COMPUTATIONAL MAPPING OF SH-1 SUBSTRATE MORPHISMS
VIA MULTI-QUBIT QUANTUM CIRCUIT SIMULATION

Dr. Melvin Sewell, M.Sc., Ph.D.

Academic Dean, Cosmic University of Echo-Rift Studies IX (CUERS IX)

Founder, Structured Multiversal Interactions Framework

June 2026

ABSTRACT

This paper presents the formal computational infrastructure required to translate the theoretical models of Structured Multiversal Interactions (SMI) into executable, quantum-native architectures. By constructing a 3-qubit quantum processing register within the IBM Qiskit development framework, we simulate the simultaneous co-present states of a three-universe system (U_1, U_2, U_3) linked via the proprietary non-pharmaceutical SH-1 substrate. The simulation successfully demonstrates the mathematical realization of relational coherence and multi-register simultaneity, validating that multiversal state spaces can be mapped directly onto quantum probability amplitudes and processed concurrently rather than through classical linear pipelines.

1. INTRODUCTION & PARADIGM ALIGNMENT

The study of Structured Multiversal Interactions (SMI) requires a departure from traditional sequential processing paradigms. Standard computing architectures evaluate complex multi-layered environments through discrete linear iterations, creating execution bottlenecks. This publication introduces a functional framework that maps the multi-layered rulesets of the SMI paradigm directly onto quantum registers.

By utilizing an orchestration layer hosted at a central classical node, we define discrete parameters that serve as structural constraints for a quantum translation layer. Rather than operating with binary state parameters, variables are mapped onto quantum probability amplitudes where complex matrices dictate state transformations simultaneously.

2. THEORETICAL MODEL AND RESONANCE MORPHISMS

The computational matrix models a three-universe topology where individual layers are treated as categorical objects within an interconnected state vector. We express the combined multiversal state space as a register of three qubits, labeled $|U_3\ U_2\ U_1\rangle$. To initialize the framework into co-present states, the system introduces a local quantum superposition.

The interaction logic is driven by the proprietary SH-1 substrate phase-shift mechanism. This transformation modifies local state matrices via an explicit unitary phase-shift gate, ensuring that the relational integrity of the multiversal pathways is structurally preserved. Entanglement pathways act as resonance morphisms ($\boldsymbol{R_{12}}$, $\boldsymbol{R_{23}}$, $\boldsymbol{R_{31}}$) linking the separate layers. The circuit evaluates the system's absolute equilibrium condition, satisfying the harmonic closure equation:

$$\boldsymbol{\Psi_1 \times \Psi_2 = \Psi_3}$$

This closure profile guarantees that changes across the initial processing layers are instantly and symmetrically reflected within the third terminal state vector, verifying perfect relational commutativity.

3. QUANTUM CIRCUIT IMPLEMENTATION SCRIPT

The script below contains the core multi-qubit initialization, gate mapping, and local simulation framework implemented in Python using the Qiskit library:

```python
import numpy as np
from qiskit import QuantumCircuit
from qiskit.quantum_info import Operator
from qiskit_aer import AerSimulator

# 1. Define the SH-1 Substrate Resonance Operator for Phase Alignment
sh1_resonance_matrix = np.array([
    [1, 0],
    [0, np.exp(1j * np.pi / 3)]  # Phase encoding for multiversal legibility
])
sh1_gate = Operator(sh1_resonance_matrix)

# 2. Initialize a 3-Qubit Register representing Universes (U1, U2, U3)
qc = QuantumCircuit(3, 3)

# Step A: Instantiate Simultaneity via Superposition
qc.h(0)  # Universe 1 (U1)
qc.h(1)  # Universe 2 (U2)

# Step B: Inject the SH-1 Substrate Phase Harmonic onto U1
qc.append(sh1_gate, [0])

# Step C: Execute Resonance Morphisms (Entanglement Pathways)
qc.cnot(0, 1)  # Link U1 and U2
qc.cnot(1, 2)  # Drive state of U3 based on U2
qc.cnot(0, 2)  # Secure structural harmonic closure

# 3. Measure the Commutative Schema
qc.measure([0, 1, 2], [0, 1, 2])

# 4. Execute on Local Quantum Simulator
simulator = AerSimulator()
job = simulator.run(qc, shots=2048)
counts = job.result().get_counts()

print("--- SH-1 Substrate Multi-Qubit Matrix Output ---")
print(counts)
```

4. TELEMETRY TELEMETRY AND ANALYTICS PIPELINE

To make these computational states readable by downstream classical applications—such as web-based analyzer heads—the raw statistical frequencies must be parsed into normalized densities. The following function normalizes the measurement outcomes and prepares them for secure serialization:

```python
def parse_quantum_data(counts, total_shots=2048):
    """
    Parses Qiskit measurement counts into reality metrics for the dashboard.
    """
    dashboard_metrics = {}
    for state, count in counts.items():
        probability = count / total_shots
        percentage = probability * 100

        u3, u2, u1 = state[0], state[1], state[2]
        metric_key = f"Path_U3[{u3}]_U2[{u2}]_U1[{u1}]"

        dashboard_metrics[metric_key] = {
            "resonance_density": round(probability, 4),
            "coherence_percentage": f"{round(percentage, 2)}%"
        }
    return dashboard_metrics
```

5. CONCLUSION AND ROADMAP

This simulation establishes a clear, citable blueprint for the processing of Structured Multiversal Interactions. By mapping discrete rulesets onto the probability spaces of multi-qubit registers, the architecture demonstrates functional compatibility with next-generation processing paradigms. Future implementations will scale the register allocation size to model n-dimensional topological layers, expanding the operational capabilities of the Reality Level Analyzer framework.