From NAND to “It from Bit”: Minimal Substrates and the Primacy of Information

Douglas C. Youvan

doug@youvan.com

www.youvan.ai

November 30, 2025

John Wheeler’s slogan “It from Bit” suggests that physical reality, at its most fundamental level, is best understood in informational rather than material terms. Traditionally, this idea has been explored through quantum measurement, black hole thermodynamics, or the role of observers in physics. This paper approaches It from Bit along a different, computational axis: the existence of extremely minimal yet universal substrates for artificial intelligence. If any sufficiently powerful pattern of information-processing can be instantiated on bare NAND gates, one-instruction machines, or even combinatory logic, then the material “stuff” becomes secondary to the informational structure it realizes. Building on the analysis of minimal computational languages and NAND-only universality, the paper argues that intelligence is a paradigmatic “it-from-bit” phenomenon. The same high-level intelligent behavior can be compiled into radically different physical embodiments, all reducible to simple, repeated local transitions on symbolic bits. This substrate fungibility, combined with information-theoretic notions of minimal description, points toward a picture in which physical devices are merely one class of carriers for patterns whose essence is informational. The goal is not to prove Wheeler’s slogan, but to show that the logic of universal computation, minimal languages, and AI provides one of the cleanest and most concrete arguments yet for taking It from Bit seriously.

Keywords: It from Bit, NAND universality, minimal substrates, artificial intelligence, substrate independence, Kolmogorov complexity, Minimum Description Length, Wheeler, information-theoretic physics, universal computation, lambda calculus, one-instruction set computer, combinatory logic, cellular automata, it from qubit.

A collaboration with GPT-5.1-Thinking-Extended. 50 pages.

Outline

1. Introduction: From Universal Gates to It from Bit

The idea that physical reality might be best understood in informational terms has moved from poetic slogan to serious conceptual proposal. John Wheeler's phrase "It from Bit" captures this shift in four words: the suggestion that what we call "stuff" is, at bottom, structured information. Traditional discussions of this idea have grown out of quantum theory, black hole thermodynamics, and the role of observers in physics. They ask whether measurement events, bits of yes/no information, are more fundamental than the continuous fields and particles that older pictures of physics take as primary.

In parallel, but largely separate, the theory and engineering of computation have shown that an astonishing variety of physical systems can implement the same abstract algorithms. From NAND gates and flip-flops to lambda calculus interpreters, from one-instruction machines to cellular automata, we now know that a tiny set of local operations, applied in the right patterns, can emulate any effective procedure. Artificial intelligence sits squarely inside this universality: learning, search, and reasoning can be implemented on top of almost any universal substrate, given enough time and memory.

This paper brings these two strands together. It argues that the existence of extremely minimal yet AI-capable computational substrates provides one of the cleanest, most concrete arguments available for taking It from Bit seriously. If intelligent behavior can be ported from one minimal language to another, and from one physical implementation to another, with the informational pattern preserved even as the underlying "stuff" changes, then matter begins to look like an interchangeable carrier of bits, rather than the other way around. The goal is not to settle metaphysical questions, but to show that minimal universality and AI make Wheeler's slogan far more than a metaphor.

1.1 Wheeler's slogan and the informational lens on reality

Wheeler's slogan condenses a radical proposal. In classical physics, matter and fields are primary; information is a derived property of how we describe them. In an informational reading, that relation is flipped. The state of the world at any moment is treated as a vast configuration of bits (or qubits), and what we call particles, fields, and objects are higher-level regularities in those configurations. The "it" is an emergent pattern; the "bit" is the underlying currency.

This reordering has several motivations. Quantum measurement suggests that discrete outcomes—clicks in detectors, yes/no events—play a central role in how physical reality manifests to observers. Thermodynamic arguments relate entropy to missing information about microstates. Coding-theoretic ideas appear in discussions of black holes and holography. Across these contexts, informational quantities behave more robustly and more universally than many familiar material descriptors.

However, most treatments of It from Bit focus on nature “as given”: on the information content of physical processes we observe, not on the systems we build. Artificial intelligence introduces a complementary angle. Here we are not merely describing reality informationally; we are designing and instantiating large, structured information-processing systems in physical hardware. The link between abstract description and physical realization becomes explicit and controllable.

