

COHERENT PROCESSOR: A NEW CPU AND MICROELECTRONICS ARCHITECTURE BASED ON ODTOE PRINCIPLES

Pankratov Anton Sergeevich

Independent researcher, Kazan, Russia

E-mail: anton.s.pankratov@gmail.com

ORCID: 0009-0002-4870-2995

UDC 004.272 + 621.3.049 + 530.145 + 167.7

ABSTRACT

A conceptual architecture for a new type of processor and microelectronics is proposed, based on the principles of ODTOE (Observer-Dependent Theory of Everything). The key differences from the von Neumann architecture and modern CPUs are: (1) ternary logic instead of binary, based on the triadic architecture of observation ($\pi > 3$: observer, observed, operator — three logical states $-1, 0, +1$); (2) φ -clocking instead of a fixed frequency — a clock signal with a duration ratio of $\varphi = 1.618$ (maximum stability according to the KAM theorem); (3) toroidal interconnect topology instead of bus/mesh (radius ratio $R/r = \varphi$); (4) self-referential feedback ($\hat{O}(\hat{O})$ -loop) — the processor continuously observes its own state and reconfigures itself; (5) coherent clocking — phase synchronization instead of a global clock; (6) a resonance window $(\pi - 3)^2 \approx 2\%$ as the tolerance for parameter variation. For each principle, specific technological implementations (CMOS, superconducting electronics, photonics), performance estimates, and falsifiable predictions are provided. The project is divided into four stages: from an FPGA prototype of ternary logic (~ 5 thousand euro) to a fully functional coherent processor.

Keywords: coherent processor, ternary logic, φ -clocking, toroidal topology, ODTOE, KAM theorem, self-referential architecture, post-binary computing, spiral gap, microelectronics.

I. INTRODUCTION: THE LIMITS OF BINARY ARCHITECTURE

1.1. The Wall

Moore's Law is slowing down. CPU frequencies have stalled at ~ 5 GHz since 2005 [1]. Transistors are approaching atomic scale (~ 2 nm). Power dissipation per unit area is growing exponentially. The von Neumann bottleneck between processor and memory remains unsolved [1]. Quantum computers promise a breakthrough but require cryogenic temperatures and work only for specific tasks.

Modern processors at technology nodes ≤ 5 nm face fundamental limitations: quantum-mechanical tunneling through the transistor gate creates leakage currents that grow exponentially with decreasing oxide thickness. Dark silicon — chip areas that cannot be simultaneously activated due to the thermal budget — reaches 50 % of the die area on modern technologies [21]. The switching energy per transistor approaches the Landauer thermodynamic limit $k_B T \ln 2 \approx 3 \times 10^{-21}$ J at room temperature [22].

1.2. Why *Architecture*, Not “More Transistors”

All current improvements (multicore, cache hierarchy, speculative execution) represent optimization *within* the von Neumann paradigm (1945): binary logic (0/1), sequential instruction fetch, global clock, bus topology [23]. The problem is not in the *number* of transistors but in the *architecture* of their organization.

This situation can be compared to attempts at improving the steam engine: no matter how advanced the materials, efficiency is bounded by the Carnot cycle. To surpass this limit, a different principle is needed — the electric motor. Similarly, overcoming the von Neumann wall requires a different architectural paradigm.

Multicore scaling has hit the Amdahl’s Law utilization ceiling: with 10 % sequential code, 64 cores provide only $\sim 10\times$ speedup, not $64\times$ [1]. Speculative execution has created security vulnerabilities (Spectre, Meltdown) [24], demonstrating that optimization within the paradigm generates systemic problems.

1.3. What ODTOE Proposes

ODTOE [2] is a formal metatheory in which reality is constituted by observation: $R = \hat{O}(\Psi)$. Three structural invariants (π , φ , $(\pi - 3)^2$) determine the architecture of *any* self-consistent system [3]. A processor is a self-consistent system (the computation result must be *correct*, i.e., consistent with input data and the program). Consequently, ODTOE principles are applicable to its design.

The central claim of this work: if a processor is a realization of the observation operator, then its architecture must contain the same structural invariants as any observation operator according to ODTOE — triadicity (π), the golden ratio (φ), the spiral gap $((\pi - 3)^2)$, and self-reference ($\hat{O}(\hat{O})$) [2, 3, 17].

II. SIX PRINCIPLES OF THE COHERENT PROCESSOR

2.1. Principle 1: Ternary Logic

2.1.1. Why Three, Not Two

Binary logic: 0 and 1. Two states. The minimum for distinction. But *not* the minimum for *self-consistency*: closing the observation loop requires three components ($\pi > 3$, triadic architecture [3]).

Ternary logic: $-1, 0, +1$. Three states. Through ODTOE:

-1 = reverse action ($\iota : \mathcal{C} \rightarrow \mathcal{H}$): data returns to “potentiality” (negation, inversion).

0 = observer (O): neutral state, reference point, “undecided” (indeterminacy, NULL).

$+1$ = forward action ($\hat{O} : \mathcal{H} \rightarrow \mathcal{C}$): data is actualized (assertion, setting).

This triadic semantics is not arbitrary — it follows directly from Axiom A of ODTOE theory [2]: observer, observed, and observation operator are three inseparable components of the act of constituting reality. Binary logic, containing only two states, is fundamentally incapable of closing the self-observation loop, which constitutes a fundamental limitation of the von Neumann architecture.

