

КОГЕРЕНТНЫЙ ПРОЦЕССОР: АРХИТЕКТУРА ЦПУ И МИКРОЭЛЕКТРОНИКИ НОВОГО ТИПА НА ОСНОВЕ ПРИНЦИПОВ ОДТОЕ

Coherent Processor: A New CPU and Microelectronics Architecture
Based on ODTOE Principles

Панкратов Антон Сергеевич

Pankratov Anton Sergeevich

Независимый исследователь, г. Казань, Россия

Independent researcher, Kazan, Russia

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

ORCID: 0009-0002-4870-2995

УДК 004.272 + 621.3.049 + 530.145 + 167.7

АННОТАЦИЯ

Предложена концептуальная архитектура процессора и микроэлектроники нового типа, основанная на принципах ОДТОЕ (наблюдатель-зависимая теория всего). Ключевые отличия от архитектуры фон Неймана и современных ЦПУ: (1) тройственная (тернарная) логика вместо бинарной, основанная на тройственной архитектуре наблюдения ($\pi > 3$: наблюдатель, наблюдаемое, оператор — три логических состояния $-1, 0, +1$); (2) φ -тактирование вместо фиксированной частоты — тактовый сигнал с отношением длительностей $\varphi = 1,618$ (максимальная устойчивость по КАМ-теореме); (3) тороидальная топология межсоединений вместо шинной/mesh (отношение радиусов $R/r = \varphi$); (4) самореферентная обратная связь ($\hat{O}(\hat{O})$ -контур) — процессор непрерывно наблюдает собственное состояние и перенастраивается; (5) когерентное тактирование — фазовая синхронизация вместо глобального клона; (6) резонансное окно $(\pi - 3)^2 \approx 2\%$ как допуск на вариацию параметров. Для каждого принципа указаны конкретные технологические реализации (КМОП, сверхпроводящая электроника, фотоника), оценки производительности и фальсифицируемые предсказания. Проект разбит на четыре этапа: от FPGA-прототипа тернарной логики (~ 5 тыс. евро) до полнофункционального когерентного процессора.

Ключевые слова: когерентный процессор, тернарная логика, φ -тактирование, тороидальная топология, ОДТОЕ, КАМ-теорема, самореферентная архитектура, постбинарные вычисления, спиральный зазор, микроэлектроника.

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. ВВЕДЕНИЕ: ПРЕДЕЛЫ БИНАРНОЙ АРХИТЕКТУРЫ

1.1. Стена

Закон Мура замедляется. Частоты ЦПУ застыли на ~ 5 ГГц с 2005 года [1]. Транзисторы приближаются к атомарному масштабу (~ 2 нм). Энерговыведение на единицу площади растёт экспоненциально. Фон-неймановское узкое горло (bottleneck) между процессором и памятью не решено [1]. Квантовые компьютеры обещают прорыв, но требуют криогенных температур и работают только для специфических задач.

Современные процессоры на техпроцессах ≤ 5 нм сталкиваются с фундаментальными ограничениями: квантово-механическое туннелирование через затвор транзистора создаёт токи утечки, растущие экспоненциально с уменьшением толщины оксида. Тёмный кремний (dark silicon) — области чипа, которые невозможно одновременно активировать из-за теплового бюджета — достигает 50 % площади кристалла на современных технологиях [21]. Энергопотребление на переключение транзистора приближается к термодинамическому пределу Ландауэра $k_B T \ln 2 \approx 3 \times 10^{-21}$ Дж при комнатной температуре [22].

1.2. Почему именно архитектура, а не «ещё больше транзисторов»

Все текущие улучшения (многоядерность, кэш-иерархия, спекулятивное выполнение) представляют собой оптимизацию *внутри* парадигмы фон Неймана (1945): бинарная логика (0/1), последовательная выборка команд, глобальный блок, шинная топология [23]. Проблема не в *числе* транзисторов, а в *архитектуре* их организации.

Эту ситуацию можно сравнить с попытками улучшить паровой двигатель: какие бы совершенные материалы ни применялись, КПД ограничен циклом Карно. Чтобы превзойти этот предел, нужен другой принцип — электродвигатель. Аналогично, для преодоления стены фон Неймана необходима другая архитектурная парадигма.

Многоядерность достигла потолка утилизации по закону Амдала: при 10% последовательного кода 64 ядра дают ускорение лишь $\sim 10\times$, а не $64\times$ [1]. Спекулятивное выполнение создало уязвимости безопасности (Spectre, Meltdown) [24], демонстрируя, что оптимизация внутри парадигмы порождает системные проблемы.

1.3. Что предлагает ODTOE

ODTOE [2] — формальная метатеория, в которой реальность конституируется наблюдением: $R = \hat{O}(\Psi)$. Три структурных инварианта (π , φ , $(\pi - 3)^2$) определяют архитектуру *любой* самосогласованной системы [3]. Процессор — самосогласованная система (результат вычисления должен быть *верным*, то есть согласованным с входными данными и программой). Следовательно, принципы ODTOE применимы к его проектированию.

Центральное утверждение данной работы: если процессор есть реализация оператора наблюдения, то его архитектура должна содержать те же структурные инварианты, что и любой оператор наблюдения по ODTOE — тройственность (π), золотое сечение (φ), спиральный зазор $((\pi - 3)^2)$ и самореференцию ($\hat{O}(\hat{O})$) [2, 3, 17].

II. ШЕСТЬ ПРИНЦИПОВ КОГЕРЕНТНОГО ПРОЦЕССОРА

2.1. Принцип 1: Тройственная (тернарная) логика

2.1.1. Почему три, а не два

Бинарная логика: 0 и 1. Два состояния. Минимум для различения. Но *не* минимум для *самосогласованности*: для замыкания петли наблюдения нужны три компонента ($\pi > 3$, тройственная архитектура [3]).