Seen through this lens, Wheeler’s slogan suggests a new question: if intelligence can be realized on many physically different substrates, all of which can be reduced to simple bit-level dynamics, does that not count as strong evidence that what is essential about intelligence is its informational structure rather than its material carrier? Minimal universal languages make that question sharper by stripping away everything that is not strictly necessary.

1.2 Minimal substrates for AI as a new angle on It from Bit

Minimal substrates for AI are systems with three properties. First, they are universal in the Turing sense: any effective procedure can be implemented on them. Second, they are defined by a very small set of primitives—perhaps only a single gate, a single instruction, or a handful of combinators. Third, we know in some detail how to encode learning, search, and representation on top of them, at least in principle.

NAND-only digital logic is the canonical example. Starting from a single two-input gate, one can construct all Boolean operations, memory elements, and control structures, and from there assemble complete processors capable of running any AI code that would run on a conventional machine. One-instruction set computers such as SUBLEQ show that, even at the instruction level, a single carefully chosen operation can suffice. Lambda calculus and SKI combinatory logic demonstrate that universality can be reached in purely symbolic systems with only a few syntactic forms and reduction rules. Universal cellular automata show that a simple local update rule on a grid of cells can simulate any algorithmic process.

The key observation is that AI does not care which of these minimal substrates is used, so long as it is universal and sufficiently resourced. A learning algorithm designed for a conventional machine can be compiled down to NAND, to a one-instruction core, or to a combinatory system. The internal code changes; the informational pattern of how the agent maps histories of observation to actions and internal states can be preserved. Minimal substrates therefore make visible a deep fungibility: the same high-level intelligence can inhabit very different low-level “bodies,” provided those bodies implement the right bit-level dynamics.

This fungibility is precisely the kind of phenomenon that an It-from-Bit view would lead one to expect. What remains fixed across substrates is the informational structure; what varies is the material realization. Minimal universality and AI provide a living laboratory in which this structure–realization distinction is not just philosophical, but technical and explicit.

1.3 From physical hardware to abstract description: where "bit" enters

Between the raw physics of a device and the intelligent behavior of an AI system lies a stack of descriptions. At the lowest engineered level, one talks about transistors, voltages, and timing constraints. A step up, one talks about logic gates and flip-flops. Above that, about instruction sets, machine code, and microarchitectures. Higher still, about high-level languages, data structures, and algorithms. At the top, about models, training procedures, and agent behaviors.

At each layer, the same physical system is being redescribed in more abstract terms. The transistor configuration is described as a network of NAND gates; that network is described as implementing adders and registers; those are described as implementing instruction semantics; and so on. By the time we reach the AI level, the underlying hardware has effectively disappeared from view. What matters are the bit patterns in memory, the transitions between machine states, and the higher-level invariants they sustain.

The "bit" in It from Bit enters precisely in this cascade of descriptions. It is not a mysterious physical atom, but a role that shows up wherever a system can robustly realize a binary distinction and propagate it through time and space. A transistor-based NAND gate, a spin in a magnetic material, a two-level quantum system, or a cell in a cellular automaton can all play that role. The choice of primitive language—NAND, SUBLEQ, lambda calculus—specifies how bits are arranged and manipulated; the AI design specifies what patterns those bits should realize.

Minimal languages make this multi-layer structure clean. They give a simple, explicit mapping from physical dynamics to abstract state transitions. They show how a small repertoire of local operations can generate the rich global patterns we label as intelligent. In doing so, they provide concrete instances of the It-from-Bit claim: the "it" of observable behavior is a compiled, layered expression of the "bit" patterns defined by the primitive language and implemented by the hardware.

1.4 Structure of the paper and relation to prior work on minimal universality

The rest of the paper develops this connection in stages. Section 2 briefly recaps the theory of minimal universal substrates for computation and AI, drawing on prior work that has already established the sufficiency of NAND, one-instruction machines, lambda calculus, combinatory logic, and universal cellular automata for implementing arbitrary algorithms. This section serves as a computational baseline: it states clearly what is already known and does not need to be re-proved.

