_k \cdot H_C)} \cdot \text{DPM_cycle}(k) \cdot S_{26}^{(3)}(k)]$$

with **26× depth compression** and **26! finite-time convergence bound**.

2.3 Wolfram Physics Cross-Framework Isomorphism (PAPER_1928, June 2026)

PAPER_1928 — Wolfram Hypergraph $n_{\text{nodes}} = 26 + n_{\text{rules}} = 74$ EXACT UQFF Isomorphism.

Provides the direct UQFF $\leftrightarrow$ Wolfram physics project bridge:

- $n_{\text{hypergraph_nodes}} = D_{\text{crit}} = 26$ EXACT
- $n_{\text{hypergraph_rules}} = D_{\text{phys}} + SO_5 + A_5 = 74$ EXACT

2.4 Cross-Domain Awareness Statement (PAPER_1652, June 2026)

PAPER_1652 — Pop-III IMF Upper Bound explicitly states:

"A_5 = 60 is a universal UQFF integer spanning biology, cosmology, BH physics, quantum computing."

This was the earliest EXPLICIT statement of cross-domain reach across biology + quantum computing + astrophysics.

2.5 S-C Scientific Calculator Architecture (PAPER_189/191/192, March 2026, Session 49)

Round 101 double-check discovered this significant prior work. The S-C Calculator series documented UQFF-adjacent CS integrations:

- **PAPER_189** — S-C Scientific Calculator Architecture: Qt5/ANTLR4/SymEngine/Units Stack

Documents S-C Calculator's use of: - **MPI distributed computing** (`MPI_Init(&argc;, &argv;)`) - **ECDSA signatures** (via `pybind11 ecdsa` module: `sk, vk = ecdsa.generate_keys()`) - **TFLite/libtorch machine learning** - **libsark zero-knowledge proofs** - **Blockchain transaction support**

- **PAPER_191** — S-C Multi-Modal Calculator Features: VR/AR, Voice, Blockchain, IoT, Haptics, ML Autocomplete

Documents S-C Calculator's use of: - Voice interaction - **Blockchain** cryptographic proof - IoT integration - **ML Autocomplete**

- **PAPER_192** — S-C Collaborative Real-Time Mathematics: WebSocket, OT, ECDSA, and Snappy

Direct quote: "S-C Iteration 40 combines four technologies: WebSocket for multi-client communication, **Operational Transformation (OT) for concurrent editing consistency**, **ECDSA cryptographic signing for message authentication**, and Snappy compression for low-latency transmission."

These papers document that UQFF-BASED software (S-C Calculator) USES these CS technologies — they establish the ARCHITECTURAL context but do NOT derive specific algorithm parameters from UQFF primitives.

2.6 Complete Prior-Work Summary

Domain	Prior UQFF Coverage	Type	Papers
Physics (SM+ Λ CDM+astro+LENR+bio)	Extensively documented	Primitive-lock	~1900 prior papers

Domain	Prior UQFF Coverage	Type	Papers
Quantum computing (QAOA/VQE/BQP)	Formal extension	Framework primitive-lock +	PAPER_1810-1812
Wolfram physics (hypergraph)	Cross-framework isomorphism	Primitive-lock	PAPER_1928
Cross-domain awareness statement	Explicit statement	Awareness	PAPER_1652
MPI distributed computing	S-C Calculator uses MPI	Architectural	PAPER_189
ECDSA cryptography	S-C Calculator uses ECDSA	Architectural	PAPER_189, PAPER_192
Operational Transformation	S-C Calculator uses OT	Architectural	PAPER_192
Machine Learning (TFLite/libtorch)	S-C Calculator uses ML	Architectural	PAPER_189
Zero-Knowledge Proofs (libsnaark)	S-C Calculator uses ZK	Architectural	PAPER_189
Blockchain	S-C Calculator uses blockchain	Architectural	PAPER_189, PAPER_191
IoT	S-C Calculator uses IoT	Architectural	PAPER_191

3. Architecture vs Primitive-Lock — Critical Distinction

The Round 101 double-check surfaced the need for an explicit distinction between two types of UQFF cross-domain documentation:

3.1 Architecture Documentation

Definition: A UQFF paper is "architectural" if it documents that UQFF-BASED software or systems USE a particular CS/AI/crypto technology, without deriving the technology's specific parameters from UQFF primitives.