Тернарная логика: $-1, 0, +1$. Три состояния. Через ОДТОЕ:

-1 = обратное действие ($\iota : C \rightarrow \mathcal{H}$): данные возвращаются в «потенциальность» (отрицание, инверсия).

0 = наблюдатель (O): нейтральное состояние, точка отсчёта, «не решено» (неопределённость, NULL).

$+1$ = прямое действие ($\hat{O} : \mathcal{H} \rightarrow C$): данные актуализируются (утверждение, установка).

Эта тройственная семантика не произвольна — она напрямую следует из аксиомы А теории ОДТОЕ [2]: наблюдатель, наблюдаемое и оператор наблюдения суть три неразделимых компонента акта конституирования реальности. Бинарная логика, содержащая лишь два состояния, принципиально не способна замкнуть петлю самонаблюдения, что является фундаментальным ограничением архитектуры фон Неймана.

2.1.2. Информационная ёмкость

Один тернарный разряд (трит) содержит $\log_2 3 = 1,585$ бит. Для представления n бит нужно $n/1,585 = 0,631n$ тритов. Экономия: 37 % меньше элементов при той же информационной ёмкости.

Оптимальное основание системы счисления — $e = 2,718$ (минимизация $n/\log_2 b$ при фиксированном числе элементов $b \times n$). Ближайшее целое к e : **три**. Тернарная логика информационно *оптимальна* [4].

Формально, экономическая функция радиальной сложности:

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

достигает минимума при $b = e \approx 2,718$. Значение $E(3) \approx 2,731$ меньше $E(2) \approx 2,885$, что математически подтверждает преимущество тернарной системы над бинарной [4].

2.1.3. Тернарные вентили

Базовые операции тернарной логики через ОДТОЕ:

TER-AND (когерентная конъюнкция): $a \otimes b = \min(a, b)$. Когерентность = минимум двух компонент (мультипликативность B : ноль в одном = ноль во всём).

TER-OR (когерентная дизъюнкция): $a \oplus b = \max(a, b)$. Лучший из двух.

TER-NOT (инверсия): $\neg a = -a$. Обращение направления ($\hat{O} \leftrightarrow \iota$).

TER-ROT (вращение): $\text{rot}(a) = a + 1 \pmod{3}$: $-1 \rightarrow 0 \rightarrow +1 \rightarrow -1$. Вращение по петле. Не имеет бинарного аналога. Фундаментальная операция ОДТОЕ: один шаг по тройственному циклу.

TER-CONS (консенсус): $\text{cons}(a, b, c) = \text{median}(a, b, c)$. Тройственное «голосование». Если два из трёх согласны — результат определён. Аналог P5.1: коллективная конфигурация из трёх наблюдателей [2].

Полнота системы тернарных вентилях доказывается стандартным образом: операции {TER-NOT, TER-AND, TER-ROT} образуют функционально полную систему, позволяющую реализовать произвольную функцию $f : \{-1, 0, +1\}^n \rightarrow \{-1, 0, +1\}$ [5, 6].

2.1.4. Реализация в КМОП

Тернарная логика реализуема на стандартном КМОП:

$+1 = V_{DD}$ (питание). $0 = V_{DD}/2$ (середина). $-1 = \text{GND}$ (земля).

Три уровня напряжения. Двухпороговые компараторы (стандартные элементы). Многопороговая КМОП-логика [5, 6] уже разработана (Motorola, Intel — экспериментальные чипы в 1970–80-е, проект «Сетунь» Брусенцова, МГУ, 1958 [7]).

Проект «Сетунь» — первая и единственная серийная троичная ЭВМ — продемонстрировал практическую реализуемость тернарной арифметики: за период 1962–1965 гг. было выпущено ~ 50 машин, эксплуатировавшихся в университетах и научных институтах СССР [7]. Современная элементная база (≤ 7 нм FinFET) позволяет реализовать тернарную логику с существенно лучшими характеристиками по быстродействию и энергопотреблению.

2.2. Принцип 2: φ -тактирование

2.2.1. Проблема фиксированного клока

Современные ЦПУ: глобальный тактовый сигнал $\sim 3\text{--}5$ ГГц. Все элементы переключаются *одновременно*. Проблемы: скос клока (clock skew, $\sim 10\%$ тактового периода); джиттер (случайные колебания фазы); электромагнитное излучение (концентрированный спектр на одной частоте); невозможность адаптации к текущей нагрузке.

Распределение клокового сигнала по кристаллу площадью ~ 100 мм² требует Н-дерева или сетки с буферами, потребляющими до 30 % общего энергобюджета чипа [12]. При переходе к технологиям ≤ 3 нм нелинейности распространения сигнала и стохастическая вариация задержек делают глобальный клок всё менее практичным.

2.2.2. φ -пульсация

Тактовый сигнал — не фиксированная частота, а *последовательность импульсов* с отношением длительностей φ :

$$\tau_{n+1} = \varphi \cdot \tau_n \quad (\text{II.1})$$

Последовательность: $\tau_0, \tau_0\varphi, \tau_0\varphi^2, \tau_0\varphi^3, \dots$ (каждый следующий в φ раз длиннее). После достижения τ_{\max} — сброс к τ_0 и повторение. Одна «фраза» содержит $N_\varphi \approx 8-12$ тактов.

Средняя длительность такта во фразе из N шагов:

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

При $N = 8$: $\bar{\tau} \approx 6,5\tau_0$, что соответствует средней частоте $\bar{f} \approx 0,72 f_0$, где $f_0 = 1/\tau_0$ — базовая (максимальная) частота.