2.1.2. Information Capacity

One ternary digit (trit) contains $\log_2 3 = 1.585$ bits. To represent n bits, $n/1.585 = 0.631n$ trits are needed. Savings: 37 % fewer elements at the same information capacity.

The optimal base of a number system is $e = 2.718$ (minimization of $n/\log_2 b$ at a fixed number of elements $b \times n$). The nearest integer to e : **three**. Ternary logic is informationally *optimal* [4].

Formally, the radix economy function:

$$E(b) = \frac{b}{\ln b} \quad (\text{II.0})$$

reaches its minimum at $b = e \approx 2.718$. The value $E(3) \approx 2.731$ is less than $E(2) \approx 2.885$, which mathematically confirms the advantage of the ternary system over the binary one [4].

2.1.3. Ternary Gates

Basic operations of ternary logic through ODTOE:

TER-AND (coherent conjunction): $a \otimes b = \min(a, b)$. Coherence = minimum of two components (multiplicativity of B : zero in one = zero in all).

TER-OR (coherent disjunction): $a \oplus b = \max(a, b)$. The best of two.

TER-NOT (inversion): $\neg a = -a$. Reversal of direction ($\hat{O} \leftrightarrow \iota$).

TER-ROT (rotation): $\text{rot}(a) = a + 1 \pmod{3}$: $-1 \rightarrow 0 \rightarrow +1 \rightarrow -1$. Rotation along the loop. Has no binary analogue. A fundamental ODTOE operation: one step along the triadic cycle.

TER-CONS (consensus): $\text{cons}(a, b, c) = \text{median}(a, b, c)$. Triadic “voting”. If two out of three agree — the result is determined. An analogue of P5.1: a collective configuration of three observers [2].

The completeness of the ternary gate system is proved in the standard manner: the operations {TER-NOT, TER-AND, TER-ROT} form a functionally complete system,

allowing the implementation of any function $f : \{-1, 0, +1\}^n \rightarrow \{-1, 0, +1\}$ [5, 6].

2.1.4. CMOS Implementation

Ternary logic is implementable on standard CMOS:

$$+1 = V_{DD} \text{ (supply)}. \quad 0 = V_{DD}/2 \text{ (midpoint)}. \quad -1 = \text{GND (ground)}.$$

Three voltage levels. Dual-threshold comparators (standard elements). Multi-threshold CMOS logic [5, 6] has already been developed (Motorola, Intel — experimental chips in the 1970s–80s, the Setun project by Brusentsov, MSU, 1958 [7]).

The Setun project — the first and only serial-production ternary computer — demonstrated the practical feasibility of ternary arithmetic: during 1962–1965, approximately 50 machines were produced and operated at universities and research institutes in the USSR [7]. The modern element base (≤ 7 nm FinFET) allows implementing ternary logic with significantly better performance and power characteristics.

2.2. Principle 2: φ -Clocking

2.2.1. The Problem of a Fixed Clock

Modern CPUs: a global clock signal at $\sim 3\text{--}5$ GHz. All elements switch *simultaneously*. Problems: clock skew ($\sim 10\%$ of the clock period); jitter (random phase fluctuations); electromagnetic radiation (concentrated spectrum at a single frequency); inability to adapt to the current workload.

Distributing the clock signal across a die of ~ 100 mm² requires an H-tree or mesh with buffers consuming up to 30% of the total chip power budget [12]. With the transition to ≤ 3 nm technologies, nonlinearities in signal propagation and stochastic delay variation make the global clock increasingly impractical.

2.2.2. φ -Pulsation

The clock signal is not a fixed frequency but a *sequence of pulses* with a duration ratio of φ :

$$\tau_{n+1} = \varphi \cdot \tau_n \tag{II.1}$$

Sequence: $\tau_0, \tau_0\varphi, \tau_0\varphi^2, \tau_0\varphi^3, \dots$ (each subsequent one is φ times longer). After reaching τ_{\max} — reset to τ_0 and repeat. One “phrase” contains $N_\varphi \approx 8\text{--}12$ cycles.

The average cycle duration in a phrase of N steps:

$$\bar{\tau} = \frac{\tau_0}{N} \sum_{k=0}^{N-1} \varphi^k = \frac{\tau_0}{N} \cdot \frac{\varphi^N - 1}{\varphi - 1} \tag{II.1a}$$

For $N = 8$: $\bar{\tau} \approx 6.5 \tau_0$, corresponding to an average frequency $\bar{f} \approx 0.72 f_0$, where $f_0 = 1/\tau_0$ is the base (maximum) frequency.

2.2.3. Justification Through KAM

The Kolmogorov–Arnold–Moser theorem [8, 9, 10]: in dynamical systems with perturbations, tori with a frequency ratio of φ (the most irrational number) are *maximally stable*. With φ -clocking:

Clock skew is *minimized*: φ -irrationality makes the system *maximally resistant* to phase perturbations.

Electromagnetic radiation is *spread* across the spectrum (no concentration at a single frequency): *fewer* electromagnetic interferences.

Resonant destruction is *suppressed*: rational frequency ratios cause resonances (lock-in), φ is maximally far from any rational ratio.

Quantitatively, the measure of destroyed tori is estimated as $\mu_{\text{res}} \sim |\omega_1/\omega_2 - p/q|^{-\gamma}$, where $\gamma > 0$ depends on the resonance order q . For $\omega_1/\omega_2 = \varphi$, all rational approximations p_n/q_n (Fibonacci numbers) converge *most slowly*, which minimizes μ_{res} [8, 9, 10].