Section 3 turns to Wheeler's It-from-Bit slogan and its interpretations in physics, clarifying what is meant by informational primacy and distinguishing strong ontological claims from weaker epistemic ones. It also notes that most of this literature has paid little attention to AI or to minimal computational languages, treating information largely as something extracted from nature rather than also something we deliberately engineer.

Section 4 introduces an explicitly descriptive viewpoint: AI systems, architectures, and even whole physical devices are treated as bit strings on some fixed reference machine. Section 5 then argues that minimal languages and universality provide strong evidence that this descriptive level has a kind of primacy over particular material implementations, at least for questions about intelligence.

Section 6 uses AI as the central case study: it analyzes intelligent behavior as an invariant computational pattern that can be realized across substrates, and suggests that, if anything in our technological repertoire deserves to be called "it from bit," it is the construction of artificial agents. Section 7 reintroduces physics and thermodynamics, noting that informational primacy does not erase the significance of energy, entropy, and noise. Section 8 connects these classical considerations to quantum information and the "it from qubit" refinement.

Section 9 presents objections and alternative readings, emphasizing that minimal universality does not by itself settle questions about consciousness or ontology. Section 10 draws together the strands to make a cautious but strong case that minimal substrates for AI significantly reinforce Wheeler's view. Section 11 closes with an outlook on future theoretical and experimental directions at the intersection of AI, information, and fundamental physics.

Throughout, the argument builds directly on prior work on minimal universality, but repurposes it: not merely as a branch of theoretical computer science, but as a key piece of evidence in a broader informational reading of intelligence and, by extension, of the physical world.

2. Recap: Minimal Substrates for Intelligence

The case for It from Bit that this paper wants to develop leans heavily on existing work about minimal universal substrates for computation. Before drawing philosophical conclusions, it is worth making the computational situation as explicit and uncontroversial as possible. This section therefore does not present new results; it simply assembles, in one place, the ingredients that later sections treat as givens.

At a high level, there are four key points. First, NAND is functionally complete for Boolean logic and sufficient, with feedback, to build any digital computer. Second, several radically different formalisms—one-instruction set computers, lambda calculus, SKI combinatory logic, and universal cellular automata—are all minimal in their own ways yet computationally universal. Third, modern AI methods can be seen as recurring patterns of learning, search, and representation that run on top of any such universal machine. Fourth, across both theory and practice, substrate independence appears not as a speculative claim but as a recurring, empirically grounded theme. These points together form the "minimal substrates for intelligence" backdrop against which the It-from-Bit argument will unfold.

2.1 NAND as a complete basis for digital logic and computation

NAND occupies a privileged position in digital design because it is functionally complete: any Boolean function can be built solely out of NAND gates. This fact is standard in logic and computer engineering. NOT, AND, OR, and XOR can each be synthesized from small networks of NANDs; once those are available, more complex combinational components such as adders, comparators, encoders, and decoders follow by standard constructions. With feedback, NAND gates can also be connected to form latches and flip-flops, the basic elements of digital memory.

From these pieces, the entire hierarchy of digital architectures can be assembled. Registers are arrays of flip-flops; adders and multiplexers, built from NANDs, form arithmetic and control logic; instruction decoders, also built from Boolean logic, interpret bit patterns as operations. A program counter is a register plus an incrementer and conditional update circuitry. An arithmetic logic unit combines adders, shifters, and logical operators under the control of instruction bits. A minimal CPU stitches these subsystems together, all ultimately realized as structured networks of NAND gates.

Because such a CPU is Turing complete, any effective procedure, including any AI algorithm runnable on a conventional machine, can in principle be implemented on this NAND-only substrate. NAND therefore serves as a concrete, hardware-oriented example of an extremely sparse primitive set that is nonetheless sufficient for universal computation. For later purposes, the important point is not the engineering detail but the logical conclusion: nothing more than the repeated application of a single two-input Boolean operation, arranged appropriately, is required to realize all of digital computation and, by extension, all digital AI.

2.2 Other minimal formalisms: SUBLEQ, lambda calculus, SKI, cellular automata

NAND is not unique in its minimality. The theory of computation offers a whole zoo of formalisms that reach universality from very small starting points. Each of these formalisms brings out a different aspect of what it means to be "minimal" while still being capable of implementing arbitrarily complex algorithms.

