Technical Research Report: The Transition to Mandated Ternary Architectures via Memristive Hysteresis

1. Abstract

This report investigates the architectural and economic feasibility of transitioning from dominant binary CMOS computing to a "Mandated Ternary" paradigm. The proposed system is defined by a physically stable, non-volatile third logic state ("Null" or "Balance") engineered into memristive devices via hysteresis. This third state is not a software convention but a hardware-enforced checkpoint that gates computational actions, providing a novel mechanism for safety, security, and auditability. The report's central thesis is that this paradigm offers a discontinuous advantage over incremental binary scaling by directly mitigating the most pressing bottlenecks in advanced semiconductor nodes: interconnect delay, the memory wall, and power density. We provide a rigorous analysis of the device physics, focusing on Tantalum Oxide (TaOx) RRAM as a primary example and comparing it to other memristive technologies like HfOx, PCM, and MTJs. The analysis quantifies the substantial "emulation tax"—over 15x in energy and 5x in latency for a simple operation—incurred when simulating ternary logic on binary hardware, thereby motivating the need for native implementations. We further argue that the rise of agentic AI, with its need for verifiable, enforceable hesitation, serves as a critical catalyst for this architectural shift. The report concludes with a roadmap outlining the key milestones required to achieve industry-standard viability for three candidate architectures by 2027.

2. Executive Summary

The semiconductor industry faces a convergence of physical and economic limits as CMOS scaling approaches its end. The "Mandated Ternary" architecture, enabled by memristive hysteresis, presents a viable path forward by fundamentally rethinking the basis of digital logic. This report provides a comprehensive technical analysis of this transition.

Core Claim: The "Mandated Ternary" paradigm, which introduces a physically stable, non-volatile third state ("Null") as a hardware-enforced authorization mechanism, provides a **discontinuous architectural advantage** over incremental binary scaling. This advantage is most pronounced in mitigating the **interconnect bottleneck**, overcoming the **memory wall**, and enabling **verifiable safety** for next-generation agentic AI systems.

Key Findings:

1. **Device Physics Feasibility:** Tantalum Oxide (TaOx) RRAM and other memristive technologies can be engineered to support a stable, intermediate resistance state. This "Null" state is achieved by controlling the partial formation or rupture of a conductive filament, creating a local energy minimum within the device's hysteresis window. This state is non-volatile and can be reliably sensed, providing the physical basis for the ternary paradigm.
2. **High Cost of Emulation:** Implementing ternary logic on a binary CMOS substrate incurs a significant "emulation tax." A worked numerical example shows that a single ternary multiply-accumulate (MAC) operation emulated on binary hardware consumes **over 15 times the energy** and takes **more than 5 times the latency** of a native binary MAC. This tax, driven by logic gate inflation and memory traffic overhead, fundamentally limits the performance of emulated systems and provides a strong economic incentive for native ternary hardware.
3. **Addressing Critical Bottlenecks:** The Mandated Ternary architecture directly addresses the most severe limitations of advanced CMOS nodes. By enabling dense, non-volatile memory and compute-in-memory (CiM) fabrics, it drastically reduces data movement, which is the dominant consumer of energy. This provides a solution to the **interconnect delay** and **memory wall** problems that are straining binary architectures.
4. **Agentic AI as a Catalyst:** The emergence of agentic AI systems, which operate in a closed loop of perception, planning, and action, creates a critical need for a verifiable "hesitation" state. A software flag is insufficient for safety and auditability. A physically stable "Null" state, enforced by a memristive device, provides a tamper-resistant, auditable checkpoint that can gate autonomous actions, making it a crucial enabling technology for safe and reliable AI.
5. **Path to Standardization:** A roadmap to 2027 is proposed, focusing on three candidate architectures: **memristor-based CiM**, **spintronic (MTJ) logic**, and **ferroelectric FETs (FeFETs)**. Achieving "industry standard viability" will require foundry PDK support, robust EDA tool flows, and the development of certified IP blocks, with key milestones in device stability, circuit integration, and system-level benchmarking.

Conclusion: The transition to a Mandated Ternary architecture is not merely an academic exercise but a necessary evolution in computing. The physical limits of binary scaling, combined with the emerging requirements of agentic AI, create a compelling case for this paradigm shift. While significant challenges in device variability, circuit design, and system integration remain, the potential rewards in terms of energy efficiency, performance, and safety are too great to ignore. The path to 2027 will be critical in determining whether this technology can move from the laboratory to the fab and become a cornerstone of post-CMOS logic.

3. Definitions and Scope

This section establishes the foundational terminology and conceptual boundaries for the "Mandated Ternary" computing paradigm. It defines the baseline binary CMOS technology, the nature of memristive systems, the critical role of hysteresis, and the specific meaning of "mandate" as a hardware-enforced mechanism. These definitions are not merely semantic; they form the technical bedrock upon which the subsequent analysis of device physics, circuit design, and system architecture is built. A precise understanding of these terms is essential for evaluating the feasibility and potential advantages of transitioning from a binary to a ternary computing framework.

3.1. Baseline: Binary CMOS

Complementary Metal-Oxide-Semiconductor (CMOS) technology is the incumbent digital logic paradigm, built upon the binary representation of information. Its operation is defined by the controlled switching of MOSFETs (Metal-Oxide-Semiconductor Field-Effect Transistors) to create two distinct, stable logic levels. The performance, power, and reliability of binary CMOS systems are fundamentally constrained by physical principles that become increasingly pronounced at advanced process nodes. Understanding these baseline characteristics is crucial for quantifying the potential benefits and trade-offs of any alternative computing paradigm, including Mandated Ternary.

3.1.1. Logic Levels and Noise Margins

In a standard binary CMOS system, logic states are represented by two distinct voltage levels: a high voltage (V_{DD}) representing logic '1' and a low voltage (V_{SS} or GND) representing logic '0'. The integrity of these states is protected by noise margins, which are the voltage ranges within which a signal is guaranteed to be interpreted correctly by a receiving gate. The high noise margin (NM_H) is the difference between the minimum output voltage for a logic '1' ($V_{OH,min}$) and the minimum input voltage required to be recognized as a logic '1' ($V_{IH,min}$). Similarly, the low noise margin (NM_L) is the difference between the maximum input voltage for a logic '0' ($V_{IL,max}$) and the maximum output voltage for a logic '0' ($V_{OL,max}$). These margins ensure robust operation against noise, process variations, and temperature fluctuations. As CMOS technology scales to smaller nodes, supply voltages (V_{DD}) decrease to prevent device breakdown and manage power density. This reduction in V_{DD} directly compresses the available noise margins, making circuits more susceptible to errors from crosstalk, power supply noise, and other sources of signal integrity degradation. The fundamental binary nature of this system means that any intermediate voltage level is undefined and represents a metastable or invalid state, which can lead to system failure if not resolved.

3.1.2. Switching Energy and Power Dissipation

The energy required to switch a CMOS gate from one state to another is a primary determinant of the overall power consumption of a digital circuit. The dynamic power dissipation, which occurs during logic transitions, is dominated by two components: capacitive switching power and short-circuit power. The capacitive switching energy (E_{switch}) for a single transition is given by the formula $\mathbf{E_{switch} = ½CV_{DD}^2}$, where C is the total load capacitance being driven and V_{DD} is the supply voltage. This energy is consumed every time the output node charges or discharges. The short-circuit power arises from a brief period during the input transition when both the pull-up (PMOS) and pull-down (NMOS) transistors are simultaneously conducting, creating a direct path from V_{DD} to ground. While significant, this component is often smaller than the capacitive switching power. In addition to dynamic power, static power dissipation due to leakage currents (subthreshold leakage, gate leakage, etc.) becomes a major concern at advanced nodes, especially when transistors are in the off-state. The total power consumption is the sum of these components, and managing it is a critical challenge, as it directly impacts battery life in mobile devices and thermal management in high-performance systems. The quadratic dependence of switching energy on V_{DD} is a key driver for voltage scaling, but this is in direct conflict with the need to maintain adequate noise margins.

3.2. Memristive Systems

Memristive systems, or memristors, are a class of passive, two-terminal electronic components whose resistance is not constant but depends on the history of the current and voltage applied to it. This property, known as hysteresis, allows them to "remember" their past state, making them a natural candidate for non-volatile memory and novel computing architectures. While the concept of an ideal memristor was first theorized in 1971, practical devices have emerged more recently, driven by advancements in nanofabrication and materials science. These devices offer a fundamentally different way to store and process information compared to traditional CMOS transistors.

3.2.1. The Ideal Memristor

The ideal memristor is a theoretical circuit element postulated by Leon Chua in 1971 as the fourth fundamental passive circuit element, alongside the resistor, capacitor, and inductor . It is defined by a constitutive relationship between the magnetic flux (φ) and the electric charge (q), where the rate of change of flux with respect to charge is a state-dependent function, M(q), known as the memristance. This leads to a voltage-current relationship where the voltage at any time t is proportional to the product of the memristance and the current. A key characteristic of the ideal memristor is its **pinched hysteresis loop** in the current-voltage (I-V) plane, which passes through the origin. This means that when the applied voltage is zero, the current is also zero, and the device retains its last resistance state. The ideal memristor is a purely passive, non-volatile memory element that can be continuously tuned to any resistance value within its range. However, no physical device perfectly matches this idealized model. Real-world devices, often called memristive systems, exhibit more complex behaviors and are typically better described by a broader class of dynamical systems with memory.

3.2.2. Practical Memristive Devices (ReRAM, PCM, etc.)

Practical memristive devices are typically based on the resistive switching phenomenon in various material systems. The most prominent examples include **Resistive Random-Access Memory (ReRAM)** , **Phase-Change Memory (PCM)** , and spintronic devices like **Magnetic Tunnel Junctions (MTJs)** . ReRAM devices, often based on metal oxides like HfO_x or TaO_x, operate by the formation and rupture of conductive filaments within the insulating material, controlled by the migration of oxygen vacancies or metal ions under an applied electric field . PCM devices utilize materials like $Ge_2Sb_2Te_5$ (GST) that can be rapidly switched between a high-resistance amorphous state and a low-resistance crystalline state using electrical pulses to control heating and cooling . MTJs rely on the quantum mechanical tunneling of electrons through a thin insulating barrier between two ferromagnetic layers, with the resistance depending on the relative orientation of their magnetizations . These devices are not ideal memristors but are classified as memristive systems because they exhibit hysteresis and memory. They are the building blocks for the "Mandated Ternary" paradigm, as their ability to maintain stable intermediate resistance states is the physical basis for the third logic state.

3.3. Hysteresis Window

The hysteresis window is a critical characteristic of memristive devices that enables their use as multi-state memory elements. It is the physical manifestation of the device's memory and is central to the concept of a stable, non-volatile third state. The properties of this window—its width, shape, and stability—directly determine the reliability and performance of a ternary logic system.

3.3.1. Definition and Physical Origin

Hysteresis in memristive devices refers to the phenomenon where the device's response (e.g., current) to an applied stimulus (e.g., voltage) depends on its previous state. When plotted on an I-V curve, this results in a loop rather than a single line. The "hysteresis window" is the region enclosed by this loop. The physical origin of this behavior is tied to the specific switching mechanism of the device. In oxide-based ReRAM, for example, applying a positive voltage above a certain threshold (V_{SET}) causes oxygen vacancies to migrate and form a conductive filament, switching the device from a high-resistance state (HRS) to a low-resistance state (LRS). Conversely, applying a negative voltage above a different threshold (V_{RESET}) can rupture this filament, returning the device to the HRS. The difference between V_{SET} and V_{RESET}, and the fact that the state of the device is path-dependent, creates the hysteresis loop. In PCM, the hysteresis arises from the thermally induced phase transition between the amorphous and crystalline states, which also exhibits different transition temperatures and kinetics depending on the thermal history. This non-volatile memory effect is the foundation for storing information in these devices.

3.3.2. Role in Enabling Multi-State Stability

The hysteresis window is what allows a memristive device to be engineered for multi-level cell (MLC) operation, which is essential for Mandated Ternary. Instead of just using the two extreme states (HRS and LRS) to represent binary '0' and '1', the continuous nature of the resistance change within the hysteresis loop can be exploited. By carefully controlling the amplitude and duration of the programming pulses, the device can be set to one of several intermediate resistance levels. For a ternary system, this means creating a stable, intermediate state (the "Null" or "Balance" state) between the high and low resistance levels. The width of the hysteresis window is crucial here; a wider window provides a larger voltage range for programming and a greater separation between the different resistance states. This separation is critical for reliable readout, as it must be large enough to be distinguished by the sensing circuitry despite the presence of noise and device variability. The stability of these intermediate states over time (retention) and across multiple switching cycles (endurance) is a key challenge and a primary focus of device engineering for ternary logic applications. A robust hysteresis window ensures that these intermediate states are not transient but are physically stable and can be reliably addressed and read.

3.4. The "Mandate" as an Enforcement Mechanism

The term "Mandate" in "Mandated Ternary" is not a philosophical or governance concept but a precise technical term referring to a hardware-enforced mechanism that couples the state of a memristive device to the operational flow of a computing system. This enforcement is what distinguishes the proposed paradigm from simple software-based ternary logic or the use of a third state as a mere flag. The mandate creates a physical, unskippable checkpoint in the compute path.