2.2.4. Implementation

φ -generator on FPGA: a programmable counter switching durations according to a φ^n table (precomputed integer approximations, e.g., through Fibonacci numbers: $\tau_n \propto F_n$). Or an analog VCO (voltage-controlled oscillator) with φ -modulation of the control voltage. Implementation based on a PLL (phase-locked loop) with a non-standard divider in the feedback path is also possible [25].

2.3. Principle 3: Toroidal Interconnect Topology

2.3.1. The Problem of Bus/Mesh Topology

Bus: all modules on a single wire. Access conflicts. Bottleneck. Mesh: a regular two-dimensional lattice. Equal paths — but not all modules need each other equally.

In an $N \times N$ mesh, the average path length is $\sim 2N/3$ hops [11]. For $N = 8$ (64 nodes), this is ~ 5.3 hops. Latency scales linearly with \sqrt{N} , which is unacceptable for large chips with hundreds of functional blocks.

2.3.2. The φ -Torus

Interconnections are organized in a toroidal topology with two “radii”:

Minor radius r : fast local connections *within* a single functional block (ALU, register file, L1 cache). Continuous π -dynamics: data circulates within the block.

Major radius R : slow global connections *between* blocks (core \leftrightarrow L2 cache \leftrightarrow memory \leftrightarrow I/O bus). Discrete φ -dynamics: data moves between hierarchy levels.

$$R/r = \varphi = 1.618 \quad (\text{II.2})$$

By the KAM theorem: the maximally stable network under perturbations (overload, node failure, noise) [8, 9, 10, 17].

The average path length in a φ -torus with N nodes:

$$\bar{L}_{\varphi\text{-torus}} \sim \frac{\sqrt{N}}{\varphi} \quad (\text{II.3})$$

compared with $\bar{L}_{\text{mesh}} \sim \sqrt{N}$ for a standard mesh. The advantage is $\sim \varphi \approx 1.6$ in average latency [11, 17].

2.3.3. Physical Implementation

Toroidal die layout: functional blocks are arranged on a toroidal grid (as in Network-on-Chip, NoC [11], but with φ -scaling of distances). In practice: a ring of clusters, each cluster being a ring of cores. Two levels of rings with a ratio of φ .

Implementation in practice: chips with toroidal NoC already exist (e.g., Tiler TILE-Gx, Kalray MPPA), but without φ -scaling of radii. Adding the φ -ratio requires only a change in the topological layout, not a new technology.

2.4. Principle 4: Self-Referential Feedback ($\hat{O}(\hat{O})$ -Loop)

2.4.1. The Problem of the “Blind” Processor

A modern CPU does not “know” what it computes. It executes instructions without understanding context. Optimization (branch prediction, speculative execution) is statistical, not semantic. The branch predictor works as a black box: the Branch History Table memorizes patterns but does not *understand* the reason for branching [1].

2.4.2. The $\hat{O}(\hat{O})$ -Loop

The processor contains a *dedicated self-observation block*: a hardware module that continuously analyzes the core state (ALU utilization, cache hits, thermal map, memory access patterns) and *reconfigures* parameters in real time:

The φ -clocking phrase (lengthen/shorten depending on the workload).

Routing priorities in the φ -torus.

Balance of ternary gates (redistribution of resources among $-1, 0, +1$ paths).

This is literally $\hat{O}(\hat{O}) = \hat{O}'$ [2, Section 6.2]: the processor *observes its own observation* and *modifies its operator*.

Formally, the loop state is described by the mapping:

$$\mathbf{s}_{n+1} = \hat{O}(\hat{O}(\mathbf{s}_n)) = \hat{O}'(\mathbf{s}_n) \quad (\text{II.4})$$

where \mathbf{s}_n is the processor state vector (utilization, temperature, cache hit-rate, branch error rate), and \hat{O}' is the updated operator after self-observation. The fixed point $\mathbf{s}^* = \hat{O}'(\mathbf{s}^*)$ corresponds to the optimal operating mode of the processor.

2.4.3. Analogue in Nature

The brain: each neuron is both a “calculator” and an “observer” (through feedback connections). No “global clock”. No “bus”. There are toroidal circuits (thalamocortical loops) with feedback. The brain is an $\hat{O}(\hat{O})$ -processor [2].

Thalamocortical loops — closed circuits between the thalamus and the cerebral cortex — provide continuous feedback: the cortex sends signals to the thalamus, which filters incoming sensory information and returns processed data back to the cortex. This mechanism remarkably corresponds to the architecture of the $\hat{O}(\hat{O})$ -loop [2, 19].

2.4.4. Implementation

Hardware block: performance monitoring unit (PMU, already present in every modern CPU) + FPGA-reconfigurable logic + ML engine (neuromorphic or a simple decision table). Closed loop: PMU → analysis → reconfiguration → PMU. Update frequency: ~ 1 MHz (every ~ 1000 cycles).

The key difference from existing PMUs: in modern CPUs, monitoring is used for *statistics* (profiling, debugging) but not for *reconfiguration* in real time. In the coherent processor, the PMU is closed in a loop with the executive circuit that continuously modifies clocking, routing, and resource allocation parameters.