One-instruction set computers, exemplified by SUBLEQ ("subtract and branch if less than or equal to zero"), show that even at the assembly-language level, a single instruction can be enough. In SUBLEQ, memory is a flat array of integers; each instruction takes three addresses and performs a subtraction plus a conditional branch. Load, store, addition, logical operations, and even indirect addressing can all be encoded as patterns of this lone primitive. Programs become verbose and hard to read, but there is no loss of expressive power: any program written for a conventional instruction set can be compiled down to sequences of SUBLEQ.

Lambda calculus approaches minimality from the side of pure function definition and application. Its syntax includes only variables, function abstractions, and applications. There are no built-in numbers or control structures; numerals, booleans, data structures, and recursion are all encoded as higher-order functions. Nevertheless, lambda calculus is equivalent in expressive power to Turing machines. Every computable function can be represented as a lambda term; every step of a computation corresponds to a small, local reduction (beta reduction) in that term.

SKI combinatory logic goes even further by eliminating variables. All computation is expressed in terms of combinators S, K, and I (and in many presentations, I is defined from S and K). Reduction rules for these combinators allow arbitrary lambda expressions, and hence arbitrary computable functions, to be encoded. Computation becomes a matter of repeatedly applying simple local rewrite rules to trees of combinator applications.

Universal cellular automata provide a spatial and dynamical flavor of minimality. A cellular automaton consists of a regular grid of cells, each with a finite set of possible states. At each discrete time step, the state of every cell is updated according to a fixed local rule that depends on its own state and those of its neighbors. Certain cellular automata, despite having only a handful of states and simple local rules, can simulate Turing machines. Here, the primitive "language" is not a set of instructions or combinators, but a single local update rule applied uniformly across space and time.

What unites these examples is universality: each can emulate any other, up to an encoding overhead. What distinguishes them is which knobs are turned down. SUBLEQ minimizes the instruction set while retaining a conventional memory model. Lambda and SKI minimize the syntactic constructs and push everything into function application and reduction. Cellular automata minimize the variety of local behavior and push everything into emergent patterns of interaction. Together, they demonstrate that minimality is a multi-dimensional concept and that there are many distinct routes to a universal computing substrate.

2.3 AI as patterns of learning, search, and representation on universal machines

Against this background, artificial intelligence can be understood as a family of computational patterns that live on top of universal machines. These patterns recur across architectures and domains, even when the details differ. They include, most prominently, learning, search, and representation.

Learning, in the broadest sense, is the process of updating internal parameters in response to data or feedback to improve future performance. In neural networks, these parameters are weights and biases; in probabilistic models, they are distributions or sufficient statistics; in symbolic learners, they may be rule sets or logical theories. Computation here consists of repeated passes over data, gradient or gradient-like updates, and the maintenance of structures that summarize past experience.

Search covers the exploration of combinatorial or continuous spaces for states or solutions that satisfy some criterion: minimal cost, maximal reward, logical consistency, or approximate optimality. Search algorithms expand and evaluate candidate states, prune unpromising branches, and sometimes incorporate stochasticity and heuristics. Planning, game playing, theorem proving, and many optimization procedures fall into this pattern.

Representation concerns how states, concepts, and hypotheses are encoded so they can be manipulated effectively. Vector encodings, graphs, logical formulas, programs, and hybrid structures each support different kinds of generalization and inference. The choice of representation shapes which operations are cheap and which are expensive, and thus shapes the overall structure of AI algorithms.

All of these patterns can be realized on any universal machine. On a NAND-based CPU, they become sequences of arithmetic and logical operations on memory-resident data structures. On SUBLEQ, they are elaborate compositions of the single subtract-and-branch primitive. In lambda calculus, they are higher-order function compositions and reductions over functionally encoded data structures. In cellular automata, they appear as self-organizing and interacting patterns of local configurations. In every case, the same high-level pattern—learn from data, search in model space, represent structure—is manifest in different low-level dialects.

This observation is central for the It-from-Bit argument later in the paper. It suggests that AI is not tied to any particular implementation style. Instead, there is a class of computational behaviors that can be instantiated in many ways, each of which is equivalent up to encoding, provided the underlying machine is universal and sufficiently resourced. The patterns, not the particulars of the substrate, are what define the intelligence.