3.4.1. Hardware-Coupled Authorization Path

The core of the mandate is the creation of an authorization path that is physically and inextricably linked to the ternary state of a memristive device. In a conventional binary system, a software flag (e.g., a bit in a register) can be used to gate an operation. However, this flag is itself just data that can be manipulated or bypassed by software. In a Mandated Ternary system, the third state (e.g., the intermediate resistance of a memristor) is not just data; it is a physical condition that must be met for a specific operation to proceed. For example, a circuit could be designed such that a downstream logic gate or a memory access is only enabled if a specific memristor is in its "Null" state. This could be implemented by using the resistance of the memristor to control the gate voltage of a CMOS transistor in the compute path. If the memristor is not in the correct state, the transistor remains off, and the operation is physically blocked. This creates a hard-wired, non-bypassable authorization mechanism. The transition into and out of this "Null" state can be made to require specific, auditable conditions, effectively creating a physical log of the decision-making process.

3.4.2. Distinguishing from Software Flags or "Unknown" States

The Mandated Ternary paradigm is fundamentally different from how a third state is often used in conventional computing. In digital design, a signal can be in a high-impedance ('Z') state, or a simulation can model an "unknown" ('X') state to represent uninitialized or conflicting values . These are transient or logical constructs. The "Null" state in Mandated Ternary is neither. It is a **stable, non-volatile physical state** of a device, intentionally engineered to be robust and persistent without power . Unlike a software flag, which can be overwritten by a rogue process or a bug, the state of the memristor is a physical property that cannot be altered without applying the specific electrical conditions required for switching. This provides a much higher level of security and reliability. Furthermore, the "Null" state is not a representation of "unknown" or "don't care"; it has a specific, defined semantic meaning within the system's logic, such as "action pending," "authorization required," or "hesitation." This meaning is enforced by the hardware, making the system's behavior more predictable and auditable, which is a critical requirement for applications in safety-critical and secure systems, particularly in the context of agentic AI.

4. Baseline: Binary CMOS Limitations at Advanced Nodes

The relentless pursuit of performance through CMOS scaling, as described by Moore's Law, has encountered a series of fundamental physical and economic barriers. At advanced process nodes (3nm class and beyond), these limitations are no longer distant concerns but are the defining constraints of modern system-on-chip (SoC) design. Understanding these bottlenecks is essential to appreciate why an alternative paradigm like Mandated Ternary is not merely an incremental improvement but a potential necessity for continued progress. The primary challenges include the escalating costs of interconnects, the breakdown of Dennard scaling leading to power and thermal crises, the persistent memory wall, the difficulty of scaling SRAM, and the growing gap between data movement and computation costs.

4.1. Interconnect Delay and Energy (RC Scaling)

While transistors have continued to shrink, the wires that connect them have not scaled at the same rate. The delay of a global interconnect is determined by its RC time constant, where R is the resistance and C is the capacitance. As wires become narrower and taller to maintain low resistance, their capacitance to adjacent wires increases. Conversely, making them wider to reduce resistance increases their area and the overall capacitance of the system. This trade-off means that the RC delay of global wires has not improved significantly with scaling and, in some cases, has worsened. This "interconnect bottleneck" means that the time it takes for a signal to travel across a chip can be a dominant factor in the overall clock cycle, limiting performance. Furthermore, the energy required to charge and discharge these long, capacitive wires is substantial, making data movement a major contributor to the total power consumption of a chip. This problem is exacerbated by wire congestion, as the increasing number of transistors demands an ever-growing number of interconnects, leading to complex routing challenges and further performance degradation.

4.2. Power Density and Thermal Limits (Dennard Scaling Failure)

For decades, the industry benefited from Dennard scaling, which allowed for a proportional reduction in transistor size, voltage, and power consumption, leading to a constant power density. However, this scaling has broken down for several reasons. First, the supply voltage cannot be scaled down indefinitely due to the need to maintain sufficient noise margins and overcome threshold voltage variations. Second, as transistors become smaller, static leakage currents (subthreshold leakage, gate leakage) increase exponentially, leading to significant power consumption even when the transistors are not switching. The failure of Dennard scaling means that power density has been increasing with each new technology node, leading to a "power wall." This has forced designers to limit clock frequencies and to adopt complex power management techniques, such as dynamic voltage and frequency scaling (DVFS) and power gating, to keep the chip within its thermal envelope. The result is that simply adding more transistors no longer guarantees a proportional increase in performance, as the chip would simply overheat.

4.3. The Memory Wall and Bandwidth Bottleneck

The "memory wall" refers to the growing disparity between the speed of processors and the speed of main memory (DRAM). While processor performance has historically improved at a much faster rate than memory performance, this gap has created a significant bottleneck. A processor can execute instructions much faster than it can fetch data from memory, leading to long periods of stalling while waiting for data to arrive. This problem is mitigated to some extent by the use of a memory hierarchy, with multiple levels of caches (L1, L2, L3) that store frequently accessed data closer to the processor. However, as the working sets of applications continue to grow, the effectiveness of caches is diminishing. The memory wall is further compounded by the "bandwidth wall," which refers to the limited rate at which data can be transferred between the processor and memory. The number of pins on a chip package and the speed of the memory bus are physical limitations that are difficult to overcome, creating a bottleneck for data-intensive applications like AI and machine learning.

4.4. SRAM Scaling Challenges (Area, Leakage, Yield)

Static Random-Access Memory (SRAM) has been the workhorse of on-chip memory for decades, used for caches, register files, and other high-speed storage. The standard 6-transistor (6T) SRAM cell is a dense and fast memory element, but it faces significant scaling challenges at advanced nodes. As transistors shrink, the area of the SRAM cell does not scale as aggressively due to the need to maintain a minimum size for the pull-up and pull-down transistors to ensure stability and readability. This means that SRAM is consuming an increasingly large portion of the chip area, limiting the amount of memory that can be integrated on-chip. Furthermore, the leakage power of SRAM increases significantly as transistors shrink, making it a major contributor to the overall static power consumption of the chip. Finally, the increased process variability at advanced nodes makes it difficult to manufacture SRAM arrays with high yield, as small variations in transistor characteristics can lead to read or write failures.

4.5. Data Movement vs. Compute Costs

A key consequence of the interconnect and memory walls is that the energy and latency costs of moving data have become a dominant factor in the overall performance and power consumption of a system. In many modern applications, especially in the field of AI, the energy required to move data from memory to the processor and between different levels of the memory hierarchy can be orders of magnitude higher than the energy required to perform the actual computation on that data. This has led to a shift in focus in computer architecture, from optimizing for raw compute performance to optimizing for data movement efficiency. Techniques like near-memory computing, where compute units are placed closer to the memory, and in-memory computing, where computation is performed directly within the memory array, are being actively explored to address this challenge. The goal is to minimize the distance that data needs to travel, thereby reducing the energy and latency associated with data movement.

4.6. Reliability and Process Variability at Extreme Scaling

As transistors are scaled down to the nanometer regime, they become increasingly susceptible to a variety of reliability and variability issues. Process variability, which refers to the random variations in the physical characteristics of transistors during manufacturing, can lead to significant differences in the performance and power consumption of different chips and even different transistors on the same chip. This can make it difficult to design circuits that work reliably across all possible variations. Furthermore, extreme scaling makes transistors more vulnerable to various failure mechanisms, such as electromigration, where the flow of current can cause the metal wires to degrade over time, and soft errors, where high-energy particles can flip the state of a memory cell. These reliability and variability challenges require the use of sophisticated design techniques, such as error correction codes, redundancy, and adaptive voltage scaling, to ensure the correct operation of the system, which adds to the complexity and cost of the design.

5. Device Physics of Memristive Hysteresis and Multi-State Stability

The foundation of a "Mandated Ternary" architecture rests upon the ability of specific solid-state devices to physically manifest and stably maintain three distinct logic states. This capability is rooted in the physics of memristive hysteresis, where the device's resistance is a function of its history of applied voltage or current. Unlike traditional CMOS transistors, which rely on charge-based switching and are fundamentally binary, memristive devices modulate their resistance based on the history of applied voltage or current, creating a continuum of states that can be discretized for logic and memory applications. The stability of these states, particularly a third "Null" or "Balance" state, is not a software abstraction but a direct consequence of underlying physical phenomena such as ionic migration, phase transitions, or spintronic effects. This section provides a rigorous examination of the device physics that enables this behavior, using Tantalum Oxide (TaOx) as a primary worked example and comparing it against other leading memristive technologies. The analysis will cover device stack archetypes, switching mechanisms, the engineering of stable intermediate states, and the critical failure modes that threaten their long-term reliability.

5.1. Tantalum Oxide (TaOx) as a Primary Example

Tantalum oxide (TaOx) based resistive random-access memory (RRAM) has emerged as a leading candidate for implementing multi-state logic due to its well-understood switching mechanism and demonstrated ability to support multiple resistance levels. The device typically consists of a metal-insulator-metal (MIM) stack, where a thin layer of tantalum oxide is sandwiched between two metal electrodes, often a reactive electrode like Ta or Ti and an inert electrode like Pt or Au. The thickness of the oxide layer is a critical parameter, typically in the range of a few nanometers, as it dictates the electric field required for switching and the overall device characteristics. The switching mechanism in TaOx RRAM is primarily attributed to the valence change mechanism (VCM), which involves the migration of oxygen vacancies (Vo) within the oxide layer under the influence of an applied electric field. This migration leads to the formation and rupture of a conductive filament (CF), which connects the two electrodes and determines the device's resistance state. The ability to precisely control the formation and partial dissolution of this filament is key to achieving stable intermediate resistance states, which can be used to represent the "Null" or "Balance" state in a ternary logic system.

5.1.1. Device Stack Archetype (Electrodes, Oxide Layers)

A typical high-performance TaOx memristor is not a simple metal-insulator-metal (MIM) structure but rather a carefully engineered bilayer stack. The archetypal device consists of a bottom electrode (BE), often made of an inert metal like Platinum (Pt), upon which a bilayer of tantalum oxide is deposited. This bilayer is composed of a thin, highly insulating Ta_2O_{5-x} layer (e.g., 5 nm thick) and a thicker, more conductive TaO_{2-x} base layer (e.g., 15 nm thick) . A top electrode (TE), which can be made of materials like Pt, Iridium (Ir), or even reactive metals like Tantalum (Ta) or Titanium (Ti), completes the stack. The choice of electrode material and its deposition conditions are critical. For instance, using an Ir electrode sputtered with a small amount of oxygen (e.g., 2% O_2 in Ar) has been shown to create a smoother TE/oxide interface, which suppresses filamentary switching and promotes more uniform, area-based valence change mechanisms (VCM) that are more conducive to creating stable multi-level states . Furthermore, a thin buffer layer, such as 2 nm of Al_2O_3, is often inserted between the bottom electrode and the TaOx stack to act as a diffusion barrier, stabilizing the device and reducing cycle-to-cycle variability without significantly contributing to the device's resistance or switching properties . This multi-layered approach is essential for controlling the location and nature of the resistive switching, confining it to the thin Ta_2O_{5-x} layer and thereby improving endurance and enabling the formation of multiple, distinct conductive states .

5.1.2. Switching Mechanism (Filament Formation/Rupture, Oxygen Vacancy Migration)

The resistive switching in TaOx devices is governed by the migration of oxygen ions and vacancies under an applied electric field. The process is often described by two competing mechanisms: Electrochemical Metallization (ECM) and Valence Change Memory (VCM) . In ECM, metal cations from an active electrode (e.g., Ag or Cu) migrate into the oxide and form a conductive filament. In VCM, which is more relevant for TaOx with inert electrodes, the switching is caused by the internal redistribution of oxygen vacancies within the oxide layer itself. Specifically, in a bilayer Ta_2O_{5-x}/TaO_{2-x} structure, applying a negative voltage to the top electrode drives oxygen vacancies from the TaO_{2-x} layer into the Ta_2O_{5-x} layer. This accumulation of vacancies creates a conductive path or "filament" through the insulating layer, transitioning the device to a Low Resistance State (LRS) . Conversely, applying a positive voltage repels the vacancies back into the TaO_{2-x} layer, rupturing the filament and returning the device to a High Resistance State (HRS) .

The dominance of either ECM or VCM is highly dependent on the device fabrication, particularly the electrode material and interface quality. A rough interface, like that formed with a standard Pt electrode, creates localized electric field hotspots that favor the formation of thick, percolating filaments characteristic of ECM . This leads to abrupt switching and poor endurance. In contrast, a smooth interface, achieved with oxygen-assisted sputtering of an Ir electrode, promotes a more uniform electric field, leading to a VCM-dominated mechanism where the resistance change is more gradual and distributed across the device area . This VCM-dominated behavior is crucial for achieving the stable, intermediate resistance levels required for multi-level cell (MLC) operation and, by extension, for the "Mandated Ternary" state. In-situ TEM studies have directly observed this process, showing that the application of a negative bias leads to the formation of Ta-rich phases in the Ta_2O_{5-x} layer, confirming the migration of oxygen ions and the creation of a conductive state .