2.5. Principle 5: Coherent Clocking (Phase Synchronization)

2.5.1. The Problem of the Global Clock

On a ~ 10 mm die: light travels in ~ 30 ps. The clock period at 5 GHz = 200 ps. Clock propagation across the die takes ~ 15 % of the period. The clock tree consumes ~ 30 % of the chip’s energy [12].

2.5.2. Coherent Synchronization

Instead of a global clock: each block has a *local* oscillator. Blocks synchronize through *phase coupling*, like neurons in the brain or metronomes on a common platform (Huygens effect [13]).

Coherence S between blocks is maintained *naturally*, without a global tree. According to P5.1 [2]: if S between local oscillators is above the threshold — they self-synchronize. If below — they operate asynchronously (each at its own rhythm, but with a φ -ratio).

Mathematically, the dynamics of phase synchronization is described by the Kuramoto model:

$$\dot{\theta}_i = \omega_i + \frac{K}{N} \sum_{j=1}^N \sin(\theta_j - \theta_i) \quad (\text{II.5})$$

where θ_i is the phase of the i -th oscillator, ω_i is its natural frequency, K is the coupling strength. When $K > K_c$ (critical value), the oscillators spontaneously synchronize — coherence emerges without global control [13, 26].

2.6. Principle 6: Resonance Window $(\pi - 3)^2 \approx 2\%$

2.6.1. Tolerance for Variation

Modern technologies: strict tolerances on transistor parameters ($\sim 1\%$ – 3%). Exceeding the tolerance = defect. The finer the process — the more expensive the control.

2.6.2. The ODTOE Approach

The spiral gap $(\pi - 3)^2 \approx 2\%$ is an *architectural* constant [2, 3]. This is not “error” but a *working gap*: the system is *designed* for $\sim 2\%$ variation. Ternary logic with three voltage levels has *two* thresholds. The distance between thresholds: $V_{DD}/3$. The tolerance $(\pi - 3)^2 \approx 2\%$ of $V_{DD}/3$ is ~ 6 mV at $V_{DD} = 1$ V. More than sufficient for modern CMOS.

Consequence: the coherent processor is *more tolerant* of process variation, which reduces manufacturing cost and increases yield [5, 6].

Quantitative estimate: at the 7 nm technology node, the standard deviation of threshold voltage is $\sigma_{V_t} \approx 20$ mV [21]. The tolerance $(\pi - 3)^2 \times V_{DD}/3 \approx 6.6$ mV is $\sim 0.33\sigma_{V_t}$, meaning that ternary logic operates correctly with variations up to $\sim 1/3$ of the threshold voltage standard deviation. Compensation of the remaining variation is performed by the $\hat{O}(\hat{O})$ -loop.

III. COHERENT PROCESSOR ARCHITECTURE

3.1. General Scheme

Three cores (α, β, γ) — triadic architecture. Connected by a φ -toroidal network. Clocked by the φ -generator. The $\hat{O}(\hat{O})$ -loop observes everything and reconfigures.

Module hierarchy:

(a) $\hat{O}(\hat{O})$ -loop (top level): self-observation + reconfiguration. Receives data from all cores and the generator, issues adaptation commands.

(b) φ -generator: adaptive φ -clocking. Receives commands from the $\hat{O}(\hat{O})$ -loop, distributes the φ -clock to the cores.

(c) Three ternary cores (α, β, γ) : connected by a φ -toroidal network. Each core contains a ternary ALU, register file, control unit, and local L1 cache.

(d) Coherent memory: ternary cells with φ -hierarchy access (L1 \rightarrow L2 \rightarrow L3 \rightarrow main memory \rightarrow external storage).

3.2. Ternary ALU

The ternary arithmetic-logic unit:

Addition: balanced ternary arithmetic $(-1, 0, +1)$. Example: $1 + 1 = 1 \cdot 3 + (-1) = (+1, -1)_3 = 2_{10}$. No separate sign bit: the sign is *built into* the representation.

Multiplication: $a \times b$ in the balanced system. Trit multiplication — a 3×3 table. In the balanced ternary system: $(-1) \times (-1) = +1$, $(-1) \times 0 = 0$, $(-1) \times (+1) = -1$, $0 \times x = 0$, $(+1) \times (+1) = +1$.

TER-ROT: a unique operation — rotation of a trit along the cycle $-1 \rightarrow 0 \rightarrow +1 \rightarrow -1$. One step along the triadic loop. No binary analogue.

TER-CONS: a majority function of three trits. A hardware “voter”. For fault tolerance: three copies of the computation, the result = consensus (TMR, triple modular redundancy, but *built into* the logic rather than layered on top) [27].

3.3. Coherent Memory

Ternary cell: three charge levels (instead of two in DRAM). Capacity: $\times 1.585$ per cell. Addressing: φ -scaled hierarchy:

L1 cache: r_0 (minimum latency, ~ 1 ns).

L2 cache: $r_0 \cdot \varphi$ latency.

L3 cache: $r_0 \cdot \varphi^2$.

Main memory: $r_0 \cdot \varphi^3$.

External storage: $r_0 \cdot \varphi^4$.

Each level is φ times slower and φ times larger in volume. This φ -hierarchy of latencies is formalized as:

$$\tau_k = r_0 \cdot \varphi^k, \quad V_k = V_0 \cdot \varphi^k, \quad k = 0, 1, 2, 3, 4 \quad (\text{III.1})$$

where τ_k is the access latency, V_k is the volume of the k -th hierarchy level.