2.2.3. Обоснование через КАМ

Теорема Колмогорова–Арнольда–Мозера [8, 9, 10]: в динамических системах с возмущениями торы с отношением частот φ (наиболее иррациональное число) — *максимально устойчивы*. При φ -тактировании:

Скос клока *минимален*: φ -иррациональность делает систему *максимально устойчивой* к фазовым возмущениям.

Электромагнитное излучение *размазано* по спектру (нет концентрации на одной частоте): *меньше* электромагнитных помех.

Резонансные разрушения *подавлены*: рациональные отношения частот вызывают резонансы (lock-in), φ — максимально далеко от любого рационального отношения.

Количественно, мера разрушенных торов оценивается как $\mu_{\text{res}} \sim |\omega_1/\omega_2 - p/q|^{-\gamma}$, где $\gamma > 0$ зависит от порядка резонанса q . Для $\omega_1/\omega_2 = \varphi$ все рациональные приближения p_n/q_n (числа Фибоначчи) сходятся *медленнее всего*, что минимизирует μ_{res} [8, 9, 10].

2.2.4. Реализация

φ -генератор на FPGA: программируемый счётчик, переключающий длительности по таблице φ^n (предвычисленные целочисленные приближения, например через числа Фибоначчи: $\tau_n \propto F_n$). Или аналоговый VCO (voltage-controlled oscillator) с φ -модуляцией управляющего напряжения. Реализация на основе PLL (phase-locked loop) с нестандартным делителем в цепи обратной связи также возможна [25].

2.3. Принцип 3: Тороидальная топология межсоединений

2.3.1. Проблема шинной/mesh топологии

Шина (bus): все модули на одном проводе. Конфликты доступа. Узкое горло. Mesh (сетка): регулярная двумерная решётка. Равные пути — но не все модули одинаково нужны друг другу.

В сетке $N \times N$ средняя длина пути составляет $\sim 2N/3$ хопов [11]. При $N = 8$ (64 узла) это $\sim 5,3$ хопов. Латентность масштабируется линейно с \sqrt{N} , что неприемлемо для крупных чипов с сотнями функциональных блоков.

2.3.2. φ -тор

Межсоединения организованы в тороидальную топологию с двумя «радиусами»:

Малый радиус r : быстрые локальные связи *внутри* одного функционального блока (АЛУ, регистровый файл, кэш L1). Непрерывная π -динамика: данные циркулируют внутри блока.

Большой радиус R : медленные глобальные связи *между* блоками (ядро \leftrightarrow кэш L2 \leftrightarrow память \leftrightarrow шина ввода-вывода). Дискретная φ -динамика: данные перемещаются между уровнями иерархии.

$$R/r = \varphi = 1,618 \quad (\text{II.2})$$

По КАМ-теореме: максимально устойчивая сеть при возмущениях (перегрузка, отказ узла, шум) [8, 9, 10, 17].

Средняя длина пути в φ -торе с N узлами:

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

по сравнению с $\bar{L}_{\text{mesh}} \sim \sqrt{N}$ для стандартной сетки. Выигрыш составляет $\sim \varphi \approx 1,6$ по средней задержке [11, 17].

2.3.3. Физическая реализация

Тороидальная компоновка кристалла: функциональные блоки расположены по тороидальной сетке (как в Network-on-Chip, NoC [11], но с φ -масштабированием расстояний). Фактически: кольцо из кластеров, каждый кластер — кольцо из ядер. Два уровня колец с отношением φ .

Реализация на практике: чипы с тороидальной NoC уже существуют (например, Tiler TILE-Gx, Kalray MPPA), однако без φ -масштабирования радиусов. Добавление φ -отношения требует лишь изменения топологического layout'a, но не новой технологии.

2.4. Принцип 4: Самореферентная обратная связь ($\hat{O}(\hat{O})$ -контур)

2.4.1. Проблема «слепого» процессора

Современный ЦПУ не «знает», что вычисляет. Он исполняет инструкции без понимания контекста. Оптимизация (предсказание ветвлений, спекулятивное выполнение) — статистическая, не содержательная. Предсказатель ветвлений работает как чёрный ящик: таблица историй (Branch History Table) запоминает паттерны, но не *понимает* причину ветвления [1].

2.4.2. $\hat{O}(\hat{O})$ -контур

Процессор содержит *выделенный блок самонаблюдения*: аппаратный модуль, непрерывно анализирующий состояние ядра (утилизация АЛУ, попадания в кэш, тепловая карта, паттерны доступа к памяти) и *перенастраивающий* параметры в реальном времени:

φ -фразу тактирования (удлинить/укоротить в зависимости от нагрузки).

Приоритеты маршрутизации в φ -торе.

Баланс тернарных вентиляей (перераспределение ресурсов между -1 , 0 , $+1$ путями).

Это буквально $\hat{O}(\hat{O}) = \hat{O}'$ [2, раздел 6.2]: процессор *наблюдает своё наблюдение и изменяет свой оператор*.

Формально, состояние контура описывается отображением:

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

где \mathbf{s}_n — вектор состояния процессора (утилизация, температура, hit-rate кэша, частота ошибок ветвлений), а \hat{O}' — обновлённый оператор после самонаблюдения. Неподвижная точка $\mathbf{s}^* = \hat{O}'(\mathbf{s}^*)$ соответствует оптимальному режиму работы процессора.

2.4.3. Аналог в природе

Мозг: каждый нейрон — и «вычислитель», и «наблюдатель» (через обратные связи). Нет «глобального клона». Нет «шины». Есть тороидальные контуры (таламокортикальные петли) с обратной связью. Мозг — $\hat{O}(\hat{O})$ -процессор [2].