5.1.3. Engineering a Stable Third State (Intermediate Filament, Partial Reset)

The creation of a stable, non-volatile third state in a TaOx memristor hinges on the ability to precisely control the extent of the filament formation or rupture process. Instead of fully forming a thick, low-resistance filament (LRS) or completely rupturing it to return to the high-resistance state (HRS), the "Mandated Ternary" or "Null" state is an intermediate resistance level. This state can be engineered through several methods. One approach is to use a **"partial reset"** operation, where the reset voltage is controlled to only partially oxidize the conductive filament, leaving a thinner or more resistive path. Another method involves controlling the "set" process, where a carefully modulated negative voltage pulse can initiate the formation of a very thin or incomplete filament, again leading to an intermediate state.

The stability of this third state is paramount. The hysteresis loop of the device must be wide enough to provide clear separation between the three states (HRS, LRS, and the intermediate "Null" state) to ensure reliable sensing and immunity to noise. The use of a bilayer Ta_2O_{5-x}/TaO_{2-x} structure is critical here, as it confines the switching dynamics to the thin insulating layer, preventing the formation of overly thick and unstable filaments that are difficult to partially rupture . Furthermore, the choice of electrode material plays a significant role. As demonstrated with Ir electrodes, promoting a VCM-dominated switching mechanism leads to more gradual and controllable resistance changes, making it easier to "lock in" a specific intermediate resistance value . The robustness of this state is also enhanced by the intrinsic properties of the material; the energy barriers associated with ion migration ensure that the state is non-volatile and can be maintained for long periods without power. Achieving this requires careful optimization of the programming pulses (voltage, duration, and shape) to navigate the device's state space with high precision, a task that presents significant challenges in terms of variability and control circuitry.

5.2. Comparative Analysis of Other Device Families

While TaOx presents a compelling case for ternary logic, it is essential to evaluate it against other emerging memory technologies that also exhibit multi-state potential. Each device family offers a unique combination of physical mechanisms, performance characteristics, and integration challenges. A comparative analysis provides a broader perspective on the landscape of post-CMOS logic and helps to identify the most promising path forward for "Mandated Ternary" architectures. This section will briefly examine Hafnium Oxide (HfOx) RRAM, Titanium Oxide (TiOx) RRAM, Phase Change Memory (PCM), Spintronic devices (MTJ), and Ferroelectric FETs (FeFET). The goal is to understand their respective strengths and weaknesses in the context of creating a reliable, physically-enforced third state.

5.2.1. Hafnium Oxide (HfOx) RRAM

Hafnium oxide (HfOx) is another leading material for RRAM, often considered a primary competitor to TaOx. HfOx-based devices typically exhibit fast switching speeds (sub-nanosecond), low operating voltages, and high endurance, with some reports exceeding 10^12 cycles . The switching mechanism is similar to TaOx, involving the formation and rupture of oxygen vacancy filaments. However, HfOx devices often require a "forming" step, an initial high-voltage stress to create the first conductive filament, which can introduce variability. To mitigate this and improve uniformity, bilayer structures are common, such as HfOx/AlOx or HfOx/TiO2 . These stacks can also enable multi-level cell (MLC) behavior by providing better control over the filament formation process. For instance, a HfOx/AlOx device was shown to have stable resistance states with a retention time of over 10^8 seconds at room temperature, demonstrating its non-volatile nature . The primary challenge for HfOx is managing variability and ensuring uniform switching across large arrays, but its technological maturity makes it a strong contender for ternary logic.

5.2.2. Titanium Oxide (TiOx) RRAM

Titanium oxide (TiOx) was one of the first materials used to demonstrate memristive behavior and remains an important material for research. TiOx devices can be fabricated using solution-based methods, which are low-cost and flexible, but may introduce more defects and variability compared to vacuum-based deposition techniques . The switching mechanism is again based on oxygen vacancy migration and filament formation. One advantage of TiOx is its potential for high scalability; studies have shown that as the device area is scaled down from 50x50 μm² to 200x200 nm², the switching becomes more uniform and reliable, likely due to the suppression of multi-filament formation . However, TiOx devices generally suffer from lower endurance and retention compared to HfOx and TaOx. For example, one study noted that TiOx devices were more prone to degradation at elevated temperatures . Despite these challenges, the low cost and potential for flexible electronics make TiOx an interesting material for specific niche applications.

5.2.3. Phase Change Memory (PCM)

Phase Change Memory (PCM) is a non-volatile memory technology that operates on a different principle than oxide-based ReRAM. PCM devices use a chalcogenide material (e.g., Ge2Sb2Te5 or GST) that can be rapidly switched between a high-resistance amorphous state and a low-resistance crystalline state. This phase transition is induced by Joule heating from an electrical pulse. A short, high-amplitude pulse melts the material, which then quenches into the amorphous state (RESET). A longer, lower-amplitude pulse heats the material to a temperature below its melting point, allowing it to crystallize (SET). PCM offers several advantages, including very fast switching speeds (nanoseconds), high endurance (10^8-10^9 cycles), and excellent retention. Furthermore, the resistance of the material can be controlled in a continuous manner by partial crystallization, making it highly suitable for multi-level cell (MLC) storage and analog computing. However, PCM requires higher programming currents than oxide-based RRAM, which can lead to higher power consumption and thermal crosstalk in dense arrays.

5.2.4. Spintronic Devices (MTJ)

Spintronic devices, such as Magnetic Tunnel Junctions (MTJs), represent another class of non-volatile memory and logic devices. An MTJ consists of two ferromagnetic layers separated by a thin insulating barrier. The resistance of the device depends on the relative orientation of the magnetizations of the two ferromagnetic layers. When the magnetizations are parallel, the resistance is low; when they are anti-parallel, the resistance is high. The state of the device can be switched by various mechanisms, including spin-transfer torque (STT) or spin-orbit torque (SOT). MTJs offer very fast switching, high endurance, and excellent retention. They are the basis for Magnetoresistive RAM (MRAM), which is already being commercialized. For ternary logic, an intermediate resistance state could be engineered by creating a non-collinear magnetization state in the free layer, although this is an active area of research. The main challenges for MTJs are the relatively high write energy and the complexity of integrating magnetic materials into a CMOS process flow.

5.2.5. Ferroelectric FETs (FeFET)

Ferroelectric FETs (FeFETs) are a novel type of transistor that incorporates a ferroelectric material into the gate stack. The ferroelectric material has a spontaneous polarization that can be switched by an applied electric field. This polarization state is non-volatile, meaning it is retained even after the electric field is removed. The polarization state of the ferroelectric material modulates the threshold voltage of the transistor, effectively creating a device with two or more stable states. For ternary logic, the key is to create a stable intermediate polarization state. This can be achieved by carefully controlling the programming voltage and pulse duration, or by using a multi-domain ferroelectric material where different domains can be switched independently. FeFETs offer several potential advantages for ternary logic, including low power consumption, high switching speed, and excellent scalability. They are also highly compatible with CMOS technology, as they can be fabricated using standard CMOS processes with the addition of a ferroelectric layer. However, FeFETs are still a relatively new technology, and there are several challenges that need to be addressed before they can be widely adopted for commercial applications. These challenges include material optimization, device reliability, and the development of suitable circuit architectures that can fully exploit the unique properties of FeFETs.

5.3. Comparative Table of Device Families

To provide a clear, at-a-glance comparison of the candidate device families for implementing Mandated Ternary logic, the following table summarizes their key performance metrics. It is crucial to note that these values are representative of the current state-of-the-art and can vary significantly based on specific material compositions, device structures, and fabrication processes. The data is compiled from recent literature and review articles, with an emphasis on characteristics relevant to logic applications, such as endurance, state stability, and energy efficiency. The table serves as a quantitative foundation for evaluating the trade-offs between different technologies and identifying the most promising candidates for further development towards a standardized ternary architecture. The "Integration Maturity" and "CMOS Compatibility" columns are particularly important for assessing the feasibility of integrating these novel devices into a conventional semiconductor manufacturing flow.

Device Family	Retention (at 85°C)	Endurance (Cycles)	State Variability (σ/μ)	Write Energy (pJ/op)	Read Energy (pJ/op)	Operating Voltage (V)	Integration Maturity	CMOS Compatibility

TaOx RRAM	>10 years	>10^9	Low to Medium	0.1 - 10	<0.01	0.5 - 3.0	High	High (BEOL)
HfOx RRAM	>10 years	10^4 - 10^6	Medium to High	0.1 - 5	<0.01	0.5 - 2.5	High	Very High (BEOL)
TiOx RRAM	<1 year	10^4 - 10^5	High	1 - 50	<0.1	1.0 - 5.0	Medium	High (BEOL)
PCM (GST)	>10 years (3LC)	10^6 - 10^8	Medium	10 - 100	0.1 - 1	1.5 - 3.0	High	Medium (Thermal Budget)
MTJ (STT)	>10 years	>10^15	Low	0.1 - 1	<0.01	0.5 - 1.5	Medium	Medium (BEOL, Magnetic Materials)
FeFET	>10 years (assumed)	10^6 - 10^8	Moderate	Very Low (fJ/op)	Very Low (fJ/op)	0.5 - 2.0	Low	Very High (FEOL)

5.3.1. Retention, Endurance, and Variability

The stability of the third, "Null" state is the most critical parameter for a Mandated Ternary system, and it is directly linked to the device's retention and endurance characteristics. Retention refers to the ability of the device to maintain its programmed state over long periods without power, while endurance refers to the number of times the device can be switched before its performance degrades. **TaOx RRAM demonstrates excellent retention**, with some studies indicating stability for over 10 years at elevated temperatures, making it a strong candidate for non-volatile logic . Its endurance is also exceptionally high, with reports of over 10^9 cycles, which is more than sufficient for most logic applications . However, even in TaOx, variability can be an issue, with cycle-to-cycle fluctuations in resistance values potentially affecting the precise definition of the intermediate state. HfOx RRAM also shows good retention but has a more limited endurance window, often failing after 10^4 to 10^6 cycles, which may be a concern for highly dynamic logic operations . Its variability is also generally higher than that of TaOx. TiOx RRAM suffers from poor retention, particularly at high temperatures, which makes it less suitable for applications requiring long-term state stability .

PCM's retention is complicated by the resistance drift phenomenon. While the crystalline state is very stable, the amorphous state drifts over time. However, as previously mentioned, a three-level cell (3LC) PCM design can mitigate this issue by eliminating the most drift-prone state, enabling non-volatile retention for over a decade . The endurance of PCM is typically in the range of 10^6 to 10^8 cycles, which is adequate for many applications but lower than that of RRAM. MTJs are the clear leaders in endurance, with the potential for over 10^15 cycles, and they also exhibit excellent retention. Their variability is also relatively low, making them a very reliable technology. However, the challenge for MTJs lies in creating a true, stable intermediate state at the device level, as their primary switching mechanism is inherently binary. The variability of the intermediate state, if achieved, would be a key area of research.

5.3.2. Write/Read Energy and Operating Voltage

The energy consumption of the memory element is a critical factor in the overall power budget of a ternary logic system. Write energy, in particular, can be a significant overhead, especially if the logic operations require frequent state changes. **TaOx and HfOx RRAM are known for their low write energies**, typically in the range of 0.1 to 10 pJ per operation, due to the low voltages and currents required for filament formation and rupture. Read energy is even lower, often less than 0.01 pJ per operation, as it only involves applying a small, non-destructive voltage to sense the resistance state. TiOx RRAM generally has higher write energies, partly due to the higher voltages sometimes required for switching. PCM has a higher write energy, typically between 10 and 100 pJ, because the phase transition requires significant Joule heating of the GST material. This can be a major drawback for logic applications where frequent switching is expected. The read energy for PCM is comparable to that of RRAM.

MTJs also offer very low write and read energies, in the range of 0.1 to 1 pJ for writing and less than 0.01 pJ for reading, making them a very energy-efficient technology. The operating voltages for all these technologies are generally low, in the range of 0.5 to 5.0 V, which is compatible with modern CMOS circuitry. The choice of technology will therefore involve a trade-off between write energy, endurance, and state stability. For applications where the state is changed infrequently, PCM might be acceptable, but for highly dynamic logic, the low write energy of RRAM or MTJs would be a significant advantage.

5.3.3. Integration Maturity and CMOS Compatibility

The feasibility of integrating a new memory technology into a standard CMOS manufacturing flow is a major determinant of its commercial viability. RRAM technologies, including TaOx, HfOx, and TiOx, are generally considered to have **high CMOS compatibility**. The switching materials can be deposited using standard techniques like atomic layer deposition (ALD) or sputtering, and the devices can be fabricated in the **back-end-of-line (BEOL)** of the CMOS process, meaning they can be built on top of the already fabricated CMOS logic circuits. This allows for a tight integration of memory and logic without requiring significant changes to the front-end-of-line (FEOL) transistor fabrication process. The thermal budget for BEOL integration is also a key consideration, and most RRAM materials are processed at temperatures that are compatible with the underlying CMOS interconnects.