3.4. Instruction Set Architecture (ISA)

The ternary ISA contains standard operations + ODTOE-specific ones:

Instruction	Description	Binary Analogue
TADD	Ternary addition	ADD
TMUL	Ternary multiplication	MUL
TROT	Trit rotation ($-1 \rightarrow 0 \rightarrow +1 \rightarrow -1$)	No analogue
TCONS	Consensus of three trits	Majority gate
TNEG	Inversion ($a \rightarrow -a$)	NOT
TCOH	Block coherence measurement	No analogue
TADAPT	φ -clock reconfiguration	No analogue
TLOOP	Launch $\hat{O}(\hat{O})$ -loop	No analogue

The instructions TCOH, TADAPT, and TLOOP have no analogues in existing ISAs. They reflect fundamentally new capabilities of the coherent architecture: self-observation (TCOH – measurement of coherence S between blocks), adaptation (TADAPT – dynamic reconfiguration of the φ -phrase), and observer recursion (TLOOP – explicit launch of the $\hat{O}(\hat{O})$ cycle).

3.5. Data Format

Ternary word (tryte): 9 trits = $9 \times 1.585 = 14.3$ bits. Equivalent to a ~ 16 -bit binary word, but with 44 % fewer elements. Double tryte: 18 trits = 28.5 bits \approx 32-bit word. Quad tryte: 36 trits = 57.1 bits \approx 64-bit.

The range of representable values for a ternary word of n trits in the balanced system:

$$\text{Range} = \left[-\frac{3^n - 1}{2}, +\frac{3^n - 1}{2} \right] \quad (\text{III.2})$$

For $n = 9$: $[-9841, +9841]$, which is comparable to a 16-bit signed integer $[-32768, +32767]$, but with a symmetric range and no separate sign bit.

IV. PERFORMANCE ESTIMATION

4.1. Information Density

Parameter	Binary	Ternary	Advantage
Bits per element	1.000	1.585	$\times 1.585$
Elements for 64 bits	64	41	-36%
Switching energy (rel.)	1.00	~ 1.2	-24% net

The net energy advantage is calculated as the product of the element count reduction factor (0.64) and the per-switching energy increase factor (1.2): $0.64 \times 1.2 = 0.768$, i.e., savings of $\sim 23\%$.

4.2. φ -Clocking vs. Fixed Clock

Average frequency of the φ -clock = $f_0 \cdot \overline{\varphi^{-n}}$ (average over the phrase). For a phrase of 8 cycles: $\bar{f} \approx 0.72 f_0$. But the *peak* frequency = f_0 (the first cycle of the phrase is the shortest). The critical path is serviced at f_0 , non-critical paths at f_0/φ^n .

Energy advantage: $\sim 25\text{--}40\%$ (longer cycles consume less due to reduced switching frequency). Interference advantage: the EMI spectrum is spread (no peak at a single frequency).

Dynamic CMOS power is proportional to $P \propto CV^2f$ [1]. With φ -clocking, the average frequency $\bar{f} \approx 0.72f_0$, which yields a $\sim 28\%$ reduction in dynamic power compared to a constant frequency f_0 while maintaining peak performance.

4.3. Toroidal Network vs. Mesh

Average path length in a φ -torus with N nodes: $\sim \sqrt{N}/\varphi$ (along the minor radius — local connections are faster). In mesh: $\sim \sqrt{N}$. Advantage: $\sim \varphi \approx 1.6$ in average latency.

Additional advantage: in the φ -torus, there are no edge effects characteristic of mesh topology, where corner nodes have half as many neighbors as central ones. The toroidal topology provides connectivity uniformity [11].

4.4. The $\hat{O}(\hat{O})$ -Loop

Modern CPUs: branch prediction $\sim 95\text{--}97\%$ (statistical) [1]. The $\hat{O}(\hat{O})$ -loop: adaptation of not only branches but also clocking, routing, and resource balance. Expected advantage: 5–15% in IPC (instructions per cycle) through contextual reconfiguration.

Comparison with existing adaptive mechanisms: Intel DVFS (Dynamic Voltage and Frequency Scaling) adapts only frequency; AMD Infinity Fabric adapts only routing. The $\hat{O}(\hat{O})$ -loop adapts *all* parameters simultaneously in a closed loop.

V. TECHNOLOGICAL IMPLEMENTATION VARIANTS

5.1. Variant A: CMOS (Room Temperature)

Standard silicon. Ternary logic on dual-threshold comparators. φ -generator on PLL (phase-locked loop) or DDS (direct digital synthesis). Toroidal NoC on standard routers. $\hat{O}(\hat{O})$ -loop on an embedded FPGA block.

Advantage: compatibility with existing infrastructure. Disadvantage: coherence S is limited by thermal fluctuations.

TRL (Technology Readiness Level) estimate: 3–4. All components (ternary logic, toroidal NoC, PLL, PMU) exist individually; integration into a single coherent architecture requires design and verification.

5.2. Variant B: Superconducting Electronics (Cryogenic)

RSFQ (Rapid Single Flux Quantum) logic [14]: current pulses in superconducting circuits. Ternary RSFQ: three levels of magnetic flux ($-\Phi_0, 0, +\Phi_0$). Coherence $S \rightarrow 1$ (superconducting state). φ -clocking on Josephson junctions.

Advantage: maximum coherence, maximum speed (~ 100 GHz), minimum power dissipation ($\sim 10^{-19}$ J per switching event). Disadvantage: cryogenic temperatures (~ 4 K).