Таламокортикальные петли — замкнутые контуры между таламусом и корой головного мозга — обеспечивают непрерывную обратную связь: кора посылает сигналы в таламус, который фильтрует входящую сенсорную информацию и возвращает обработанные данные обратно в кору. Этот механизм удивительно точно соответствует архитектуре $\hat{O}(\hat{O})$ -контура [2, 19].

2.4.4. Реализация

Аппаратный блок: performance monitoring unit (PMU, уже есть в каждом современном ЦПУ) + FPGA-перенастраиваемая логика + ML-движок (нейроморфный или простая таблица решений). Замкнутая петля: PMU → анализ → перенастройка → PMU. Частота обновления: ~ 1 МГц (каждые ~ 1000 тактов).

Ключевое отличие от существующих PMU: в современных ЦПУ мониторинг используется для *статистики* (профилирование, отладка), но не для *перенастройки* в реальном времени. В когерентном процессоре PMU замкнут в петлю с исполнительным контуром, который непрерывно модифицирует параметры тактирования, маршрутизации и распределения ресурсов.

2.5. Принцип 5: Когерентное тактирование (фазовая синхронизация)

2.5.1. Проблема глобального клона

На кристалле ~ 10 мм: свет проходит за ~ 30 пс. Тактовый период при 5 ГГц = 200 пс. Распространение клона по кристаллу занимает ~ 15 % периода. Клоковое дерево потребляет ~ 30 % энергии чипа [12].

2.5.2. Когерентная синхронизация

Вместо глобального клона: каждый блок имеет *локальный* осциллятор. Блоки синхронизируются через *фазовую связь* (phase coupling), как нейроны в мозге или как метрономы на общей платформе (эффект Гюйгенса [13]).

Когерентность S между блоками поддерживается *естественно*, без глобального дерева. По P5.1 [2]: если S между локальными осцилляторами выше порога — они самосинхронизируются. Если ниже — работают асинхронно (каждый в своём ритме, но с φ -отношением).

Математически, динамика фазовой синхронизации описывается моделью Курамото:

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

где θ_i — фаза i -го осциллятора, ω_i — его собственная частота, K — сила связи. При $K > K_c$ (критическое значение) осцилляторы спонтанно синхронизируются — возникает когерентность без глобального управления [13, 26].

2.6. Принцип 6: Резонансное окно $(\pi - 3)^2 \approx 2\%$

2.6.1. Допуск на вариацию

Современные технологии: жёсткие допуски на параметры транзисторов ($\sim 1\% - 3\%$). Выход за допуск = брак. Чем мельче техпроцесс — тем дороже контроль.

2.6.2. ОДТОЕ-подход

Спиральный зазор $(\pi - 3)^2 \approx 2\%$ — архитектурная константа [2, 3]. Это не «погрешность», а *рабочий зазор*: система *рассчитана* на вариацию $\sim 2\%$. Тернарная логика с тремя уровнями напряжения имеет *два* порога. Расстояние между порогами: $V_{DD}/3$. Допуск $(\pi - 3)^2 \approx 2\%$ от $V_{DD}/3$ — это ~ 6 мВ при $V_{DD} = 1$ В. Более чем достаточно для современного КМОП.

Следствие: когерентный процессор *терпимее* к вариациям процесса (process variation), что снижает стоимость производства и повышает выход годных [5, 6].

Количественная оценка: при техпроцессе 7 нм стандартное отклонение порогового напряжения составляет $\sigma_{V_i} \approx 20$ мВ [21]. Допуск $(\pi - 3)^2 \times V_{DD}/3 \approx 6,6$ мВ — это $\sim 0,33\sigma_{V_i}$, что означает, что тернарная логика корректно работает при вариациях до $\sim 1/3$ стандартного отклонения порогового напряжения. Компенсация оставшейся вариации осуществляется $\hat{O}(\hat{O})$ -контуром.

III. АРХИТЕКТУРА КОГЕРЕНТНОГО ПРОЦЕССОРА

3.1. Общая схема

Три ядра (α, β, γ) — тройственная архитектура. Связаны φ -тороидальной сетью. Тактируются φ -генератором. $\hat{O}(\hat{O})$ -контур наблюдает всё и перенастраивает.

Иерархия модулей:

(а) $\hat{O}(\hat{O})$ -контур (верхний уровень): самонаблюдение + перенастройка. Получает данные от всех ядер и генератора, выдаёт команды адаптации.

(б) φ -генератор: адаптивное φ -тактирование. Принимает команды от $\hat{O}(\hat{O})$ -контура, распределяет φ -клок по ядрам.

(в) Три тернарных ядра (α, β, γ) : связаны φ -тороидальной сетью. Каждое ядро содержит тернарное АЛУ, регистровый файл, блок управления и локальный кэш L1.

(г) Когерентная память: тернарные ячейки с φ -иерархией доступа (L1 \rightarrow L2 \rightarrow L3 \rightarrow основная память \rightarrow внешнее хранилище).

3.2. Тернарное АЛУ

Тернарное арифметико-логическое устройство:

Сложение: сбалансированная тернарная арифметика $(-1, 0, +1)$. Пример: $1 + 1 = 1 \cdot 3 + (-1) = (+1, -1)_3 = 2_{10}$. Нет отдельного знакового бита: знак *встроен* в представление.

Умножение: $a \times b$ в сбалансированной системе. Перемножение тритов — таблица 3×3 . В сбалансированной тернарной системе: $(-1) \times (-1) = +1$, $(-1) \times 0 = 0$, $(-1) \times (+1) = -1$, $0 \times x = 0$, $(+1) \times (+1) = +1$.