PCM also has a high degree of integration maturity, with commercial products already available. However, the fabrication of PCM cells can involve higher temperature processes, which may pose a challenge for BEOL integration. The use of GST and other chalcogenide materials also introduces new materials into the CMOS fab, which requires careful process control to avoid contamination. MTJ integration is also possible in the BEOL, but it involves the use of magnetic materials, which can be sensitive to process conditions and may require specialized equipment. The fabrication of high-quality, uniform MTJ stacks with low resistance-area (RA) products is still a significant manufacturing challenge. Ferroelectric FETs (FeFETs), another emerging memory technology, require the integration of ferroelectric materials into the gate stack of the transistor itself, which is a FEOL process and presents a different set of integration challenges. Overall, RRAM technologies appear to have the highest integration maturity and CMOS compatibility for BEOL integration, making them a very attractive option for Mandated Ternary architectures.

5.4. Failure Modes and Stability Threats

The long-term reliability of any memristive technology is a critical concern, and the introduction of a third, intermediate state introduces new potential failure modes that must be carefully understood and mitigated. The stability of this "Null" state is not guaranteed and can be threatened by a variety of physical phenomena, ranging from random thermal fluctuations to systematic degradation mechanisms induced by repeated switching cycles. These failure modes can manifest as noise, state drift, and variability, all of which can compromise the integrity of the ternary logic system.

5.4.1. Noise Sources (Thermal, Read Disturb, RTN)

Several noise sources can threaten the stability of the intermediate ternary state. **Thermal noise** can cause random fluctuations in the device's resistance, potentially pushing it across the threshold into an adjacent state, especially if the energy barrier between states is low. **Read disturb** is another significant issue, where the voltage applied during a read operation is sufficient to cause a small amount of ionic migration, gradually altering the state over many read cycles. **Random Telegraph Noise (RTN)** , caused by the trapping and de-trapping of charge carriers at defect sites in the material, can lead to discrete resistance fluctuations, which can be particularly problematic for sensing the intermediate state. These noise sources are inherent to the physics of the devices and must be managed through careful circuit design, such as using low-voltage read schemes and robust sense amplifiers.

5.4.2. State Drift and Degradation Mechanisms

State drift is a phenomenon where the physical properties of the device change over time, causing its resistance to drift away from its programmed value. This is a particularly well-known issue in PCM, where the amorphous phase is not perfectly stable and tends to crystallize slowly over time, leading to a decrease in resistance . This "resistance drift" can cause the intermediate state to drift into the crystalline state, leading to data corruption. In oxide-based RRAM, drift can be caused by the slow relaxation of defects or the diffusion of oxygen ions after a programming operation. These degradation mechanisms can be accelerated by temperature and cycling, and they pose a significant long-term reliability challenge. Mitigation strategies, such as periodic refreshing or the use of drift-tolerant encoding schemes, are essential for practical applications .

5.4.3. Cycle-to-Cycle and Device-to-Device Variability

Variability is one of the most significant challenges for all emerging memory technologies. **Cycle-to-cycle (C2C) variability** refers to the fact that the resistance of a device after a programming operation is not perfectly repeatable. Each time the device is switched, the final resistance value will be slightly different, following a statistical distribution. **Device-to-device (D2D) variability** refers to the fact that two identical devices fabricated on the same wafer will have different electrical characteristics. This is due to the inherent randomness of the fabrication process at the nanoscale, such as variations in film thickness, grain boundaries, and the number and location of defects. For a ternary system, this variability means that the resistance distributions for the three states will overlap, making it difficult to distinguish between them reliably. This requires the use of large sensing margins, which can reduce the number of available states, or sophisticated statistical sensing and error correction techniques.

6. "Mandated Ternary" Device Requirements and State Definition

The "Mandated Ternary" paradigm is defined by the physical properties and operational semantics of its third state. This section specifies the requirements for the memristive device that will host this state, the conditions under which state transitions occur, and the circuitry needed to reliably read it out. The goal is to move beyond a conceptual understanding and define a concrete set of specifications that a device must meet to be considered a "Mandated Ternary" element. This includes the physical representation of the "Null" state, its stability requirements, the traceability of its transitions, and the design of robust sensing circuits.

6.1. Defining the "Null" or "Balance" State

The "Null" or "Balance" state is the cornerstone of the Mandated Ternary paradigm. It is not a logical placeholder but a distinct, physical state of a memristive device, engineered to be stable and non-volatile. Its properties must be carefully defined to ensure it can serve its dual purpose as a data value and an authorization token.

6.1.1. Physical Representation (Intermediate Resistance)

The "Null" state is physically represented as an **intermediate resistance level** within the hysteresis window of a memristive device. This resistance value, R_{Null}, must be clearly distinguishable from the device's High Resistance State (R_{HRS}) and Low Resistance State (R_{LRS}). For example, if R_{HRS} is in the megaohm range and R_{LRS} is in the kilohm range, R_{Null} might be engineered to be in the tens of kilohms. This state is achieved by precisely controlling the device's switching dynamics, for instance, by performing a "partial reset" that leaves a partially ruptured or a very thin conductive filament. The physical nature of this state is what allows it to be used as a hardware-enforced gate; the resistance value itself is the signal that enables or disables a computational path.

6.1.2. Stability and Non-Volatility Requirements

For the "Null" state to be a reliable basis for logic and authorization, it must meet stringent stability and non-volatility requirements. **Non-volatility** means that the state must be retained for long periods (e.g., >10 years at 85°C) without the need for a power supply or refresh cycles. This is an intrinsic property of the memristive device's hysteresis. **Stability** refers to the state's resistance to perturbations. The energy barrier separating the "Null" state from the "0" and "1" states must be high enough to prevent spontaneous transitions due to thermal fluctuations (noise) or read-disturb events. The state must also exhibit low drift, meaning its resistance value should not change significantly over time. These requirements place strong constraints on the material properties and device engineering, particularly in managing variability and ensuring a wide, well-behaved hysteresis window.

6.2. State Transition and Trace Conditions

The "Mandate" is not just about the state itself, but about the conditions under which a system is allowed to transition into or out of it. This creates a traceable and auditable record of critical decision points in a computation.

6.2.1. Hysteretic Coupling of Compute and Authorization Paths

The core of the "Mandate" is the **hysteretic coupling** of the compute path and the authorization path. A transition from the "Null" state to an "Action" state (e.g., corresponding to the "1" state) should not be a simple, single-step process. Instead, it should be conditional on a series of checks being met. For example, a system might require that a specific set of input conditions be satisfied, a cryptographic signature be verified, and a human-in-the-loop approval signal be received. Only when all these conditions are met is the specific electrical pulse sequence applied to the memristor to transition it out of the "Null" state. The hysteresis of the device is key here; the transition requires a specific energy input, making it a deliberate and logged event. This coupling ensures that the authorization is not just a software check that can be bypassed, but a physical state change that is a prerequisite for action.

6.2.2. Logging and Anchoring of State Changes

Every transition of the memristive device between its three states represents a critical event that must be logged for auditability. This logging can be implemented at multiple levels. At the lowest level, the electrical pulse used to induce the state change can be monitored and recorded. At a higher level, the state of the device can be read and verified after the transition, and this event can be time-stamped and stored in a secure, append-only log. This creates an **immutable record** of the system's decision-making process. The physical state of the memristor itself serves as an "anchor" for this log. Unlike a software log, which can be altered, the physical state of the device provides a non-repudiable piece of evidence. This is particularly important for safety-critical and financial applications, where the ability to reconstruct the exact sequence of events that led to a decision is paramount.

6.3. Readout and Sensing Circuitry

Reliably distinguishing between three states requires specialized sensing circuitry that is more complex than the simple sense amplifiers used in binary memory. The design of this circuitry is critical for the overall performance and reliability of the ternary system.

6.3.1. Sense Amplifiers for Ternary States

A ternary sense amplifier must be able to compare the resistance of the memristive device against two different reference values to determine if it is in the "0", "1", or "Null" state. This can be achieved with a two-stage sensing scheme. In the first stage, the device's resistance is compared to a first reference (R_{Ref1}) that is set between R_{HRS} and R_{Null}. If the device's resistance is higher than R_{Ref1}, it is in the "0" state. If it is lower, a second stage is triggered, where the resistance is compared to a second reference (R_{Ref2}) set between R_{Null} and R_{LRS}. This two-step process allows for the unambiguous identification of all three states. The design of these sense amplifiers must be robust to noise and process variations, and they must be able to perform the read operation quickly and with low energy consumption.

6.3.2. Reference Cells and Margining Strategies

The reliability of the sensing scheme depends critically on the stability of the reference resistances, R_{Ref1} and R_{Ref2}. These references are typically generated using dedicated reference cells, which are memristive devices that are programmed to specific resistance values and are not used for data storage. To account for process variations and temperature changes, these reference cells are often placed in close proximity to the data array and are designed to track the characteristics of the data cells. **Margining strategies** are also employed to ensure reliable operation. This involves setting the reference values with a sufficient margin from the nominal resistance values of the three states to account for variability and noise. For example, R_{Ref1} would be set to a value that is several standard deviations away from the mean of the R_{HRS} and R_{Null} distributions. This ensures that even with some overlap in the resistance distributions, the probability of a read error is kept to a minimum.

7. Circuit Primitives and Sensing Margins for Ternary Logic

The transition from binary to ternary logic requires a new set of circuit primitives. These are the fundamental building blocks from which more complex digital systems are constructed. This section explores the design of native ternary logic gates, hybrid memristor-CMOS gates, and the critical issue of sensing margins and noise immunity in a three-state system. The design of these primitives is a key challenge, as they must be efficient in terms of area, power, and speed, while also being robust to the inherent variability of memristive devices.

7.1. Native Ternary Logic Gates

Native ternary logic gates are circuits that directly implement ternary logic functions without the need for binary encoding and decoding. They are designed to operate on three distinct voltage or current levels, corresponding to the three logic states. The design of these gates is a non-trivial task, as it requires a departure from the well-established principles of binary CMOS design.

7.1.1. Ternary Inverters (STI, PTI, NTI)

The simplest ternary logic gate is the ternary inverter. Unlike a binary inverter, which simply inverts the input (0 -> 1, 1 -> 0), a ternary inverter can be defined in several ways. The most common types are the **Standard Ternary Inverter (STI)** , the **Positive Ternary Inverter (PTI)** , and the **Negative Ternary Inverter (NTI)** . The STI inverts all three states (0 -> 2, 1 -> 1, 2 -> 0). The PTI inverts the "0" and "2" states but leaves the "1" state unchanged (0 -> 2, 1 -> 1, 2 -> 0). The NTI inverts the "1" state but leaves the "0" and "2" states unchanged (0 -> 0, 1 -> 2, 2 -> 2). These gates can be implemented using a variety of technologies, including CMOS with multiple threshold voltages, CNFETs, or memristors. The choice of implementation will depend on the specific characteristics of the technology and the performance requirements of the system.

7.1.2. Ternary NAND/NOR Analogs

Ternary NAND and NOR gates are the ternary equivalents of the universal binary gates. A ternary NAND gate produces an output of "2" (the highest logic level) only when all of its inputs are "2". Otherwise, it produces an output of "0" or "1", depending on the specific definition of the gate. Similarly, a ternary NOR gate produces an output of "0" only when all of its inputs are "0". The design of these gates is significantly more complex than their binary counterparts, as they must be able to handle multiple input combinations and produce the correct output for each. These gates can be implemented using a combination of ternary inverters and other logic functions, or they can be designed as a single, monolithic circuit. The development of efficient and reliable ternary NAND and NOR gates is a key step towards building a complete ternary logic family.

7.1.3. Threshold Logic Gates

Threshold logic gates are a generalization of binary logic gates that can be used to implement a wide range of logic functions, including ternary logic. A threshold gate has multiple inputs, each with an associated weight. The gate computes the weighted sum of its inputs and compares it to a threshold value. If the sum is greater than or equal to the threshold, the gate outputs a "1"; otherwise, it outputs a "0". By using multiple thresholds, it is possible to implement a ternary logic function. For example, a gate with two thresholds could produce a "0" if the sum is below the first threshold, a "1" if it is between the two thresholds, and a "2" if it is above the second threshold. Threshold logic gates can be implemented using a variety of technologies, including memristors, which can be used to store the weights and perform the summation.

7.2. Hybrid Memristor-CMOS Logic Gates

A promising approach for implementing ternary logic is to use a hybrid architecture that combines the strengths of memristors and CMOS transistors. In this approach, memristors are used to store the state or to perform analog computation, while CMOS transistors are used for logic switching and signal amplification.

7.2.1. Memristor as a Programmable Resistor

In a hybrid gate, the memristor can be used as a programmable resistor. By setting the memristor to a specific resistance value, it is possible to control the behavior of the gate. For example, in a memristor-CMOS hybrid inverter, the memristor can be placed in series with a CMOS transistor. The resistance of the memristor will determine the voltage at the output node, allowing for the implementation of a ternary inverter. The non-volatile nature of the memristor means that the state of the gate is retained even when the power is removed, which can be useful for low-power applications.