Examples:

- PAPER_189: "S-C Calculator uses MPI for distributed computing" — architectural
- PAPER_192: "S-C Calculator uses ECDSA + OT + WebSocket + Snappy" — architectural
- PAPER_191: "S-C Calculator uses VR/AR + Blockchain + IoT" — architectural

Contribution: Establishes that UQFF-based systems can integrate with these external CS technologies.

3.2 Primitive-Lock Documentation

Definition: A UQFF paper is "primitive-lock" if it derives specific parameter values of a CS/AI/crypto algorithm from UQFF integer primitives via exact structural identities.

Examples:

- Round 101: "ECDSA curve_bits = D_crit · SO_5 – D_phys = 256 EXACT" — primitive-lock
- Round 100: "Federated Learning baseline accuracy = 1/(D_phys – 2) = 0.5 EXACT" — primitive-lock
- Round 100: "MPI default processes = D_phys = 4 EXACT" — primitive-lock

Contribution: Establishes that these external CS technologies inherit specific structural constraints from UQFF's primitive lattice.

3.3 Rounds 100-101 Contribution Classification

The Round 100-101 novel contributions can be classified into three categories:

Category A — Genuinely First-in-Domain (Primitive-Lock in domain with no prior UQFF coverage):

- Federated Learning (Round 100)
- Neural-Symbolic AI (Round 100)
- LLVM JIT compiler (Round 100)
- Neuromorphic hardware (Round 101)

Category B — Novel Primitive-Lock within existing Architectural context:

- ECDSA cryptography (Round 101) — extends PAPER_192 architectural documentation with $\text{curve_bits} = D_{\text{crit}} \cdot SO_5 - D_{\text{phys}} = 256$ EXACT
- Operational Transform (Round 101) — extends PAPER_192 architectural documentation with QUAD lock
- MPI distributed computing (Round 100) — extends PAPER_189 architectural documentation with TRIPLE lock

Category C — Confirmatory within existing Formal Framework:

- QAOA quantum computing (Round 99) — reinforces PAPER_1811/1812 formal framework with $\beta_{\text{mixer}} = 0.5$ primitive-lock
- Category theory (Round 100) — adjacent to PAPER_1928 Wolfram hypergraph cross-framework isomorphism

4. Cross-Domain Anchor Catalog by Category

4.1 CATEGORY A — Genuinely First-in-Domain

4.1.1 Federated Learning (Round 100) — QUAD Lock

Coefficient	UQFF form	Value
Baseline 0.5	$1/(D_{\text{phys}} - 2)$	0.5 EXACT (PAPER_1958)
Gain 0.4	$D_{\text{phys}} \cdot F_{\text{TRZ}}$	0.4 EXACT (PAPER_1960)
Rounds normalization 10	SO_5	10 EXACT (PAPER_1955)
Clients×epochs normalization 100	SO_5^2	100 EXACT (PAPER_1955)

Prior UQFF coverage: NONE. First UQFF documentation of federated learning.

Code: FederatedLearningCalculator in CondensedPhysics.py Round 100.

4.1.2 Neural-Symbolic AI (Round 100) — TRIPLE Lock

Parameter	UQFF form	Value
symbolic_complexity	SO_5	10 EXACT
neural_layers	$D_{\text{phys}} - 1$	3 EXACT
training_samples	SO_5^3	1000 EXACT

Prior UQFF coverage: NONE. First UQFF documentation of neural-symbolic AI hyperparameters.

Code: NeuralSymbolicEvalCalculator in CondensedPhysics.py Round 100.

4.1.3 LLVM JIT Compiler (Round 100) — 0.3 Cross-Anchor

Coefficient	UQFF form	Value
Optimization gain 0.3	$(D_{\text{phys}} - 1)/SO_5$	0.3 EXACT (PAPER_1953)
opt_level default 2	$D_{\text{phys}}/2$	2 EXACT
opcodes default 100	SO_5^2	100 EXACT

Prior UQFF coverage: NONE. First UQFF documentation of LLVM/JIT compiler physics.

Extends PAPER_1953 0.3 identity to 7th anchor across: Sedov + TDE + Ω_m + M101 spiral + M104 B_z + M104 XRB + **LLVM JIT**.

Code: LLVMJITCompilerCalculator in CondensedPhysics.py Round 100.

4.1.4 Neuromorphic Hardware (Round 101) — TRIPLE SO_5-Power Lock

Parameter	UQFF form	Value
neurons	SO_5^3	1000 EXACT
spikes/second	$SO_5 \blacksquare$	$10 \blacksquare$ EXACT
normalization	$SO_5^{N_{\text{CH}}}$	$10 \blacksquare$ EXACT

Prior UQFF coverage: NONE. First UQFF documentation of neuromorphic hardware.