TER-ROT: уникальная операция — вращение трита по циклу $-1 \rightarrow 0 \rightarrow +1 \rightarrow -1$. Один шаг по тройственной петле. Нет бинарного аналога.

TER-CONS: мажоритарная функция от трёх тритов. Аппаратный «голосователь». Для отказоустойчивости: три копии вычисления, результат = консенсус (TMR, triple modular redundancy, но *встроенный* в логику, а не надстроенный) [27].

3.3. Когерентная память

Тернарная ячейка: три уровня заряда (вместо двух в DRAM). Ёмкость: $\times 1,585$ на ячейку. Адресация: φ -масштабированная иерархия:

L1 кэш: r_0 (минимальная задержка, ~ 1 нс).

L2 кэш: $r_0 \cdot \varphi$ задержка.

L3 кэш: $r_0 \cdot \varphi^2$.

Основная память: $r_0 \cdot \varphi^3$.

Внешнее хранилище: $r_0 \cdot \varphi^4$.

Каждый уровень — в φ раз медленнее и в φ раз больше по объёму. Эта φ -иерархия задержек формализуется как:

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

где τ_k — задержка доступа, V_k — объём k -го уровня иерархии.

3.4. Архитектура набора команд (ISA)

Тернарная ISA содержит стандартные операции + ODTOE-специфичные:

Команда	Описание	Бинарный аналог
TADD	Тернарное сложение	ADD
TMUL	Тернарное умножение	MUL
TROT	Вращение трита ($-1 \rightarrow 0 \rightarrow +1 \rightarrow -1$)	Нет аналога
TCONS	Консенсус трёх тритов	Мажоритарный вентиль

TNEG	Инверсия ($a \rightarrow -a$)	NOT
ТСОН	Измерение когерентности блока	Нет аналога
TADAPT	Перенастройка φ -клока	Нет аналога
TLOOP	Запуск $\hat{O}(\hat{O})$ -контура	Нет аналога

Команды ТСОН, TADAPT и TLOOP не имеют аналогов в существующих ISA. Они отражают фундаментально новые возможности когерентной архитектуры: самонаблюдение (ТСОН — измерение когерентности S между блоками), адаптацию (TADAPT — динамическая перенастройка φ -фразы) и рекурсию наблюдателя (TLOOP — явный запуск цикла $\hat{O}(\hat{O})$).

3.5. Формат данных

Тернарное слово (трайт): 9 тритов = $9 \times 1,585 = 14,3$ бит. Эквивалентно \sim 16-битному бинарному слову, но на 44 % меньше элементов. Двойной трайт: 18 тритов = 28,5 бит \approx 32-битное слово. Четверной трайт: 36 тритов = 57,1 бит \approx 64-битное.

Диапазон представимых значений для тернарного слова из n тритов в сбалансированной системе:

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

Для $n = 9$: $[-9841, +9841]$, что сравнимо с 16-битным знаковым целым $[-32768, +32767]$, но с симметричным диапазоном и без отдельного знакового бита.

IV. ОЦЕНКА ПРОИЗВОДИТЕЛЬНОСТИ

4.1. Информационная плотность

Параметр	Бинарный	Тернарный	Выигрыш
Бит на элемент	1,000	1,585	$\times 1,585$
Элементов для 64 бит	64	41	-36 %
Энергия переключения (отн.)	1,00	$\sim 1,2$	-24 % нетто
Помехоустойчивость	$V_{DD}/2$	$V_{DD}/3$	-33 % (компенсируется)

Нетто-выигрыш по энергии вычисляется как произведение коэффициента снижения числа элементов (0,64) на коэффициент увеличения энергии одного переключения (1,2): $0,64 \times 1,2 = 0,768$, то есть экономия ~ 23 %.

4.2. φ -тактирование vs. фиксированный клок

Средняя частота φ -клока = $f_0 \cdot \overline{\varphi^{-n}}$ (среднее по фразе). Для фразы из 8 тактов: $\bar{f} \approx 0,72 f_0$. Но *пиковая* частота = f_0 (первый такт фразы — самый короткий). Критический путь обслуживается при f_0 , некритический — при f_0/φ^n .

Выигрыш по энергии: $\sim 25\text{--}40\%$ (длинные такты потребляют меньше из-за снижения частоты переключения). Выигрыш по помехам: ЭМИ-спектр размазан (нет пика на одной частоте).

Динамическая мощность КМОП пропорциональна $P \propto CV^2f$ [1]. При φ -тактировании средняя частота $\bar{f} \approx 0,72f_0$, что даёт снижение динамической мощности на $\sim 28\%$ по сравнению с постоянной частотой f_0 при сохранении пиковой производительности.

4.3. Тороидальная сеть vs. mesh

Средняя длина пути в φ -торе с N узлами: $\sim \sqrt{N}/\varphi$ (по малому радиусу — локальные связи быстрее). В mesh: $\sim \sqrt{N}$. Выигрыш: $\sim \varphi \approx 1,6$ по средней задержке.

Дополнительное преимущество: в φ -торе отсутствуют граничные эффекты (edge effects), характерные для mesh-топологии, где угловые узлы имеют вдвое меньше соседей, чем центральные. Тороидальная топология обеспечивает однородность связности [11].

4.4. $\hat{O}(\hat{O})$ -контур

Современные ЦПУ: предсказание ветвлений $\sim 95\text{--}97\%$ (статистическое) [1]. $\hat{O}(\hat{O})$ -контур: адаптация не только ветвлений, но и тактирования, маршрутизации, баланса ресурсов. Ожидаемый выигрыш: 5–15% по IPC (instructions per cycle) за счёт контекстной перенастройки.