7.2.2. CMOS Transistors for Logic Switching

While memristors are excellent for storing state and performing analog computation, they are not ideal for high-speed logic switching. CMOS transistors, on the other hand, are very good at this. In a hybrid architecture, CMOS transistors are used to perform the high-speed logic operations, while memristors are used to store the configuration or the state of the system. This allows for the best of both worlds: the high density and non-volatility of memristors, and the high speed and reliability of CMOS transistors. The design of these hybrid gates requires careful co-design of the memristor and CMOS components to ensure that they work together effectively.

7.3. Sensing Margins and Noise Immunity

The reliability of a ternary logic system depends critically on its ability to correctly sense the state of the memristive devices. This requires a robust sensing scheme with adequate noise margins.

7.3.1. Defining Noise Margins for Three States

In a binary system, there is a single noise margin that separates the "0" and "1" states. In a ternary system, there are two noise margins: one between the "0" and "Null" states, and another between the "Null" and "1" states. These noise margins are defined as the difference between the minimum output voltage of the higher state and the maximum input voltage of the lower state. For example, the noise margin between "0" and "Null" is the difference between the minimum output voltage for a "Null" state and the maximum input voltage that is still recognized as a "0". These noise margins must be large enough to ensure reliable operation in the presence of noise and process variations.

7.3.2. Impact of Variability on Sensing Reliability

Variability is a major challenge for memristive devices, and it can have a significant impact on sensing reliability. Cycle-to-cycle and device-to-device variability can cause the resistance values of the three states to overlap, making it difficult to distinguish between them. This can lead to read errors, which can propagate through the system and cause it to fail. To mitigate the impact of variability, several techniques can be used. These include the use of error correction codes, the design of more robust sense amplifiers, and the use of adaptive reference voltages that can be adjusted to account for changes in the device characteristics. The development of these techniques is a key area of research for the realization of reliable ternary logic systems.

8. System Architectures

The successful implementation of a Mandated Ternary paradigm requires the development of new system architectures that can fully exploit the properties of ternary logic and memristive devices. This section outlines three candidate architectures: a native ternary logic pipeline, a crossbar compute-in-memory (CiM) fabric, and a ternary neural network accelerator. These architectures represent different approaches to leveraging the benefits of ternary computing, from a general-purpose processor to a specialized accelerator.

8.1. Native Ternary Logic Pipeline

A native ternary logic pipeline is a general-purpose processor architecture that is designed from the ground up to operate on ternary data. This is the most ambitious of the proposed architectures, as it requires the development of a complete ternary instruction set architecture (ISA), a ternary arithmetic logic unit (ALU), and a ternary register file.

8.1.1. Ternary ALU and Register File

The heart of the ternary processor is the ternary ALU. This unit must be able to perform a variety of arithmetic and logical operations on ternary operands, including addition, subtraction, multiplication, and division, as well as logical operations like AND, OR, and NOT. The design of the ternary ALU is a major challenge, as it requires the development of new algorithms and circuits for performing these operations. The register file is another key component. It must be able to store ternary values and provide fast access to them. The design of the register file will depend on the specific memory technology used, but it will likely be based on a memristive memory array.

8.1.2. Control Unit for Ternary Instructions

The control unit of the ternary processor is responsible for fetching, decoding, and executing ternary instructions. This requires a new control logic that is designed to handle the complexities of the ternary ISA. The control unit must be able to manage the flow of data through the pipeline, handle branches and jumps, and interface with the memory system. The design of the control unit is a complex task that will require significant research and development.

8.2. Crossbar Compute-in-Memory (CiM) Fabric

A crossbar compute-in-memory (CiM) fabric is a specialized architecture that is designed to perform computations directly within a memristive memory array. This approach is particularly well-suited for data-intensive applications like machine learning, where the cost of moving data between memory and the processor is a major bottleneck.

8.2.1. Memristor Crossbar Array for In-Memory Computing

The core of the CiM fabric is a memristor crossbar array. In this array, memristors are placed at the intersection of wordlines and bitlines. The memristors can be used to store data, and they can also be used to perform computations. For example, by applying voltages to the wordlines, it is possible to perform a matrix-vector multiplication, where the conductance of the memristors represents the matrix elements and the input voltages represent the vector elements. This allows for the parallel execution of a large number of multiply-accumulate (MAC) operations, which is a key operation in many machine learning algorithms.

8.2.2. Ternary Control Plane for Data Flow

To support the Mandated Ternary paradigm, the CiM fabric must include a ternary control plane. This control plane is responsible for managing the flow of data through the array and for enforcing the "Mandate." The control plane can be implemented using ternary logic gates, and it can be used to gate the access to the crossbar array. For example, a "Null" state in a control memristor could be used to disable a row or a column of the array, preventing any computation from being performed on the data stored in that row or column. This provides a fine-grained control over the computation and allows for the implementation of complex security and safety policies.

8.3. Ternary Neural Network Accelerators

Ternary neural networks (TNNs) are a type of neural network where the weights and activations are restricted to three values: -1, 0, and 1. This makes them very efficient in terms of storage and computation, as the weights can be stored in a single trit and the multiplications can be replaced by simple additions and subtractions. A ternary neural network accelerator is a specialized hardware architecture that is designed to execute TNNs efficiently.

8.3.1. Ternary Weights and Activations

The key feature of a TNN accelerator is its ability to store and process ternary weights and activations. The weights can be stored in a memristive memory array, with each memristor representing a single weight. The activations can be represented by voltage levels, with three distinct levels corresponding to the three states. This allows for a very dense and efficient representation of the neural network, which can lead to significant reductions in memory footprint and power consumption.

8.3.2. Efficient Multiply-Accumulate (MAC) Units

The core of a TNN accelerator is the multiply-accumulate (MAC) unit. In a TNN, the MAC operation is very simple, as the multiplication by a ternary weight can be implemented as an addition, a subtraction, or a no-operation. This allows for the design of very efficient MAC units that are much simpler and faster than their binary counterparts. The MAC units can be implemented using a combination of adders, subtractors, and multiplexers, and they can be arranged in a parallel fashion to perform a large number of operations simultaneously. This allows for the high-performance execution of TNNs, making them a promising solution for edge AI applications.

9. Emulation Tax vs. Native Ternary: Quantified Comparison

The transition from a well-established binary paradigm to a novel ternary one necessitates a rigorous quantification of the costs associated with emulation. "Emulation tax" refers to the overhead—measured in area, energy, latency, and complexity—incurred when implementing ternary logic and data representation on a binary hardware substrate. This section provides a detailed analysis of this tax by examining encoding strategies, logic gate inflation, and system-level overheads. A worked numerical example and a comparative table are used to bound the performance and energy penalties, providing a clear justification for the pursuit of native ternary hardware. The analysis demonstrates that while emulation is a necessary stepping stone for research and early adoption, the associated tax is substantial and fundamentally limits the potential benefits of ternary computing, making a strong case for the development of native hardware as outlined in this report.

9.1. Encoding Strategies for Ternary on Binary Hardware

When ternary logic is emulated on a binary machine, the three distinct states (e.g., -1, 0, +1 in balanced ternary or 0, 1, 2 in unbalanced ternary) must be mapped onto binary representations. This encoding is the primary source of the emulation tax, as it immediately inflates data storage and complicates logic operations. Several strategies exist, each with its own trade-offs in terms of storage efficiency, logic complexity, and performance. The choice of encoding has a cascading effect on the entire system, from memory traffic to the complexity of arithmetic logic units (ALUs). Understanding these strategies is crucial for accurately modeling the overhead of emulation and for appreciating the inherent inefficiencies that native ternary hardware is designed to eliminate. The most common approaches include multi-bit binary encoding and more complex schemes like balanced ternary, which, while mathematically elegant, present their own set of challenges when implemented on a binary substrate.

9.1.1. 2-bit Encoding (One-Hot, Signed Magnitude)

The most straightforward method for representing a ternary value in a binary system is to use multiple bits. A simple **2-bit encoding** can represent four states, allowing for the mapping of three ternary states with one state left unused (or used for error detection). For example, the states {0, 1, 2} could be mapped to binary {00, 01, 10}. This approach immediately **doubles the memory footprint** for storing ternary data compared to a hypothetical native ternary memory cell. Furthermore, logic operations become significantly more complex. A ternary AND gate, for instance, would require a lookup table or a complex combinational circuit to process the 2-bit inputs and produce a 2-bit output, leading to a substantial increase in transistor count and propagation delay compared to a simple binary AND gate. An alternative is a **one-hot encoding**, where each ternary state is represented by a unique bit being set high (e.g., {0, 1, 2} -> {001, 010, 100}). While this simplifies decoding, it is even more wasteful of storage, requiring three bits per trit, and necessitates complex encoding logic to convert from a compact binary form to the one-hot representation for computation. **Signed magnitude representation**, where one bit represents the sign and another the magnitude, is another option, particularly for balanced ternary, but it similarly suffers from increased bit width and the need for specialized logic to handle the sign and magnitude separately.

9.1.2. Balanced Ternary Representation

Balanced ternary, using the digits {-1, 0, 1}, offers unique mathematical advantages, such as the ability to represent negative numbers without a separate sign bit and simplified arithmetic operations (e.g., subtraction is equivalent to addition of a negated number) . However, emulating this on binary hardware introduces its own complexities. A common approach is to use a 2-bit encoding, such as mapping {-1, 0, 1} to {11, 00, 01}. While this is storage-efficient, the logic required to perform arithmetic is non-trivial. For example, a balanced ternary adder must handle carries and borrows in a way that is fundamentally different from a binary adder. Research has shown that while balanced ternary can offer advantages in information density and arithmetic simplicity, its implementation on binary hardware requires specialized circuits that can be complex to design and may not always yield a net benefit due to the overhead of the emulation logic . The promise of balanced ternary is more fully realized when the physical device itself can natively represent the three states, as is the case with certain memristive or CNFET technologies, which can then directly implement the required threshold logic without the need for binary-to-ternary conversion layers .

9.2. Logic Gate Inflation Factor

A critical component of the emulation tax is the "logic gate inflation factor," which quantifies how many binary gates are required to approximate the function of a single ternary gate. This factor directly impacts the area, power, and speed of any emulated ternary system. A simple ternary inverter, for example, has a well-defined truth table that maps {0, 1, 2} to {2, 1, 0}. Implementing this with binary gates requires a combinational logic circuit that can decode the 2-bit input representing the ternary state, perform the inversion logic, and then re-encode the result into a 2-bit output. This process involves multiple binary gates, leading to a significant increase in area and a longer propagation delay compared to a single binary inverter. The inflation is even more pronounced for more complex gates like ternary NAND or NOR, which must process multiple 2-bit inputs to produce a single 2-bit output. This overhead is a fundamental reason why ternary computing has struggled to gain traction on traditional CMOS platforms; the benefits of higher information density per wire are often negated by the increased complexity and cost of the logic gates themselves .

9.2.1. Area and Energy Implications of Emulation

The area and energy costs of emulating ternary logic are substantial. As noted in "The Ternary Manifesto," a ternary computer may require approximately **1.62 times as much logic in its adder** as a comparable binary computer . This increased gate count translates directly to a larger silicon area. Furthermore, the increased number of transistors and the longer signal paths within the emulated logic gates lead to higher dynamic power consumption due to increased capacitance and switching activity. Static power consumption also increases with the number of transistors. While some studies on novel devices like CNFETs have claimed dramatic reductions in power and delay for ternary gates, these are often compared against baseline binary designs that are not optimized for the same technology node or are theoretical projections that may not account for all real-world parasitics and variability . A more conservative and realistic assessment, based on implementing ternary logic with standard CMOS, indicates a significant overhead. For example, a 2-digit ternary adder/subtractor built with CNFET-based ternary gates was shown to consume over **12 times less power** and have a **5 times better power-delay product (PDP)** than a comparable circuit built with previous ternary gate designs, highlighting that even within the ternary domain, design choices have a massive impact on efficiency, and that emulation on binary hardware would likely be even less efficient .

9.2.2. Transistor Count Comparison

The transistor count is a direct measure of the hardware resources required to implement a logic function and serves as a proxy for area and cost. A standard binary inverter can be implemented with just two transistors in a CMOS design. In contrast, an emulated ternary inverter, which must decode a 2-bit input and produce a 2-bit output, would require a significantly larger number of transistors to implement the necessary combinational logic. Research into native ternary gates using emerging technologies provides some insight into the potential complexity. For instance, a proposed CNFET-based ternary NAND gate was designed with a focus on minimizing transistor count and power consumption . While the exact transistor count for an emulated version is not provided, the complexity of the required logic suggests it would be substantially higher than the native CNFET design and vastly higher than a simple binary NAND gate. Similarly, a memristor-CMOS hybrid design for a balanced ternary half adder required **10 transistors and 59 memristors** for one implementation scheme, and **10 transistors and 64 memristors** for another . This illustrates that even in hybrid native-ternary designs, the component count is non-trivial, and a full emulation on binary hardware would likely be even more resource-intensive, further solidifying the argument for native ternary implementations to achieve true efficiency gains.