Superconducting ternary logic is of particular interest: the magnetic flux quantum $\Phi_0 = h/(2e)$ naturally admits three states ($-\Phi_0, 0, +\Phi_0$), making ternary RSFQ more natural than binary [14].

5.3. Variant C: Photonics

Optical computing: ternary logic on photon phase states ($0^\circ, 120^\circ, 240^\circ$ – three equally spaced phases on the circle). Toroidal optical resonators (microring resonators [15]) with $R/r = \varphi$. φ -clocking through mode interference.

Advantage: speed of light, parallelism, low losses. Disadvantage: integration with electronics; nonlinear optical elements are expensive.

The photonic variant of the coherent processor has a unique property: coherence S is maintained *at the physical level* through the coherence of laser radiation, which simplifies the implementation of Principle 5 (coherent clocking).

5.4. Variant D: Hybrid (Recommended for Prototype)

CMOS cores with ternary logic + φ -generator on FPGA + toroidal NoC on standard CMOS routers + $\hat{O}(\hat{O})$ -loop on an embedded ML accelerator. All on standard silicon, room temperature.

The hybrid variant minimizes technological risk: each component is implemented on the most mature technology. The FPGA block for the φ -generator and $\hat{O}(\hat{O})$ -loop provides reconfigurability during prototyping, with subsequent transfer to ASIC during scaling.

VI. IMPLEMENTATION STAGES

Stage 0: Simulation (0 euro, 1–3 months)

Software model of the ternary ALU + φ -clocking in Python/SystemVerilog. Benchmarks: comparison with a binary ALU on the same tasks (arithmetic, sorting, FFT). Falsifiable prediction: the ternary ALU with $0.631n$ elements achieves $\geq n$ -bit precision.

Specific simulation plan: (a) implementation of a balanced ternary adder and multiplier in Python; (b) implementation of a φ -clock generator in SystemVerilog; (c) benchmark: 1024-point FFT in ternary and binary arithmetic; (d) comparison of precision, latency, and operation count.

Stage 1: FPGA Prototype (5–20 thousand euro, 6–12 months)

Implementation of a ternary core on FPGA (Xilinx Zynq or Intel Cyclone). φ -generator on the FPGA's PLL block. Ternary logic is emulated with two binary bits per trit (00 = -1, 01 = 0, 10 = +1, 11 = forbidden). $\hat{O}(\hat{O})$ -loop: PMU + lookup table.

Falsifiable predictions: (a) φ -clock produces *fewer* errors at the same average frequencies than a fixed clock; (b) ternary FFT on 41 trits is comparable in precision to binary FFT on 64 bits; (c) $\hat{O}(\hat{O})$ -reconfiguration increases IPC by $\geq 5\%$.

Stage 2: ASIC Prototype (200 thousand–2 million euro, 18–36 months)

Custom chip (28/14 nm CMOS): a full-fledged ternary core, φ -toroidal NoC, hardware $\hat{O}(\hat{O})$. The first coherent processor on silicon. Use of shuttle services (MOSIS, Europractice) to reduce the cost of the first run.

Stage 3: Scaling (10 million+ euro, 3–5 years)

Multi-core coherent processor. Three cores (triadic minimal architecture) or $3 \times 3 = 9$ cores (full self-observation). Superconducting variant (RSFQ) for supercomputers. Photonic variant for AI accelerators.

At the scaling stage, the $\hat{O}(\hat{O})$ -loop becomes hierarchical: each core has a local $\hat{O}(\hat{O})$ -block, and the global $\hat{O}(\hat{O})$ observes the local ones. This is a recursive structure $\hat{O}(\hat{O}(\hat{O}))$ — self-observation of self-observation.

VII. RELATIONSHIP WITH QUANTUM COMPUTERS

7.1. The Coherent Processor Is Not Quantum

Quantum computer: operates on superpositions ($|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$). Requires cryogenic temperatures. Vulnerable to decoherence. Specialized (factorization, optimization).

Coherent processor: operates on trits ($-1, 0, +1$). Works at room temperature (CMOS). Stable (φ -KAM, $(\pi - 3)^2$ -tolerance). Universal (any computation).

It is important to emphasize: “coherence” in the processor’s name does not mean quantum coherence but *architectural coherence* — the consistency of all subsystems through phase synchronization and self-referential feedback.

7.2. Connection

The coherent processor can function as a *classical controller* for a quantum computer: the $\hat{O}(\hat{O})$ -loop controls quantum gates in real time, φ -clocking synchronizes qubits, ternary logic processes measurement results (three outcomes: $|0\rangle, |1\rangle$, error = $-1, +1, 0$).

Ternary logic naturally represents quantum measurement results: two “physical” qubit states ($|0\rangle$ and $|1\rangle$) plus a third state — error/decoherence/indeterminacy. In a standard binary architecture, the error must be encoded with an additional bit; in ternary, it is built in.

7.3. García-Pintos and the Coherent Processor

The work of García-Pintos et al. (PRX, 2026) [16]: H_{meas} — a Hamiltonian replicating the stochasticity of measurements. Through feedback ($X \cdot H_{\text{meas}}$) — control over the arrow of time. Through ODTQE: $X = \text{the ratio } \iota/\hat{O}$ [2].

Application to the coherent processor: the $\hat{O}(\hat{O})$ -loop *literally* implements the García-Pintos feedback. The parameter X is the “depth” of self-observation: at $X = 0$ — the processor is “blind” (standard). At $X > 0$ — adaptive. At $X < -2$ — the processor can *roll back* a computation (reverse the arrow of computational time) without storing intermediate states.