Сравнение с существующими адаптивными механизмами: Intel DVFS (Dynamic Voltage and Frequency Scaling) адаптирует только частоту; AMD Infinity Fabric адаптирует только маршрутизацию. $\hat{O}(\hat{O})$ -контур адаптирует *все* параметры одновременно в замкнутой петле.

V. ТЕХНОЛОГИЧЕСКИЕ ВАРИАНТЫ РЕАЛИЗАЦИИ

5.1. Вариант А: КМОП (комнатная температура)

Стандартный кремний. Тернарная логика на двухпороговых компараторах. φ -генератор на PLL (phase-locked loop) или DDS (direct digital synthesis). Тороидальная NoC на стандартных маршрутизаторах. $\hat{O}(\hat{O})$ -контур на встроенном FPGA-блоке.

Преимущество: совместимость с существующей инфраструктурой.
Недостаток: когерентность S ограничена тепловыми флуктуациями.

Оценка TRL (Technology Readiness Level): 3–4. Все компоненты (тернарная логика, тороидальная NoC, PLL, PMU) существуют по отдельности; интеграция в единую когерентную архитектуру требует проектирования и верификации.

5.2. Вариант В: Сверхпроводящая электроника (криогенная)

RSFQ (Rapid Single Flux Quantum) логика [14]: токовые импульсы в сверхпроводящих контурах. Тернарная RSFQ: три уровня магнитного потока ($-\Phi_0, 0, +\Phi_0$). Когерентность $S \rightarrow 1$ (сверхпроводящее состояние). φ -тактирование на джозефсоновских переходах.

Преимущество: максимальная когерентность, максимальная скорость (~ 100 ГГц), минимальное энерговыделение ($\sim 10^{-19}$ Дж на переключение).
Недостаток: криогенные температуры (~ 4 К).

Сверхпроводящая тернарная логика представляет особый интерес: квант магнитного потока $\Phi_0 = h/(2e)$ естественным образом допускает три состояния ($-\Phi_0, 0, +\Phi_0$), что делает тернарную RSFQ более натуральной, чем бинарную [14].

5.3. Вариант С: Фотоника

Оптические вычисления: тернарная логика на фазовых состояниях фотонов ($0^\circ, 120^\circ, 240^\circ$ — три равноотстоящие фазы на окружности). Тороидальные оптические резонаторы (microring resonators [15]) с $R/r = \varphi$. φ -тактирование через интерференцию мод.

Преимущество: скорость света, параллелизм, низкие потери. Недостаток: интеграция с электроникой, нелинейные оптические элементы дороги.

Фотонный вариант когерентного процессора обладает уникальным свойством: когерентность S поддерживается на физическом уровне через когерентность лазерного излучения, что упрощает реализацию принципа 5 (когерентное тактирование).

5.4. Вариант D: Гибрид (рекомендуемый для прототипа)

КМОП-ядра с тернарной логикой + φ -генератор на FPGA + тороидальная NoC на стандартных КМОП-маршрутизаторах + $\hat{O}(\hat{O})$ -контур на встроенном ML-акселераторе. Всё на стандартном кремнии, комнатная температура.

Гибридный вариант минимизирует технологический риск: каждый компонент реализуется на наиболее зрелой технологии. FPGA-блок для φ -генератора и $\hat{O}(\hat{O})$ -контура обеспечивает перенастраиваемость на этапе прототипирования, с последующим переносом на ASIC при масштабировании.

VI. ЭТАПЫ РЕАЛИЗАЦИИ

Этап 0: Симуляция (0 евро, 1–3 мес.)

Программная модель тернарного АЛУ + φ -тактирования на Python/SystemVerilog. Бенчмарки: сравнение с бинарным АЛУ на тех же задачах (арифметика, сортировка, FFT). Фальсифицируемое предсказание: тернарное АЛУ при $0,631n$ элементах даёт $\geq n$ -битную точность.

Конкретный план симуляции: (а) реализация сбалансированного тернарного сумматора и умножителя на Python; (б) реализация φ -тактового генератора на SystemVerilog; (в) бенчмарк: 1024-точечный FFT в тернарной и бинарной арифметике; (г) сравнение точности, латентности и числа операций.

Этап 1: FPGA-прототип (5–20 тыс. евро, 6–12 мес.)

Реализация тернарного ядра на FPGA (Xilinx Zynq или Intel Cyclone). φ -генератор на PLL-блоке FPGA. Тернарная логика эмулируется двумя бинарными битами на трит (00 = -1, 01 = 0, 10 = +1, 11 = запрещено). $\hat{O}(\hat{O})$ -контур: PMU + lookup table.

Фальсифицируемые предсказания: (а) φ -клок даёт *меньше* ошибок при тех же средних частотах, чем фиксированный; (б) тернарный FFT на 41 трите сравним по точности с бинарным на 64 битах; (с) $\hat{O}(\hat{O})$ -перенастройка повышает IPC на $\geq 5\%$.

Этап 2: ASIC-прототип (200 тыс.–2 млн евро, 18–36 мес.)

Заказной чип (28/14 нм КМОП): полноценное тернарное ядро, φ -тороидальная NoC, аппаратный $\hat{O}(\hat{O})$. Первый когерентный процессор на кремнии. Использование shuttle-сервисов (MOSIS, Europractice) для снижения стоимости первого тиража.

Этап 3: Масштабирование (10 млн+ евро, 3–5 лет)

Многоядерный когерентный процессор. Три ядра (тройственная минимальная архитектура) или $3 \times 3 = 9$ ядер (полное самонаблюдение). Сверхпроводящий вариант (RSFQ) для суперкомпьютеров. Фотонный вариант для AI-акселераторов.