9.3. Memory Traffic and Control Path Overhead

Beyond the logic gate level, emulating ternary logic on a binary architecture introduces significant overhead in memory traffic and control path complexity. The need to represent each ternary digit (trit) with multiple bits immediately increases the amount of data that must be moved between memory and the processor. This "memory traffic inflation" exacerbates the already critical memory wall problem, where data movement is a major bottleneck and a significant consumer of energy. Furthermore, the control logic of the processor, including the instruction decode unit and the pipeline control, must be modified or augmented to handle the new, more complex instructions required to operate on the encoded ternary data. This can lead to increased branch mispredictions, pipeline stalls, and a general degradation of the processor's ability to exploit instruction-level parallelism, all of which contribute to the overall emulation tax.

9.3.1. Additional Loads/Stores for Third State

The most direct consequence of using multi-bit encoding for ternary states is the increase in memory traffic. If a trit is represented by two bits, a 32-trit value would require 64 bits of storage, **doubling the memory footprint** compared to a 32-bit binary value. This means that every load and store operation for a ternary variable will transfer twice the amount of data over the memory bus. Given that memory access is already a major performance and energy bottleneck in modern systems, this doubling of traffic can have a severe impact. For example, in a data-intensive application like a ternary neural network, the weights and activations are often ternary. Emulating this on a binary machine would require loading and storing these values in their multi-bit encoded form, significantly increasing the pressure on the memory hierarchy (caches, DRAM) and consuming more energy for data movement. This overhead is a fundamental limitation of emulation and cannot be fully mitigated by software optimization alone. Native ternary memory, where each physical cell stores a single trit, would eliminate this source of overhead entirely.

9.3.2. Branch Mispredictions and Pipeline Stalls

The control path of a modern processor is a highly optimized system designed to predict the flow of instructions and keep the execution pipeline full. Emulating ternary logic can disrupt this flow. Ternary operations, especially those involving conditional logic based on the three states, may not map cleanly onto the binary processor's branch prediction mechanisms. For example, a `switch`-like statement based on a ternary variable would have three cases, which might be compiled into a series of binary `if-else` statements. This can lead to less predictable branching patterns and a **higher rate of branch mispredictions**, which cause the pipeline to be flushed, resulting in a significant performance penalty. Furthermore, the complex logic required to emulate ternary arithmetic can lead to longer latency for individual instructions, increasing the likelihood of pipeline stalls due to data hazards (e.g., waiting for the result of a long-latency ternary multiplication). While techniques like predication (using a `select` instruction) can help to reduce the number of branches, they introduce their own overhead, as both sides of the conditional must be evaluated before the result is selected, which can increase power consumption .

9.4. Worked Numerical Example: Emulation Tax Calculation

To provide a concrete, albeit simplified, estimate of the emulation tax, this section presents a worked numerical example comparing the energy and latency of a simple ternary operation when emulated on binary hardware versus a hypothetical native ternary implementation. The example focuses on a single ternary multiply-accumulate (MAC) operation, a fundamental building block in many computing applications, especially in AI accelerators. The goal is to quantify the overhead in terms of energy consumption and execution time, highlighting the key factors that contribute to the emulation tax. The calculations are based on a set of stated assumptions and simplified models, but they serve to illustrate the order-of-magnitude differences that can be expected. This example will consider the costs of logic gate inflation, memory traffic, and control path overhead to arrive at a comprehensive estimate of the tax.

9.4.1. Energy Model ($E = \alpha CV^2$)

The energy consumed by a digital circuit can be approximated by the formula **$E = \alpha CV^2$**, where C is the load capacitance, V is the supply voltage, and α is the switching activity factor. For this example, let's assume a baseline binary CMOS technology with a supply voltage V = 1V. We will compare the energy of a single binary MAC operation with an emulated ternary MAC.

- **Binary MAC:** A simple binary MAC involves a multiplication and an addition. Let's assume this can be implemented with a combined circuit that has an equivalent capacitance of C_bin = 10 fF and a switching activity α = 0.5.

- E_bin_MAC = 0.5 * 10 fF * $(1V)^2$ = **5 fJ**.

- **Emulated Ternary MAC:** Emulating a ternary MAC is far more complex. The logic gate inflation factor is significant. Let's assume it takes approximately **10 times the number of binary gates** to implement a ternary MAC. This increases the equivalent capacitance to C_emu = 100 fF. The switching activity might also be higher due to the complex decoding and encoding logic, so we assume α = 0.7. The supply voltage remains the same.

- E_emu_MAC_logic = 0.7 * 100 fF * $(1V)^2$ = **70 fJ**.

- **Memory Traffic Overhead:** Now, we add the cost of memory traffic. Assume the ternary operands are encoded as 2-bit values. Loading two operands and storing the result involves transferring 6 bits. Let's assume the energy cost per bit transferred over the memory bus is E_bit = 1 pJ/bit.

- E_emu_MAC_memory = 6 bits * 1 pJ/bit = **6 pJ**.

- **Total Emulation Energy:** The total energy for the emulated ternary MAC is the sum of the logic and memory costs.

- E_emu_MAC_total = 70 fJ + 6 pJ = **76 fJ**.

- **Emulation Tax (Energy):** The emulation tax is the ratio of the emulated energy to the binary energy.

- Tax_Energy = E_emu_MAC_total / E_bin_MAC = 76 fJ / 5 fJ = **15.2x**.

This simplified model suggests that the energy cost of emulating a single ternary MAC operation could be more than 15 times higher than a native binary MAC, primarily due to the increased complexity of the logic and the memory traffic overhead.

9.4.2. Latency Model (Pipeline Stages)

Latency is another critical component of the emulation tax. We can model this by considering the number of pipeline stages required for each operation.

- **Binary MAC:** A well-designed binary MAC can often be completed in a single pipeline stage, with a latency of L_bin_MAC = **1 cycle**.
- **Emulated Ternary MAC:** The complex combinational logic required for the emulated ternary MAC will have a much longer propagation delay. This will likely require the operation to be split across multiple pipeline stages. Let's assume it takes **5 cycles** to complete.
- L_emu_MAC_logic = **5 cycles**.
- **Control Path Overhead:** The increased complexity can also lead to pipeline stalls. Let's assume that, on average, the emulated instruction causes a stall 20% of the time, adding an effective 0.2 cycles to its latency.
- L_emu_MAC_stall = **0.2 cycles**.
- **Total Emulation Latency:**
- L_emu_MAC_total = 5 + 0.2 = **5.2 cycles**.
- **Emulation Tax (Latency):**
- Tax_Latency = L_emu_MAC_total / L_bin_MAC = 5.2 / 1 = **5.2x**.

This model indicates that the latency of the emulated operation could be more than five times longer than the native binary operation. This is a direct consequence of the logic gate inflation and the resulting increase in propagation delay.

9.4.3. Best-Case and Worst-Case Bounds

The calculations above represent a plausible average case. The actual emulation tax can vary significantly.

- **Best-Case Scenario:** The best case would occur if the ternary operation is highly optimized and maps well to the underlying binary hardware. For example, if a ternary operation can be implemented using a small number of binary instructions and the data is already in the cache, the overhead would be lower. In this case, the logic inflation might be closer to 5x, and the memory traffic overhead could be negligible. This might reduce the energy tax to around **5-10x** and the latency tax to **2-3x**.

- **Worst-Case Scenario:** The worst case would involve frequent cache misses, high branch misprediction rates, and complex, multi-instruction sequences to implement the ternary logic. The logic inflation could be 20x or more, and the memory traffic could be a major bottleneck. In this scenario, the energy tax could easily exceed **50x**, and the latency tax could be **10x or higher**, especially if the operation involves data-dependent memory accesses.

These bounds highlight the high variability and significant potential cost of emulation, reinforcing the need for native hardware to achieve predictable and efficient ternary computation.

9.5. Comparative Table: Quantifying the Emulation Tax

The following table summarizes the key metrics and provides a comparative analysis of the emulation tax across different aspects of the system. The values are based on the worked example and the qualitative discussions in the preceding sections.

Metric	Binary Baseline (CMOS)	Emulated Ternary (on Binary CMOS)	Native Ternary (Hypothetical)	Emulation Tax Factor (vs. Binary)	Rationale and Supporting Evidence
Data Storage (bits/trit)	1	2 (for 2-bit encoding)	1	**2x**	Multi-bit encoding is required to represent 3 states, doubling memory traffic and storage needs .
Logic Gate Area	1x (normalized)	~10x	~1.5-2x (estimated)	**~10x**	Complex combinational logic is needed to decode, compute, and re-encode ternary states .

Energy per Operation (MAC)	5 fJ (example)	76 fJ (example)	~10-20 fJ (estimated)	**~15x**	Increased gate capacitance and memory traffic dominate energy consumption. The worked example shows a 15.2x tax .
Latency (cycles per MAC)	1 cycle (example)	5.2 cycles (example)	~1-2 cycles (estimated)	**~5x**	Longer propagation delay due to complex logic leads to deeper pipelines or more cycles per instruction .
Control Path Overhead	Baseline	High	Low	**Significant**	Increased branch mispredictions and pipeline stalls due to complex instruction sequences and data-dependent operations .
Memory Bandwidth	Baseline	High	Baseline	**High**	Doubling the memory traffic for ternary data puts significant pressure on the memory hierarchy and bus bandwidth.

This table clearly illustrates that the emulation tax is not a single number but a collection of overheads that affect every level of the system, from data storage to control flow. The tax is particularly severe in terms of energy and area, which are critical constraints in modern semiconductor design. While native ternary hardware would still have some overhead compared to binary (e.g., potentially more complex sensing circuitry), it would eliminate the most significant sources of the emulation tax, such as the logic gate inflation and the memory traffic doubling, thereby unlocking the true potential of ternary computing.

10. The Saint Spot: Bottlenecks and the Architectural Advantage

The "Saint Spot" refers to the specific market gap or technological bottleneck that a new architecture can address more effectively than incremental improvements to the existing paradigm. For Mandated Ternary, this spot is defined by the convergence of several critical limitations in advanced binary CMOS systems. The proposed architecture does not simply offer a marginal improvement; it provides a fundamentally different approach to computation and data storage that can alleviate these bottlenecks in a way that "just better binary" cannot. This section maps the key problems of advanced nodes to the specific mechanisms in a Mandated Ternary system that solve them, and argues why these solutions are not achievable through further binary scaling alone.

10.1. Problem-Mechanism-Advantage Mapping

The value proposition of Mandated Ternary can be understood by mapping the dominant bottlenecks of modern computing to the specific architectural mechanisms that alleviate them. This "problem-mechanism-advantage" framework clarifies why a paradigm shift is necessary and how the proposed system delivers its benefits.

Problem (Bottleneck)	Mechanism in Mandated Ternary	Architectural Advantage
Interconnect Delay & Energy	**Dense, Non-Volatile Memory + Compute-in-Memory (CiM)**	**Eliminates long data movement.** Computation is performed directly within the memory array, drastically reducing the energy and latency associated with traversing long global wires.
Memory Wall / Bandwidth Wall	**High-Density Ternary Storage**	**Increases effective memory density.** Storing 1.58 bits per cell (log2(3)) reduces the number of memory accesses and the pressure on the memory hierarchy, mitigating the bandwidth bottleneck.
Power Density & Thermal Limits	**Low-Switching Energy Devices**	**Reduces dynamic power.** Memristive devices have inherently low switching energies (pJ/op) compared to charging/discharging large capacitive loads in CMOS, lowering overall power consumption.
SRAM Scaling Pain	**Alternative State Storage (Memristors)**	**Replaces 6T SRAM.** A single memristor can replace a bulky 6T SRAM cell, freeing up valuable silicon area and eliminating the leakage power associated with SRAM.

Data Movement vs. Compute Costs	**In-Memory Computing Paradigm**	**Shifts the cost balance.** By making computation cheap and local, the architecture fundamentally changes the trade-offs in system design, prioritizing data locality over raw compute throughput.
Reliability & Variability	**Hardware-Coupled "Mandate"**	**Provides verifiable safety.** The physical "Null" state offers a non-spoofable, auditable checkpoint for safety-critical operations, a feature that cannot be replicated by software on a binary system.

10.1.1. Interconnect Bottleneck → In-Memory Computing

The interconnect bottleneck, driven by the RC delays of long wires, is a fundamental limiter of performance and a major consumer of energy in modern SoCs. **Mandated Ternary addresses this directly through the principle of compute-in-memory (CiM).** By using a dense memristor crossbar array, computation (e.g., matrix-vector multiplication) is performed at the location of the data. The physical laws of the crossbar (Ohm's Law for multiplication and Kirchhoff's Law for summation) are leveraged to execute operations in a massively parallel fashion, without the need to move data across the chip. This eliminates the energy and latency costs associated with global interconnects, providing a discontinuous advantage that is not achievable by simply making binary wires smaller or adding more metal layers.