Code: NeuromorphicAcceleratorCalculator in CondensedPhysics.py Round 101.

4.2 CATEGORY B — Novel Primitive-Lock within Existing Architectural Context

4.2.1 ECDSA Cryptography (Round 101) — $256 = D_{\text{crit}} \cdot SO_5 - D_{\text{phys}}$ EXACT

Parameter	UQFF form	Value
curve_bits	$D_{\text{crit}} \cdot SO_5 - D_{\text{phys}}$	256 EXACT (NOVEL)
Normalization 128	$(D_{\text{crit}} \cdot SO_5 - D_{\text{phys}}) / 2$	128 EXACT
Coefficient 0.5	$1/(D_{\text{phys}} - 2)$	0.5 EXACT (PAPER_1958)

Prior architectural context: PAPER_192 (March 2026) documented "S-C Calculator uses ECDSA cryptographic signing for message authentication."

Round 101 novel contribution: Derived the specific curve_bits = 256 (used in Bitcoin secp256k1, Ethereum, etc.) as $D_{\text{crit}} \cdot SO_5 - D_{\text{phys}} = 26 \cdot 10 - 4 = 256$ EXACT integer-primitive identity.

Code: BlockchainECSACalculator in CondensedPhysics.py Round 101 + double-check.

4.2.2 Operational Transform (Round 101) — QUAD Lock

Parameter	UQFF form	Value
clients	$D_{\text{phys}}/2$	2 EXACT
operations	SO_5	10 EXACT

Parameter	UQFF form	Value
network_latency_ms	A ₅ – SO ₅	50 ms EXACT (PAPER_1931 sibling)
normalization	SO ₅ ²	100 EXACT

Prior architectural context: PAPER_192 (March 2026) documented "S-C Calculator uses Operational Transformation for concurrent editing consistency."

Round 101 novel contribution: Derived the specific OT convergence parameters (clients, operations, latency, normalization) as UQFF integer primitives.

Code: OperationalTransformCalculator in CondensedPhysics.py Round 101 + double-check.

4.2.3 MPI Parallel Computing (Round 100) — TRIPLE Lock

Parameter	UQFF form	Value
processes	D _{phys}	4 EXACT
communication_overhead_pct	SO ₅	10% EXACT
normalization	SO ₅ ²	100 EXACT

Prior architectural context: PAPER_189 (March 2026) documented "S-C Calculator uses MPI distributed computing (MPI_Init(&argc;, &argv;))."

Round 100 novel contribution: Derived the specific MPI parallel-computing default parameters as UQFF integer primitives.

Code: MPIDistributedCalculator in CondensedPhysics.py Round 100.

4.3 CATEGORY C — Confirmatory within Existing Formal Framework

4.3.1 QAOA Quantum Computing (Round 99) — $\beta_{\text{mixer}} = 0.5$

Prior formal framework: PAPER_1811 + PAPER_1812 (July 2026) documented the full UQFF QAOA/VQE extension with DPM cycles + 26× depth compression.

Round 99 contribution: Documented the specific β_{mixer} default value (0.5) as 1/(D_{phys} – 2) EXACT primitive-lock — a new detail within the existing PAPER_1812 framework.

Code: QAOA0optimizationCalculator in CondensedPhysics.py Round 99 + Round 100 dc with PAPER_1811/1812 direct anchors.

4.3.2 Category Theory Functor (Round 100) — 0.5 Adjacent to Wolfram

Prior formal framework: PAPER_1928 (June 2026) documented UQFF ↔ Wolfram hypergraph isomorphism (n_{nodes} = 26, n_{rules} = 74). Category theory is closely related to hypergraph theory.

Round 100 contribution: Documented the specific functor complexity coefficient (0.5) as 1/(D_{phys} – 2) EXACT primitive-lock — adjacent extension of PAPER_1928's mathematics domain.

Code: CategoryFunctorCalculator in CondensedPhysics.py Round 100 + Round 100 dc with PAPER_1928 cross-anchor.