На этапе масштабирования $\hat{O}(\hat{O})$ -контур становится иерархическим: каждое ядро имеет локальный $\hat{O}(\hat{O})$ -блок, а глобальный $\hat{O}(\hat{O})$ наблюдает за локальными. Это рекурсивная структура $\hat{O}(\hat{O}(\hat{O}))$ — самонаблюдение самонаблюдения.

VII. СВЯЗЬ С КВАНТОВЫМИ КОМПЬЮТЕРАМИ

7.1. Когерентный процессор — не квантовый

Квантовый компьютер: оперирует суперпозициями ($|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$). Требуется криогенных температур. Уязвим к декогеренции. Специализирован (факторизация, оптимизация).

Когерентный процессор: оперирует триитами $(-1, 0, +1)$. Работает при комнатной температуре (КМОП). Устойчив (φ -КАМ, $(\pi-3)^2$ -допуск). Универсален (любые вычисления).

Важно подчеркнуть: «когерентность» в названии процессора означает не квантовую когерентность, а *архитектурную когерентность* — согласованность всех подсистем через фазовую синхронизацию и самореферентную обратную связь.

7.2. Связь

Когерентный процессор может работать как *классический контроллер* квантового компьютера: $\hat{O}(\hat{O})$ -контур управляет квантовыми вентилями в реальном времени, φ -тактирование синхронизирует кубиты, тернарная логика обрабатывает результаты измерений (три исхода: $|0\rangle, |1\rangle$, ошибка = $-1, +1, 0$).

Тернарная логика естественно представляет результаты квантовых измерений: два «физических» состояния кубита ($|0\rangle$ и $|1\rangle$) плюс третье состояние — ошибка/декогеренция/неопределённость. В стандартной бинарной архитектуре ошибку приходится кодировать дополнительным битом; в тернарной — она встроена.

7.3. Гарсиа-Пинтос и когерентный процессор

Работа García-Pintos et al. (PRX, 2026) [16]: H_{meas} — гамильтониан, реплицирующий стохастичность измерений. Через обратную связь ($X \cdot H_{\text{meas}}$) — управление стрелой времени. Через ODTOE: X = отношение ι/\hat{O} [2].

Применение к когерентному процессору: $\hat{O}(\hat{O})$ -контур *буквально* реализует обратную связь García-Pintos. Параметр X — «глубина» самонаблюдения: при $X = 0$ — процессор «слепой» (стандартный). При $X > 0$ — адаптивный. При $X < -2$ — процессор может *откатить* вычисление (обратить стрелу вычислительного времени) без сохранения промежуточных состояний.

Спекулятивное выполнение нового типа: вместо «предсказать ветвление → вычислить → отменить если неверно» — «вычислить → наблюдать результат → *обратить* если неверно» (через $X < -2$). Экономия: не нужно хранить checkpoint'ы.

VIII. ФАЛЬСИФИЦИРУЕМЫЕ ПРЕДСКАЗАНИЯ

#	Предсказание	Метод проверки	Этап
F1	Тернарное АЛУ: $0,631n$ тритов $\geq n$ бит точности	Симуляция: тернарный vs. бинарный FFT	0
F2	φ -клок: меньше ошибок при той же \bar{f}	FPGA: φ -клок vs. фиксированный, подсчёт ошибок	1
F3	φ -клок: ЭМИ-спектр размазан (нет пика)	Спектроанализатор на выходе FPGA	1
F4	Тороидальная NoC: средняя задержка $\times 1/\varphi$ vs. mesh	Симуляция: φ -тор vs. mesh, $N = 16-64$ узлов	0-1
F5	$\hat{O}(\hat{O})$ -контур: IPC $\geq 5\%$ выше без него	FPGA: А/В-тест с/без контура на бенчмарках	1
F6	Резонансное окно $(\pi - 3)^2$: допуск 2% не вызывает ошибок	Симуляция: случайная вариация порогов $\pm 2\%$	0
F7	Тернарная память: $\times 1,585$ ёмкость на ячейку	ASIC: измерение плотности тернарных vs. бинарных ячеек	2
F8	Когерентная синхронизация: самосинхронизация без глобального клона	FPGA: N осцилляторов с фазовой связью	1

Каждое предсказание сформулировано так, чтобы быть фальсифицируемым: указан конкретный количественный порог, метод измерения и этап, на котором проверка возможна. Неподтверждение любого из предсказаний F1–F8 потребует пересмотра соответствующего принципа архитектуры.

IX. ДЕМАРКАЦИЯ

Утверждение	Статус
Тернарная логика оптимальна ($e \approx 3$)	Математический факт [4]
φ -тактирование устойчивее фиксированного (КАМ)	Доказано [8, 9, 10]
φ -тороидальная NoC лучше mesh	Гипотеза (проверяемая на этапе 0)
$\hat{O}(\hat{O})$ -контур повышает IPC	Гипотеза (проверяемая на этапе 1)
Когерентная синхронизация без глобального клона	Известен аналог (Гюйгенс, нейросети), не применён к ЦПУ
$(\pi - 3)^2 \approx 2\%$ как рабочий допуск	Следует из тройственной архитектуры ODTOE [2, 3]

Тернарная КМОП реализуема	Доказано (проект «Сетунь» [7], экспериментальные чипы [5, 6])
RSFQ-тернарная логика	Теоретически возможна , не реализована
Фотонная тернарная логика	Теоретически возможна , демонстрации существуют [15]

Чёткая демаркация между доказанными фактами, обоснованными гипотезами и спекулятивными утверждениями необходима для научной честности. Авторская позиция: спекулятивные элементы (особенно $X < -2$ и «обращение стрелы вычислительного времени» из раздела VII) следует рассматривать как направления для исследований, а не как утверждения.