A new type of speculative execution: instead of “predict branch \rightarrow compute \rightarrow cancel if wrong” — “compute \rightarrow observe result \rightarrow reverse if wrong” (via $X < -2$). Savings: no need to store checkpoints.

VIII. FALSIFIABLE PREDICTIONS

#	Prediction	Verification Method	Stage
---	------------	---------------------	-------

F1	Ternary ALU: $0.631n$ trits $\geq n$ bits precision	Simulation: ternary vs. binary FFT	0
F2	φ -clock: fewer errors at the same \bar{f}	FPGA: φ -clock vs. fixed, error count	1
F3	φ -clock: EMI spectrum spread (no peak)	Spectrum analyzer at FPGA output	1
F4	Toroidal NoC: average latency $\times 1/\varphi$ vs. mesh	Simulation: φ -torus vs. mesh, $N = 16-64$ nodes	0-1
F5	$\hat{O}(\hat{O})$ -loop: IPC $\geq 5\%$ higher without it	FPGA: A/B test with/without loop on benchmarks	1
F6	Resonance window $(\pi - 3)^2$: 2% tolerance causes no errors	Simulation: random threshold variation $\pm 2\%$	0
F7	Ternary memory: $\times 1.585$ capacity per cell	ASIC: density measurement of ternary vs. binary cells	2
F8	Coherent synchronization: self-synchronization without global clock	FPGA: N oscillators with phase coupling	1

Each prediction is formulated to be falsifiable: a specific quantitative threshold, measurement method, and stage at which verification is possible are indicated. Non-confirmation of any prediction F1–F8 would require revision of the corresponding architectural principle.

IX. DEMARCATION

Claim	Status
Ternary logic is optimal ($e \approx 3$)	Mathematical fact [4]
φ -clocking is more stable than fixed (KAM)	Proven [8, 9, 10]
φ -toroidal NoC outperforms mesh	Hypothesis (testable at Stage 0)
$\hat{O}(\hat{O})$ -loop increases IPC	Hypothesis (testable at Stage 1)
Coherent synchronization without global clock	Known analogue (Huygens, neural networks), not applied to CPUs
$(\pi - 3)^2 \approx 2\%$ as working tolerance	Follows from the triadic architecture of ODTOE [2, 3]
Ternary CMOS is feasible	Proven (Setun project [7], experimental chips [5, 6])
RSFQ ternary logic	Theoretically possible , not implemented
Photonic ternary logic	Theoretically possible , demonstrations exist [15]

Clear demarcation between proven facts, grounded hypotheses, and speculative claims is necessary for scientific integrity. The author’s position: speculative elements

(especially $X < -2$ and “reversal of the computational arrow of time” from Section VII) should be regarded as directions for research, not as assertions.

X. CONCLUSION

10.1. What Is Proposed

Not “yet another CPU”. Rather, a *paradigm shift*: from the binary, bus-based, globally synchronous von Neumann architecture to the ternary, toroidal, coherently synchronous, self-referential ODTOE architecture.

10.2. Six Differences in One Table

Parameter	Von Neumann	Coherent Processor
Logic	Binary (0, 1)	Ternary ($-1, 0, +1$)
Clocking	Fixed frequency	φ -pulsation (KAM stability)
Topology	Bus / mesh	φ -torus ($R/r = \varphi$)
Self-observation	None (blind execution)	$\hat{O}(\hat{O})$ -loop
Synchronization	Global clock	Coherent (phase coupling)
Tolerances	Strict ($< 1\%$)	$(\pi - 3)^2 \approx 2\%$ (arch. gap)

10.3. Philosophy

The von Neumann processor is a *mechanism*: it blindly executes instructions, knows no context, does not observe itself.

The coherent processor is an *observer*: ternary logic ($\pi > 3$), φ -dynamics (KAM), self-reference ($\hat{O}(\hat{O})$). It does not “compute” – it *constitutes the result* through the observation loop [2].

$$\boxed{R_{\text{result}} = \hat{O}_{\text{proc}}(\Psi_{\text{input}})} \quad (\text{X.1})$$

A processor is a special case of an observer. Its architecture must follow the architecture of observation: triad (π), step (φ), gap ($(\pi - 3)^2$), self-reference ($\hat{O}(\hat{O})$). Not because “it is aesthetically pleasing” but because a *self-consistent system* is an observer, and its architecture *must* contain these invariants [2, 3, 17].

ACKNOWLEDGMENTS AND TOOLS

During the development of ODTOE theory and all articles based on it, artificial intelligence tools were used: Claude Sonnet / Opus 4.6 Extended (Chat & Code)

(Anthropic), ChatGPT 5.3 (OpenAI), Google Gemini (Google DeepMind). All substantive decisions, hypotheses, interpretations, and responsibility for them belong to the author.

CONFLICT OF INTEREST

The author declares no conflict of interest.

FUNDING

This research received no financial support.

DISCUSSION AND LIMITATIONS

(a) Ternary logic on standard CMOS requires dual-threshold comparators, which increases switching delay by $\sim 20\text{--}30\%$ compared to binary logic. This is partially compensated by the reduction in element count (37%), but experimental verification of the net advantage is required.

(b) φ -clocking produces an aperiodic spectrum, which may complicate standard JTAG/scan-chain testing based on a periodic clock. Development of new testing methods is required.