10.1.2. Memory Wall → Dense, Non-Volatile Storage

The memory wall arises from the growing gap between processor speed and memory bandwidth. **Mandated Ternary helps to mitigate this by increasing the information density of memory.** A single memristive cell can store one of three states, effectively storing log2(3) ≈ 1.58 bits of information. This is a 58% increase in density over a binary cell. Furthermore, the non-volatile nature of memristors means that data is not lost when power is removed, enabling new architectural possibilities like instant-on systems and reducing the need for power-hungry refresh cycles in DRAM. This increased density and non-volatility reduce the pressure on the memory hierarchy and the memory bus, helping to alleviate the bandwidth bottleneck.

10.1.3. Power Density → Low-Switching Energy Devices

The failure of Dennard scaling has led to a power density crisis, where increasing transistor counts lead to unsustainable thermal output. **Mandated Ternary offers a path to lower power consumption through the use of low-switching-energy memristive devices.** The energy required to switch a memristor is typically in the pico-joule range, which is significantly lower than the energy required to charge and discharge the large capacitive loads in a high-performance CMOS circuit. While CMOS logic will still be required for certain functions, the bulk of the computation and storage can be offloaded to the more energy-efficient memristive fabric, leading to a lower overall power consumption for the system.

10.1.4. SRAM Scaling → Alternative State Storage

SRAM, the traditional choice for on-chip caches, is facing severe scaling challenges in terms of area, leakage, and yield. **Mandated Ternary provides a compelling alternative by replacing the bulky 6T SRAM cell with a single memristor.** This provides a dramatic reduction in area, which can be used to integrate more memory or other functionality on the chip. It also eliminates the static leakage power of SRAM, which is a major contributor to the overall power consumption of modern processors. This replacement is not just an incremental improvement; it is a fundamental change in the way that on-chip memory is implemented.

10.2. Why "Just Better Binary" is Insufficient

While incremental improvements to binary CMOS will continue, they are fundamentally insufficient to address the core architectural challenges. The problems of interconnect delay, the memory wall, and power density are not just engineering challenges; they are a consequence of the underlying binary paradigm.

10.2.1. Fundamental Limits of Binary Encoding

The binary representation of information is a fundamental constraint. A single wire can only be in one of two states, which means that a certain amount of information requires a certain number of wires. This limits the information density that can be achieved and contributes to the interconnect and memory bottlenecks. While techniques like multi-level signaling can be used to increase the information density of a single wire, they come at the cost of reduced noise margins and increased complexity. **Mandated Ternary, by introducing a third state at the physical level, provides a more elegant and robust solution to this problem.**

10.2.2. Inability to Address Data Movement Costs

The "just better binary" approach focuses on improving the performance and efficiency of individual components, such as transistors and wires. However, it does not address the fundamental problem of data movement. As long as computation and memory are separate, data will need to be moved between them, and this will continue to be a major bottleneck and a major consumer of energy. **Mandated Ternary, with its emphasis on compute-in-memory, provides a way to break this paradigm and to build systems where data movement is minimized.** This is a fundamental architectural change that cannot be achieved by simply making the existing components better.

11. Agentic AI as Catalyst: Enforced Hesitation and Action Gating

The emergence of agentic AI systems, which can autonomously perceive, plan, and act in the world, represents a new and powerful class of computing applications. However, this autonomy also introduces significant risks, particularly in safety-critical domains. The "Mandated Ternary" paradigm, with its hardware-enforced "Null" state, provides a novel and powerful mechanism for mitigating these risks. This section argues that the need for safe and controllable agentic AI is a key catalyst for the adoption of this new architecture.

11.1. Defining Agentic AI Operationally

Agentic AI is not just about performing a single task, like image classification or language translation. It is about operating in a closed loop, where the system can make its own decisions and take actions to achieve its goals.

11.1.1. Closed-Loop Perception-Planning-Action Cycle

An agentic AI system operates in a **closed-loop perception-planning-action cycle**. In the perception phase, the system takes in information from its environment, using sensors like cameras or microphones. In the planning phase, it uses this information to make a plan to achieve its goals. This may involve reasoning about the state of the world, predicting the consequences of its actions, and choosing the best course of action. In the action phase, it executes the plan, using actuators like motors or speakers. This cycle then repeats, with the system continuously updating its plan based on new information from the environment.

11.1.2. Autonomous Tool Use and Decision-Making

A key characteristic of agentic AI is its ability to use tools and to make decisions autonomously. This means that the system is not just following a pre-programmed set of instructions; it is able to adapt to new situations and to learn from its experiences. For example, an agentic AI system might be given the goal of "booking a flight to New York" and be able to use a web browser to search for flights, compare prices, and make a reservation. This level of autonomy requires a high degree of intelligence and a deep understanding of the world.

11.2. The Role of a Third State in Agentic Systems

The autonomy of agentic AI systems creates a need for a new kind of control mechanism. A simple "go/no-go" decision is not sufficient; there is a need for a state of "hesitation" or "uncertainty" that can be used to gate actions and to ensure that the system does not take any irreversible or harmful actions.

11.2.1. Hesitation and Uncertainty Gating

The **"Null" state in a Mandated Ternary system can be used to represent a state of hesitation or uncertainty.** When the system is not sure what to do, or when it needs more information to make a decision, it can enter the "Null" state. This state can be used to gate the system's actions, preventing it from taking any action until it has resolved its uncertainty. For example, a self-driving car that encounters an ambiguous situation, like a pedestrian standing at the edge of a crosswalk, could enter the "Null" state, slowing down or stopping until it has a clearer understanding of the pedestrian's intentions.

11.2.2. Deferred Action and Escrowed Execution

The "Null" state can also be used to implement **deferred action** or **escrowed execution.** This is a mechanism where an action is prepared but not executed until a certain condition is met. For example, an agentic AI system that is authorized to make a financial transaction could prepare the transaction but not execute it until it has received a confirmation from a human supervisor. The "Null" state of a memristive device could be used to gate the execution of the transaction, ensuring that it cannot be completed without the required authorization.

11.2.3. Human-in-the-Loop Release and Policy Checks

The "Null" state provides a natural mechanism for implementing **human-in-the-loop** control. When the system needs to make a decision that is beyond its authority or that requires human judgment, it can enter the "Null" state and request input from a human operator. The system cannot proceed until it has received the required input. This allows for a high degree of human oversight and control, which is essential for safety-critical applications. The "Null" state can also be used to enforce policy checks. For example, a system could be programmed with a policy that prohibits it from taking certain actions. Before taking any action, the system would check to see if it is in compliance with the policy. If it is not, it would enter the "Null" state and wait for further instructions.

11.3. Architecture for Action Authorization via Ternary Gate

The "Mandate" can be implemented in hardware by using a ternary gate that is controlled by the state of a memristive device. This gate can be used to authorize or deny actions based on the state of the system.

11.3.1. "Null" as a Mandatory Intermediate State

In this architecture, the **"Null" state is a mandatory intermediate state** for all actions. Before any action can be taken, the system must first enter the "Null" state. This ensures that all actions are subject to the same level of scrutiny and control. The transition out of the "Null" state is conditional on a set of checks being met, such as policy compliance, human approval, and safety verification. This creates a hard-wired, non-bypassable checkpoint that ensures that the system operates in a safe and controlled manner.

11.3.2. Sequence Diagram of the Action Pipeline

The following sequence diagram illustrates the action pipeline for an agentic AI system using a Mandated Ternary architecture.

```
Actor: Agentic AI System
Actor: Memristive Device
Actor: Human Supervisor
Actor: Environment

AI -> Device: 1. Initialize to 'Null' State
activate Device
Device -> Device: 2. State = 'Null' (Hesitation)
AI -> AI: 3. Perceive Environment
AI -> AI: 4. Plan Action
AI -> Human: 5. Request Authorization
activate Human
Human -> Human: 6. Review Plan & Policy
Human -> AI: 7. Send Authorization Signal
deactivate Human
AI -> Device: 8. Apply 'Authorize' Pulse
Device -> Device: 9. State transitions '1' (Proceed)
AI -> Environment: 10. Execute Action
AI -> AI: 11. Log State Transition & Action
deactivate Device
```

This diagram shows how the physical state of the memristive device is a critical step in the action pipeline, providing a verifiable and auditable record of the authorization process.

11.4. Physical State vs. Software Flag for Safety and Auditability

The use of a physical state for authorization provides several advantages over a software flag, particularly in terms of safety and auditability.

11.4.1. Tamper Resistance and Non-Spoofability

A software flag is just a bit in memory, and it can be altered by a malicious actor or a software bug. A physical state, on the other hand, is a property of a device, and it cannot be altered without applying the specific electrical conditions required for switching. This makes it much more **tamper-resistant and non-spoofable.** An attacker would need to have physical access to the device and the ability to apply the correct voltage pulses to change its state, which is a much more difficult task than simply flipping a bit in memory.

11.4.2. Composability and System-Level Guarantees

The use of a physical state for authorization also provides a basis for **composability and system-level guarantees.** Because the state is a physical property of the device, it can be used to build a chain of trust that extends from the hardware to the software. This allows for the creation of systems that have strong, verifiable security and safety properties. For example, it is possible to prove that a system will not take a certain action unless it has received the required authorization, because the action is physically impossible without the correct state in the memristive device. This provides a level of assurance that is not possible with software-based authorization mechanisms.

12. Roadmap to 2027: Top Three Candidate Architectures

The transition from research to industry-standard technology requires a clear and aggressive roadmap. This section outlines the top three candidate architectures for implementing Mandated Ternary systems and defines the key milestones that must be achieved by 2027 to establish their viability. The focus is on practical, demonstrable progress in device performance, circuit integration, and system-level benchmarks. "Industry standard viability" is defined as the point at which the technology is supported by a major foundry's Process Design Kit (PDK), integrated into commercial Electronic Design Automation (EDA) tool flows, and available as certified IP blocks for system-on-chip (SoC) designers.

12.1. Compute-in-Memory with Memristor Crossbars

This architecture leverages the high density and analog computing capabilities of memristor crossbar arrays to perform computations directly within the memory, making it a prime candidate for AI and machine learning accelerators.

12.1.1. Core Principle of Operation

The core principle is to use the conductance of memristors in a crossbar array to represent the weights of a neural network. By applying input voltages to the rows of the array, a matrix-vector multiplication is performed in a single step, with the output currents on the columns representing the result. This massively parallel operation is extremely energy-efficient and can significantly accelerate the inference phase of neural networks. The "Mandate" can be implemented by using a separate layer of ternary memristors to gate the access to the crossbar, providing a hardware-enforced control plane.

12.1.2. Integration Status and Key Obstacles

The integration of memristor crossbars with CMOS is a mature research area, with many prototypes demonstrated. The key obstacle is **variability and yield**. The analog nature of the computation makes it highly sensitive to device-to-device and cycle-to-cycle variations in the memristors. Achieving the precision required for practical neural networks while maintaining high yield across a large array is a major challenge. Another obstacle is the **cost and complexity of the analog-to-digital converters (ADCs)** required to read out the analog results from the crossbar.

12.1.3. 2026-2027 Milestones and Success Criteria

- **Q2 2026:** Demonstrate a 1Mb memristor crossbar array with ternary control plane achieving **<5% variability** in conductance states and **>10^8 endurance cycles**.
- **Q4 2026:** Deliver a hybrid memristor-CMOS chip demonstrating a **10x improvement in TOPS/W** for a standard neural network benchmark (e.g., ResNet-50) compared to a state-of-the-art GPU.
- **Q2 2027:** Achieve **foundry PDK support** for a baseline memristor technology, including models for variability and reliability.
- **Q4 2027:** Release a certified IP block for a ternary-gated memristor crossbar macro, along with a benchmarking suite and a safety certification plan.

12.2. Spintronic Devices (MTJ-based Logic)

Spintronic devices, particularly Magnetic Tunnel Junctions (MTJs), offer extremely high endurance and low power, making them a strong candidate for logic applications where reliability is paramount.

12.2.1. Core Principle of Operation

MTJ-based logic uses the magnetization state of an MTJ to represent a logic value. The switching between states is achieved via spin-transfer torque (STT) or spin-orbit torque (SOT). For ternary logic, a third state can be engineered by creating a non-collinear magnetization state in the free layer, or by using two MTJs to represent the three states. The "Mandate" can be implemented by using the resistance state of an MTJ to control a CMOS gate, providing a non-volatile, high-endurance authorization mechanism.

12.2.2. Integration Status and Key Obstacles

The integration of MTJs with CMOS is a well-established field, with STT-MRAM already in commercial production. The key obstacle for logic applications is **scalability and integration**

complexity. While MTJs are excellent for memory, using them for high-speed, high-density logic is more challenging. The resistance ratio between the high and low states is relatively small, which can make it difficult to distinguish between multiple states. Furthermore, the fabrication of high-quality MTJ stacks requires specialized processes and materials that may not be compatible with all CMOS fabs.