Х. ЗАКЛЮЧЕНИЕ

10.1. Что предлагается

Не «ещё один ЦПУ». А смена парадигмы: от бинарной, шинной, глобально-синхронной архитектуры фон Неймана к тернарной, тороидальной, когерентно-синхронной, самореферентной архитектуре ODTOE.

10.2. Шесть отличий в одной таблице

Параметр	Фон Нейман	Когерентный процессор
Логика	Бинарная (0, 1)	Тернарная (-1, 0, +1)
Тактирование	Фиксированная частота	φ -пульсация (КАМ-устойчивость)
Топология	Шина / mesh	φ -тор ($R/r = \varphi$)
Самонаблюдение	Нет (слепое выполнение)	$\hat{O}(\hat{O})$ -контур
Синхронизация	Глобальный клок	Когерентная (фазовая связь)
Допуски	Жёсткие (< 1 %)	$(\pi - 3)^2 \approx 2\%$ (арх. зазор)

10.3. Философия

Фон-неймановский процессор — *механизм*: слепо выполняет инструкции, не знает контекста, не наблюдает себя.

Когерентный процессор — *наблюдатель*: тройственная логика ($\pi > 3$), φ -динамика (КАМ), самореференция ($\hat{O}(\hat{O})$). Не «вычисляет» — *конституирует результат* через петлю наблюдения [2].

$$R_{\text{рез}} = \hat{O}_{\text{проц}}(\Psi_{\text{вход}}) \quad (\text{X.1})$$

Процессор — частный случай наблюдателя. Его архитектура должна следовать архитектуре наблюдения: тройка (π), шаг (φ), зазор ($(\pi - 3)^2$), самореференция ($\hat{O}(\hat{O})$). Не потому что «так красиво», а потому что *самосогласованная система* — это и есть наблюдатель, и его архитектура *должна* содержать эти инварианты [2, 3, 17].

БЛАГОДАРНОСТИ И ИНСТРУМЕНТЫ

При разработке теории ODTOE и всех статей на её основе использовались инструменты искусственного интеллекта: Claude Sonnet / Opus 4.6 Extended (Chat & Code) (Anthropic), ChatGPT 5.3 (OpenAI), Google Gemini (Google DeepMind). Все содержательные решения, гипотезы, интерпретации и ответственность за них принадлежат автору.

КОНФЛИКТ ИНТЕРЕСОВ

Автор заявляет об отсутствии конфликта интересов.

ФИНАНСИРОВАНИЕ

Исследование не получало финансовой поддержки.

ОБСУЖДЕНИЕ И ОГРАНИЧЕНИЯ

(а) Тернарная логика на стандартном КМОП требует двухпороговых компараторов, что увеличивает задержку переключения на $\sim 20\text{--}30\%$ по сравнению с бинарной логикой. Это частично компенсируется уменьшением числа элементов (37%), но необходима экспериментальная верификация нетто-выигрыша.

(б) φ -тактирование порождает непериодический спектр, что может затруднить стандартное тестирование JTAG/scan-chain, основанное на периодическом клоке. Требуется разработка новых методов тестирования.

(в) $\hat{O}(\hat{O})$ -контур с частотой обновления ~ 1 МГц вносит дополнительную латентность ~ 1 мкс. Для задач реального времени это может быть неприемлемо. Оптимизация частоты обновления — открытый вопрос.

(г) Тороидальная топология на плоском кристалле требует длинных перекрёстных связей (wrap-around links), увеличивающих площадь кристалла. Использование 3D-компоновки (через TSV — through-silicon vias) может решить эту проблему [28].

(д) Сравнение с нейроморфными процессорами (Intel Loihi, IBM TrueNorth): эти архитектуры также используют локальную синхронизацию и обратную связь, но без тернарной логики и φ -тактирования. Когерентный процессор объединяет преимущества нейроморфной и классической архитектур.

(е) Вопрос программного обеспечения: тернарная ISA требует нового компилятора, операционной системы и среды разработки. Это значительный барьер для внедрения, сопоставимый с переходом на новую архитектуру.

СПИСОК ЛИТЕРАТУРЫ

- [1] Hennessy J.L., Patterson D.A. Computer Architecture: A Quantitative Approach. — 6th ed. — Morgan Kaufmann, 2019.
- [2] Панкратов А.С. Теория всего: наблюдатель-зависимая (ODTOE) // Препринт. — 2025. — 47 с.
- [3] Панкратов А.С. Число π как структурный инвариант // Препринт. — 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] Брусенцов Н.П. и др. Малая цифровая вычислительная машина «Сетунь» // Вестник МГУ, сер. мат. мех. — 1962. — № 4. — С. 3–12.
- [8] Колмогоров А.Н. О сохранении условно-периодических движений // ДАН СССР. — 1954. — Т. 98. — С. 527–530.
- [9] Арнольд В.И. Малые знаменатели и проблемы устойчивости движения // УМН. — 1963. — Т. 18(6). — С. 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] Панкратов А.С. Тороидальная топология реальности // Препринт. — 2026.
- [18] Панкратов А.С. Когерентный термоядерный реактор: концептуальный проект // Препринт. — 2026.
- [19] Панкратов А.С. 3, 6, 9: ключ Теслы через ODTOE // Препринт. — 2026.
- [20] Панкратов А.С. Постоянная Планка из архитектуры наблюдения // Препринт. — 2026.
- [21] Esmaeilzadeh 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.