(c) The $\hat{O}(\hat{O})$ -loop with an update frequency of ~ 1 MHz introduces additional latency of $\sim 1 \mu\text{s}$. For real-time tasks, this may be unacceptable. Optimization of the update frequency remains an open question.

(d) Toroidal topology on a flat die requires long wrap-around links, increasing die area. The use of 3D stacking (via TSV — through-silicon vias) may solve this problem [28].

(e) Comparison with neuromorphic processors (Intel Loihi, IBM TrueNorth): these architectures also use local synchronization and feedback but without ternary logic and φ -clocking. The coherent processor combines the advantages of neuromorphic and classical architectures.

(f) The software question: a ternary ISA requires a new compiler, operating system, and development environment. This is a significant adoption barrier, comparable to transitioning to a new architecture.

REFERENCES

- [1] Hennessy J.L., Patterson D.A. Computer Architecture: A Quantitative Approach. — 6th ed. — Morgan Kaufmann, 2019.

- [2] Pankratov A.S. Theory of Everything: Observer-Dependent (ODTOE) // Preprint. — 2025. — 47 p.
- [3] Pankratov A.S. The Number π as a Structural Invariant // Preprint. — 2025.
- [4] Hayes B. Third Base // American Scientist. — 2001. — Vol. 89(6). — P. 490–494.
- [5] Hurst S.L. Multiple-Valued Logic — Its Status and Its Future // IEEE Transactions on Computers. — 1984. — Vol. C-33(12). — P. 1160–1179.
- [6] Smith K.C. The Prospects for Multivalued Logic: A Technology and Applications View // IEEE Transactions on Computers. — 1981. — Vol. C-30(9). — P. 619–634.
- [7] Brusentsov N.P. et al. Small Digital Computing Machine “Setun” // Vestnik MGU, ser. mat. mekh. — 1962. — No. 4. — P. 3–12.
- [8] Kolmogorov A.N. On Conservation of Conditionally Periodic Motions // Doklady Akad. Nauk SSSR. — 1954. — Vol. 98. — P. 527–530.
- [9] Arnold V.I. Small Denominators and Problems of Stability of Motion // Uspekhi Mat. Nauk. — 1963. — Vol. 18(6). — P. 91–192.
- [10] Moser J. On Invariant Curves of Area-Preserving Mappings // Nachr. Akad. Wiss. Göttingen. — 1962. — P. 1–20.
- [11] Benini L., De Micheli G. Networks on Chips: A New SoC Paradigm // Computer. — 2002. — Vol. 35(1). — P. 70–78.
- [12] Restle P.J. et al. A Clock Distribution Network for Microprocessors // IEEE Journal of Solid-State Circuits. — 2001. — Vol. 36(5). — P. 792–799.
- [13] Bennett M. et al. Huygens’s Clocks // Proceedings of the Royal Society A. — 2002. — Vol. 458. — P. 563–579.
- [14] Likharev K.K., Semenov V.K. RSFQ Logic/Memory Family: A New Josephson-Junction Technology for Sub-Terahertz-Clock-Frequency Digital Systems // IEEE Transactions on Applied Superconductivity. — 1991. — Vol. 1(1). — P. 3–28.
- [15] Bogaerts W. et al. Silicon Microring Resonators // Laser & Photonics Reviews. — 2012. — Vol. 6(1). — P. 47–73.
- [16] García-Pintos L.P., Liu Y.-K., Gorshkov A.V. Reshaping the Quantum Arrow of Time // Physical Review X. — 2026. — Vol. 16. — Art. 011028. DOI: 10.1103/118s-9vmh.
- [17] Pankratov A.S. Toroidal Topology of Reality // Preprint. — 2026.
- [18] Pankratov A.S. Coherent Thermonuclear Reactor: Conceptual Design // Preprint. — 2026.
- [19] Pankratov A.S. 3, 6, 9: Tesla’s Key Through ODTOE // Preprint. — 2026.
- [20] Pankratov A.S. Planck’s Constant from the Architecture of Observation // Preprint. — 2026.

- [21] Esmailzadeh H. et al. Dark Silicon and the End of Multicore Scaling // IEEE Micro. — 2012. — Vol. 32(3). — P. 122–134.
- [22] Landauer R. Irreversibility and Heat Generation in the Computing Process // IBM Journal of Research and Development. — 1961. — Vol. 5(3). — P. 183–191.
- [23] Von Neumann J. First Draft of a Report on the EDVAC // Moore School of Electrical Engineering, University of Pennsylvania. — 1945. — 101 p.
- [24] Kocher P. et al. Spectre Attacks: Exploiting Speculative Execution // Proceedings of the 40th IEEE Symposium on Security and Privacy. — 2019. — P. 1–19.
- [25] Razavi B. Design of Analog CMOS Integrated Circuits. — 2nd ed. — McGraw-Hill, 2017.
- [26] Kuramoto Y. Chemical Oscillations, Waves, and Turbulence. — Berlin: Springer, 1984. — 156 p.
- [27] Lyons R.E., Vanderkulk W. The Use of Triple-Modular Redundancy to Improve Computer Reliability // IBM Journal of Research and Development. — 1962. — Vol. 6(2). — P. 200–209.
- [28] Pavlidis V.F., Friedman E.G. Three-Dimensional Integrated Circuit Design. — Morgan Kaufmann, 2009. — 312 p.