2.4 Substrate independence as an empirical and theoretical theme

Substrate independence is often treated as a speculative philosophical thesis: the idea that minds, or at least intelligent behavior, can be implemented in multiple physical media. In the context of minimal computational substrates and AI, it is also an empirical and engineering fact.

On the theoretical side, the Church–Turing framework formalizes the equivalence of many different computational models. Once a system is Turing complete, it can simulate any other such system, with at most polynomial overhead. This establishes an abstract substrate independence: algorithms, including AI algorithms, do not fundamentally depend on whether they run on a Turing machine, a register machine, a NAND CPU, or a lambda calculus interpreter.

On the practical side, everyday computing already exhibits a weaker but still striking substrate independence. The same high-level language program can be compiled and run on different instruction set architectures, using different microarchitectures, implemented in different semiconductor technologies. The rise of virtual machines, containers, and cloud platforms adds further layers of indirection. AI frameworks that present a uniform interface can target CPUs, GPUs, and specialized accelerators interchangeably, with only configuration changes.

Minimal formalisms make this theme more extreme and more transparent. They show that substrate independence is not merely the compatibility of similar chips or instruction sets, but a general feature of universality itself. If a learning algorithm can be implemented in SUBLEQ, a cellular automaton, and SKI combinatory logic, then the relationship between intelligent behavior and low-level machinery is clearly many-to-one. Multiple, very different physical and formal substrates can realize the same high-level agent.

This does not imply that all substrates are equally efficient or equally easy to engineer. It also does not, by itself, settle questions about consciousness or subjective experience. What it does establish is that, for the purposes of describing and analyzing intelligence as a computational pattern, there is a robust layer of description at which different substrates become interchangeable. The rest of the paper takes this interchangeability seriously and treats it as one of the strongest concrete motivations for reading Wheeler's "It from Bit" as more than a metaphor.

3. Wheeler's It from Bit: Interpretations and Tensions

Wheeler's "It from Bit" did not appear in a vacuum. It emerged from decades of work at the intersection of gravitation, quantum theory, and information, and it was meant as both a slogan and a provocation. The phrase condenses the idea that physical entities and events ("it") ultimately derive their meaning and even their existence from discrete units of information ("bit"). Over time, the slogan has acquired multiple interpretations, some modest and epistemic, others bold and ontological. It has also attracted criticism from physicists and philosophers who worry that it either says too little, or far too much.

This section situates It from Bit historically, distinguishes strong and weak readings of informational primacy, examines the role of observers and measurements, and highlights a notable gap: while the physics community has debated Wheeler largely in the context of quantum measurement and cosmology, the computational realities of AI and minimal universality have remained mostly offstage. The later argument of this paper aims to bring those realities onto the same stage.

3.1 Historical context of It from Bit in physics and cosmology

Wheeler's slogan arose from a specific cluster of concerns in mid- to late-twentieth-century physics. On the one hand, quantum theory had already eroded classical intuitions about an observer-independent world of definite properties. Measurement outcomes appeared to be created, not merely revealed, in the act of observation. On the other hand, developments in black hole physics and thermodynamics suggested deep links between gravity, quantum mechanics, and entropy, a quantity with both physical and informational meanings.

Wheeler was particularly struck by the way yes/no questions seemed to sit at the heart of quantum experiments. A detector either clicks or it does not; a spin measurement yields up or down; a photon is registered here or there. He began to speak of "elementary acts of observer-participancy" as the building blocks of reality, suggesting that such binary events might be more fundamental than the continuous mathematical structures used to describe fields and spacetime.

In cosmology, Wheeler's ideas about a "participatory universe" further emphasized the role of information. The universe, in this picture, is not a ready-made stage waiting for observers to appear. Instead, the very tapestry of spacetime and matter is, in some sense, shaped by informational events, including measurements carried out by observers embedded within the system. It from Bit was his way of saying that the universe's "hardware" may be less fundamental than the "software" of informational interactions.

These proposals were never fully formalized into a single, accepted theory. Instead, they served as a conceptual north star for subsequent work in quantum information, black hole thermodynamics, and holography. They also left behind a set of tensions: how literally should one take the slogan, and what exactly is the relationship between physical states and informational bits?