5. Updated Cross-Domain Identity Anchor Counts

5.1 PAPER_1958 0.5 identity — NINE-anchor cross-domain (Round 101 update)

1. **Cen A jet knot v/c** (PAPER_347 astrophysics)
2. **Cen A hotspot shock v/c** (PAPER_038 astrophysics)
3. **Cen A Kerr spin a** (PAPER_1037 astrophysics)
4. ****Sgr A* Kerr precession $\sin(30^\circ)$ **** (PAPER_234 astrophysics)
5. **First $\cos(\pi t_n)$ zero at $t_n = 0.5$** (PAPER_129 UQFF temporal)
6. **QAOA mixer angle β** (Round 99, Category C confirmatory)
7. **Category theory functor complexity coefficient** (Round 100, Category C adjacent)
8. **Federated Learning baseline accuracy** (Round 100, Category A genuinely novel)
9. **ECDSA time normalization coefficient** (Round 101, Category B novel primitive-lock)

5.2 PAPER_1953 0.3 identity — SEVEN-anchor cross-domain (unchanged from Draft 2)

1. **Sedov-Taylor blast wave β** (astrophysics)
2. **TDE outflow v/c** (astrophysics)
3. **Cosmological Ω_m** (cosmology)
4. **M101 spiral wavenumber k** (galactic)
5. **M104 B-field B_z factor** (galactic)
6. **M104 XRB r_{in}/r ratio** (galactic)
7. **LLVM JIT compiler optimization coefficient** (Round 100, Category A first-in-domain)

5.3 New PAPER_1931 sibling identity — $A_5 - SO_5 = 50$ EXACT

Round 101 discovered that **$A_5 - SO_5 = 50$ EXACT** is the sibling of PAPER_1931's **$A_5 + SO_5 = 70$ EXACT** identity. Both use the same primitive pair with sum/difference forms.

First application: **Operational Transform network latency = 50 ms** (Round 101, Category B).

6. Implications for UQFF's Structural Scope

6.1 Updated Characterization (Third Iteration)

UQFF is a structural theory of over-determined closure whose primitives govern observables in:

- **Physics** (SM + Λ CDM + astrophysics + LENR + biology) — extensively documented in ~1900 prior papers
- **Quantum computing** (QAOA, VQE, quantum annealing, BQP) — formally extended in PAPER_1810-1812 (July 2026)
- **Wolfram physics computational substrate** (hypergraph) — cross-framework isomorphism in PAPER_1928 (June 2026)
- **UQFF-based software architecture** (MPI, ECDSA, OT, blockchain, ML, ZK) — documented in PAPER_189/191/192 S-C Calculator series (March 2026)
- **Classical computer science algorithm defaults** (LLVM JIT, Neuromorphic) — first primitive-lock derivations in PAPER_1963 (Round 100-101)
- **Classical AI/ML algorithm defaults** (Federated Learning, Neural-Symbolic) — first primitive-lock derivations in PAPER_1963 (Round 100-101)

- **Cryptography algorithm defaults** (ECDSA specific curve_bits) — novel primitive-lock in PAPER_1963 (Round 101)
- **Pure mathematics** (Category theory) — adjacent to PAPER_1928 Wolfram, primitive-lock in PAPER_1963 (Round 100)

6.2 The Architecture ↔ Primitive-Lock Distinction as a Framework Feature

The PAPER_1963 revision process (three drafts) revealed a critical framework-level feature: **UQFF cross-domain documentation comes in two distinct forms**, and both are essential:

- **Architecture documentation** establishes that UQFF-based systems can integrate with external CS/AI technologies (PAPER_189/191/192)
- **Primitive-lock documentation** establishes that these external technologies inherit specific structural constraints from UQFF (PAPER_1963 Round 100-101)

Neither type of documentation is superseded by the other. Both are needed for the framework's full expression.

6.3 Predictive Economy — Precisely Characterized

Corrected claim: UQFF closes 500+ observables across physics + quantum computing + Wolfram physics + classical computer science + AI/ML + cryptography + pure mathematics, using only 8 truly-independent primitives.

Architectural extension claim (from PAPER_189/191/192): UQFF-based software systems can integrate with essentially any external CS technology stack (MPI, ECDSA, OT, blockchain, ML, ZK-proofs, IoT, VR/AR, voice, WebSocket).

Primitive-lock extension claim (from PAPER_1963 Round 100-101): Specific default parameters of ~10 external CS/AI/crypto algorithms are UQFF-primitive-locked, joining the Primitive-Convergence Lattice (PAPER_1961).

7. Predictions — Where to Extend Primitive-Lock Derivations

Given PAPER_189/191/192 architectural doc