12.2.3. 2026-2027 Milestones and Success Criteria

- **Q2 2026:** Demonstrate a ternary MTJ logic gate with **>10^{15} endurance** and a **sub-nanosecond switching time**.
- **Q4 2026:** Deliver a small-scale MTJ-based ternary processor (e.g., an ALU) demonstrating **>5x improvement in power-delay product** compared to an equivalent binary CMOS design.
- **Q2 2027:** Develop a process for the **3D integration of MTJ logic layers** with a CMOS control plane.
- **Q4 2027:** Achieve **industry-standard viability** with a PDK for an MTJ-based logic process and a certified IP block for a ternary flip-flop.

12.3. Ferroelectric FETs (FeFET)

FeFETs are a promising emerging technology that offers the potential for extremely low power consumption and high CMOS compatibility, making them an attractive option for low-power, edge AI applications.

12.3.1. Core Principle of Operation

FeFETs use the polarization state of a ferroelectric material in the gate stack to modulate the threshold voltage of a transistor. This provides a non-volatile memory function that can be used to store logic states. For ternary logic, a third state can be created by controlling the polarization to an intermediate level. The "Mandate" can be implemented by using the threshold voltage of the FeFET to control a current path, providing a low-power authorization mechanism.

12.3.2. Integration Status and Key Obstacles

The integration of ferroelectric materials into a CMOS process is a key challenge for FeFETs. The materials and processes are not yet as mature as those for other emerging memory technologies. The key obstacle is **reliability and endurance**. The ferroelectric material can degrade over time, leading to a loss of the stored polarization state. Furthermore, the endurance of FeFETs is not yet as high as that of other technologies, which may limit their use in applications that require frequent switching.

12.3.3. 2026-2027 Milestones and Success Criteria

- **Q2 2026:** Demonstrate a ternary FeFET with **>10^{6} endurance cycles** and a **sub-pJ switching energy**.
- **Q4 2026:** Deliver a FeFET-based ternary memory array demonstrating **>2x density** compared to an equivalent SRAM array.
- **Q2 2027:** Develop a **CMOS-compatible process** for the fabrication of FeFETs in a standard logic fab.
- **Q4 2027:** Achieve **industry-standard viability** with a PDK for a FeFET process and a certified IP block for a ternary memory cell.

12.4. Defining "Industry Standard Viability"

For any of these architectures to be successful, they must be adopted by the semiconductor industry. This requires a clear definition of "industry standard viability," which encompasses several key areas, starting with physical alignment with foundry roadmaps.

12.4.1. Foundry Technology Alignment

The transition to Mandated Ternary is supported by the specific roadmaps of leading semiconductor foundries, which are deploying the necessary physical enablers (Backside Power, 3D Integration) in the 2025–2027 timeframe.

- **TSMC (N2 Node & CoWoS):** TSMC's 2nm node (N2), entering volume production in 2025, utilizes Nanosheet GAA transistors. Crucially, their **CoWoS (Chip-on-Wafer-on-Substrate)** platform allows for the heterogeneous integration of high-density RRAM tiles with N2 logic, decoupling the yield risks of ternary memory from the logic die.
- **Intel (18A & PowerVia):** Intel's 18A node introduces **PowerVia** (Backside Power Delivery). By moving power routing to the back of the wafer, the front-side metal layers are freed up for logic routing and the deposition of high-density memristor crossbars directly above the transistors.
- **Samsung (In-Memory Compute):** Samsung has demonstrated the world's first MRAM-based In-Memory Computing chip, validating the architectural paradigm of processing data directly within the memory array—a cornerstone of the Mandated Ternary approach.

Table 3: Foundry Technology Roadmap Summary

Foundry	Node	Logic Tech	Memory Tech	Target Volume
TSMC	N2 (2nm)	Nanosheet GAA	Embedded RRAM/MRAM	2025/2026

Intel	18A (1.8nm)	RibbonFET + PowerVia	Hybrid Bonding (Foveros)	2H 2025
Samsung	SF2 (2nm)	MBCFET (GAA)	eMRAM In-Memory Compute	2025
GlobalFoundries	22FDX+	FD-SOI	Embedded RRAM	2026

12.4.2. Foundry PDK Support and EDA Flows

The most critical requirement is **foundry PDK support.** This means that a major semiconductor foundry (e.g., TSMC, Samsung, GlobalFoundries) must offer a Process Design Kit (PDK) for the technology... [continue with existing text]

12.4.3. Foundry PDK Support and EDA Flows

The most critical requirement is **foundry PDK support.** This means that a major semiconductor foundry (e.g., TSMC, Samsung, GlobalFoundries) must offer a Process Design Kit (PDK) for the technology. The PDK includes all the necessary design rules, device models, and verification decks that are required to design and manufacture a chip using the technology. Without PDK support, it is impossible for designers to create commercial products. In addition, the technology must be supported by the major EDA tool vendors (e.g., Cadence, Synopsys, Siemens EDA), so that designers can use their existing design flows to create and verify their chips.

12.4.4. IP Blocks and Benchmarking Suites

To make it easier for designers to adopt the new technology, it is essential to provide a library of **pre-designed and pre-verified IP blocks.** These IP blocks can include basic logic gates, memory arrays, and more complex functional units, such as processors or accelerators. The availability of these IP blocks can significantly reduce the time and cost of developing a new chip. In addition, it is important to develop a set of **standardized benchmarking suites** that can be used to evaluate the performance, power, and area of the new technology. This will allow for a fair and accurate comparison of the new technology with existing technologies.

12.4.5. Safety Certification Regimes

For applications in safety-critical domains, such as automotive or aerospace, it is essential to have a **safety certification regime** in place. This involves developing a set of standards and procedures for testing and certifying that the technology is safe to use. This can be a long and expensive process, but it is a necessary step for the adoption of the technology in these markets. The development of a safety certification plan should be a key part of the roadmap for any new technology.

12.5. 2026-2027 Milestone Table

The following table summarizes the key milestones for the three candidate architectures over the next two years.

Architecture	Q2 2026	Q4 2026	Q2 2027	Q4 2027
Memristor CiM	<5% variability, >10^8 endurance	10x TOPS/W vs. GPU	Foundry PDK support	Certified IP block & benchmark suite
MTJ Logic	>10^{15} endurance, sub-ns switching	5x PDP vs. CMOS	3D integration process	PDK & certified ternary flip-flop IP
FeFET	>10^6 endurance, sub-pJ energy	2x density vs. SRAM	CMOS-compatible process	PDK & certified ternary memory IP

13. Falsifiability: Predictions and Failure Conditions

A rigorous research report must be falsifiable; it must make specific, testable predictions and clearly state the conditions under which its central thesis would be proven wrong. This section provides a list of such predictions and failure conditions at the device, circuit, and architecture levels. These statements serve as a guide for future research and a benchmark against which the progress of the Mandated Ternary paradigm can be measured.

13.1. Testable Predictions

The following are testable predictions that, if validated, would provide strong support for the central thesis of this report.

13.1.1. Device-Level Predictions

1. A TaOx-based memristor with a bilayer structure will be demonstrated to have a **stable, non-volatile intermediate resistance state** with a retention time of **>10 years at 85°C** and an endurance of **>10^9 cycles**.
2. The resistance distribution of the intermediate state in a ternary memristor will be shown to have a **coefficient of variation (σ/μ) of <10%**, enabling reliable sensing with a bit-error-rate of **<10^{-15}**.
3. A memristive device will be shown to be programmable to **at least 5 distinct, stable resistance levels** with clear separation between the levels, demonstrating the potential for higher-radix logic beyond ternary.

13.1.2. Circuit-Level Predictions

1. A ternary sense amplifier will be designed and demonstrated to correctly distinguish between three resistance states with a **read latency of <10ns** and a **read energy of <1 pJ**.
2. A hybrid memristor-CMOS ternary inverter will be shown to have a **propagation delay of <1ns** and a **power-delay product of <1 fJ**, outperforming an emulated ternary inverter by at least an order of magnitude.
3. A 1Kb array of ternary memristor cells will be demonstrated to operate with a **yield of >95%**, with all cells passing a test of 10^6 read/write cycles.

13.1.3. Architecture-Level Predictions

1. A memristor-based compute-in-memory (CiM) fabric will be shown to achieve a **TOPS/W rating of >100** for a standard neural network inference benchmark, a 10x improvement over state-of-the-art GPUs.
2. A native ternary processor will be shown to execute a set of general-purpose computing benchmarks with an **energy-delay product that is 2x better** than an equivalent binary processor, demonstrating the efficiency of native ternary computation.
3. A Mandated Ternary system will be shown to successfully **gate 100% of unauthorized actions** in a simulated agentic AI environment, with a **0% false positive rate** for legitimate actions.

13.2. Failure Conditions that Disprove the Thesis

The following are failure conditions that, if observed, would significantly weaken or disprove the central thesis of this report.

13.2.1. Device-Level Failure Conditions

1. **Inability to engineer a stable third state:** If it is shown to be physically impossible to create a stable, non-volatile intermediate resistance state in any memristive device with a retention time of **>1 year at room temperature**, the core premise of the report would be invalidated.
2. **Excessive variability:** If the coefficient of variation for the intermediate state in any memristive device is shown to be **>50%**, making it impossible to reliably distinguish between the three states, the practicality of the paradigm would be called into question.
3. **Low endurance for logic:** If the endurance of the intermediate state is shown to be **<10^6 cycles**, it would not be suitable for most logic applications, limiting the scope of the architecture.

13.2.2. Circuit-Level Failure Conditions

1. **High sensing overhead:** If the energy required to sense the ternary state is shown to be **>10 pJ**, it would negate the low-power benefits of the memristive devices.
2. **Low-speed logic:** If the propagation delay of a native ternary gate is shown to be **>10ns**, it would be too slow for high-performance computing applications.
3. **Poor yield:** If the yield of a ternary memristor array is shown to be **<50%**, the technology would not be economically viable for commercial production.

13.2.3. Architecture-Level Failure Conditions

1. **High emulation tax is unavoidable:** If it is shown that the emulation tax for ternary logic on binary hardware is **<2x** in terms of energy and latency, the economic incentive for developing native ternary hardware would be significantly reduced.
2. **No advantage over binary:** If a native ternary processor is shown to have an **energy-delay product that is worse** than an equivalent binary processor on a set of standard benchmarks, the case for the new architecture would be severely weakened.
3. **Mandate is bypassable:** If it is shown that the "Mandate" can be **bypassed by software or physical means** with a success rate of **>1%**, the security and safety claims of the report would be invalidated.

14. Conclusion: What Would Make This Inevitable

This report has presented a comprehensive technical analysis of the transition from binary CMOS to a "Mandated Ternary" architecture, arguing that this paradigm shift is not only feasible but also necessary to overcome the fundamental limitations of current computing systems. The analysis has covered the device physics, circuit primitives, system architectures, and economic drivers for this transition, providing a clear roadmap for the path to 2027.

14.1. Summary of Key Findings

The key findings of this report can be summarized as follows:

- **Device Feasibility:** The physics of memristive hysteresis, particularly in materials like TaOx, allows for the engineering of a stable, non-volatile third state. This provides the physical foundation for the Mandated Ternary paradigm.
- **High Emulation Tax:** The cost of emulating ternary logic on binary hardware is substantial, with a potential energy tax of over 15x and a latency tax of over 5x. This provides a strong economic incentive for the development of native ternary hardware.
- **Architectural Advantage:** Mandated Ternary directly addresses the critical bottlenecks of advanced CMOS nodes, including interconnect delay, the memory wall, and power density, by enabling dense, non-volatile memory and compute-in-memory fabrics.

- **Agentic AI Catalyst:** The rise of agentic AI creates a critical need for a verifiable, hardware-enforced "hesitation" state for safety and auditability, a need that can be uniquely met by the Mandated Ternary architecture.
- **Path to Standardization:** A clear roadmap to 2027 has been outlined, with specific milestones for three candidate architectures: memristor-based CiM, spintronic (MTJ) logic, and ferroelectric FETs.

14.2. Conditions for Widespread Adoption

The widespread adoption of the Mandated Ternary paradigm will depend on the fulfillment of several key conditions:

1. **Demonstration of a Clear Economic Advantage:** The technology must be shown to provide a significant improvement in performance, power, or cost over existing binary solutions for a commercially relevant application.
2. **Achievement of Industry Standard Viability:** The technology must be supported by a major foundry's PDK, integrated into commercial EDA tool flows, and available as certified IP blocks.
3. **Solution to the Variability Problem:** A robust solution to the problem of device variability must be developed, either through improved device engineering or through sophisticated circuit-level techniques.
4. **Development of a Complete Ecosystem:** A complete ecosystem, including design tools, IP libraries, and a skilled workforce, must be developed to support the new technology.

14.3. Final Assessment

The transition to a Mandated Ternary architecture is a bold and ambitious goal, but it is one that is well-supported by the technical analysis in this report. The physical limits of binary scaling, combined with the emerging requirements of agentic AI, create a compelling case for this paradigm shift. While significant challenges remain, the potential rewards in terms of energy efficiency, performance, and safety are too great to ignore. The next few years will be critical in determining whether this technology can move from the laboratory to the fab and become a cornerstone of post-CMOS logic. If the milestones outlined in this report are met, the transition to Mandated Ternary may not just be possible; it may be inevitable.