3.2 Strong versus weak readings of informational primacy

At least two broad families of interpretations can be extracted from Wheeler's slogan. A weak, or epistemic, reading takes It from Bit to mean that information is the most appropriate language for describing physical systems, not that it literally constitutes them. On this view, bits summarize experimental outcomes and constraints; they capture what can be known and communicated. Information is primary for us as theorists, because it provides a unifying framework, but the underlying reality may still be fundamentally "stuff-like."

A strong, or ontological, reading is bolder. It suggests that physical states are nothing over and above informational structures, that the correct base-level description of the world is not in terms of fields or particles, but in terms of abstract information-bearing degrees of freedom. Matter, in this view, is a particular organization of information, rather than information being a property of matter.

These readings are not purely academic distinctions. The weak reading is relatively uncontroversial and meshes well with existing practice in statistical mechanics, coding theory, and quantum information. The strong reading raises deeper questions. If "it" really comes from "bit," what fixes the underlying information-theoretic substrate? Are bits tied to some specific physical system, such as spin-1/2 particles and qubits, or can they be completely abstract? What, if anything, distinguishes physical information from the information in a computer simulation or an abstract mathematical structure?

This paper does not attempt to resolve these issues. Instead, it uses minimal substrates for AI to put pressure on the boundary between the weak and strong readings. The fact that the same intelligent behavior can be instantiated in so many different computational substrates suggests that, at least for intelligence, an ontologically thin notion of "bit patterns and their dynamics" captures something stable and central that material specifics do not.

3.3 Observers, measurements, and informational events

Wheeler's emphasis on bits was closely tied to his views on observation and measurement. For him, the fundamental events in quantum theory were yes/no answers to questions posed to the world: did the detector fire, did the particle pass through this slit, did the geiger counter register a decay? Each such event could be represented as one bit of information, and the accumulation of these bits over time constituted, in his view, the unfolding of physical reality.

This observer-centric framing raises familiar questions. What counts as an observer? Are measurements special physical processes, or are they just interactions that happen to leave stable records? Is the bit created at the moment of measurement, or does it merely become accessible to agents like us? How does this picture reconcile with relativistic constraints and the apparent continuity of fields and spacetime?

Many responses in quantum foundations have tried to de-emphasize the role of the observer, either by treating measurements as ordinary physical interactions analyzed within the theory, or by adopting interpretations where all outcomes occur and "bits" are just branch-relative records. Nonetheless, the idea that discrete informational events play a crucial role in the effective description of quantum phenomena has persisted. Even in interpretations that downplay observers, key quantities of interest—such as entanglement entropy or channel capacities—are explicitly informational.

What has been less explored is the connection between these measurement-centric bits and the bits manipulated in artificial computing systems. Wheeler's original context was photomultipliers and cosmic-scale thought experiments, not GPUs and neural networks. Yet at a formal level, the distinction is blurry: a detector click is a macroscopic, stable change in a physical device; a digital bit flip in a memory cell is of the same general type. Both are physical events that instantiate an informational distinction.

Once this parallel is taken seriously, It from Bit starts to encompass not only the bits extracted from nature by experiments, but also the bits we arrange in machines to realize artificial agents. The line between "observational" and "computational" bits begins to look more like a matter of perspective than of principle.

3.4 Why AI and minimal computation have been mostly absent from the debate

Despite these parallels, discussions of It from Bit in physics have largely proceeded without much reference to artificial intelligence or to minimal computational formalisms. There are several reasons for this separation.

Historically, Wheeler's work predated the modern explosion of AI and the development of rich, concrete models of learning and representation. When he coined his slogan, computation was certainly well understood in theoretical terms, but large-scale artificial agents and deep learning architectures did not yet exist. It was natural to focus on measurement outcomes in physical experiments rather than on bits in engineered systems.

Disciplinary boundaries have also played a role. Quantum foundations and cosmology have their own communities, questions, and technical frameworks. Theoretical computer science and AI have another set. While ideas have flowed between them—witness the rise of quantum computing and quantum information theory—much of the AI community's attention has been directed toward performance, scalability, and applications rather than foundational metaphysics. Conversely, many physicists interested in It from Bit have treated computation as an analogy rather than as an object of direct study.

Minimal computational languages, such as NAND-only logic, one-instruction set computers, and combinatory systems, sit at an awkward intersection. They belong to the theory of computation and to low-level hardware design, but they have not generally been enlisted in debates about informational ontology. Their role has been to demonstrate universality and to inform processor design, not to serve as evidence about the fundamental nature of reality.

The central suggestion of this paper is that this separation has left an opportunity on the table. Minimal substrates for AI provide concrete, technically grounded examples of how informational structures can be implemented in many different physical and formal guises. They show, in an especially vivid way, how "it" can be moved from one "bit-carrier" to another without loss of function. Bringing these examples into the It-from-Bit debate enriches it, not by offering new physics, but by providing a sharper, engineering-informed lens on what informational primacy could plausibly mean.

4. Bit Before It: Description, Program, and Device

The slogan It from Bit suggests that informational structure underlies physical structure. In the context of computation and AI, that claim can be sharpened by looking closely at how we describe the systems we build. A physical device does not arrive with its own meaning; we impose a layered description on it, specifying how its physical degrees of freedom should be interpreted as logical states, instructions, and algorithms. In practice, everything from a bare processor to a large AI model is packaged and manipulated as a bit string. Architectures are bit strings, programs are bit strings, and trained models are bit strings. The physical hardware is one particular way of realizing those descriptions.

This section develops that viewpoint. It begins with the simple but powerful idea of describing machines themselves as bit strings. It then invokes Kolmogorov complexity to formalize the notion of shortest description, treating physical processes as objects that can, in principle, be generated by programs. From there, it reverses the usual explanatory order: instead of saying that descriptions are abstractions of devices, it treats devices as realizations of descriptions. Finally, it notes the central role of compilers and interpreters as mechanisms that turn raw bit strings into running systems, making visible the route from bit to it.

4.1 Describing machines as bit strings: architectures, programs, and models

In everyday programming, "the program" is a text file in some high-level language. But from the standpoint of a universal machine, everything that matters about that program is captured by a finite bit string: the compiled machine code, the libraries it links against, and any configuration data or model parameters it uses. More generally, any digital artifact we care about can be represented as a bit string, so long as we fix an encoding scheme.

This applies not only to application code but also to the machines that run it. A processor's instruction set architecture can be specified as a description: a table of opcodes, operand formats, and effects on an abstract machine state. A particular microarchitecture can be described as a netlist of gates and flip-flops, together with timing constraints and control logic. A higher-level architecture, such as a neural network, can be described as a graph of layers, activation functions, and connectivity patterns. Each of these descriptions can be serialized as bits.

When we talk about an AI system, we often distinguish three levels: the architecture, the training procedure, and the resulting trained model. The architecture is a template: a parameterized description of a computation graph or a rule system. The training procedure is an algorithm that, given data, returns parameter values that fit some objective. The trained model is a particular setting of those parameters. All three can be written down as bit strings. They differ not in kind but in their role: template, process, and outcome.

Behind this is a choice of reference machine. To say that a bit string "describes" a system is to say that if we feed that string into some agreed-upon interpreter or compiler, we obtain a functioning instance of the system. A hardware description language, for example, comes with a toolchain that turns textual descriptions into masks for a chip. A high-level programming language comes with a compiler that turns source into machine code. Model description formats come with runtimes that instantiate computation graphs from serialized metadata and weights.

From this descriptive standpoint, the hardware device is one particular realization of a prior informational object: the design. The AI model in memory is one realization of a prior informational object: the set of weights and topology. The fact that all of these can be moved, copied, and transformed as bit strings suggests a natural sense in which "bit" comes before "it," at least in the engineering chain.

4.2 Kolmogorov complexity and the shortest descriptions of physical processes

Once machines, programs, and models are treated as bit strings, it is natural to ask how concise those descriptions can be made. Kolmogorov complexity offers a way to formalize that question. Given a fixed universal reference machine, the Kolmogorov complexity of an object is the length of the shortest program that causes the machine to output that object and halt. Intuitively, it is the ultimate measure of how compressible an object is, given unlimited cleverness in choosing its encoding.

Applied to AI, one can imagine the Kolmogorov complexity of an agent's policy or behavior. How long is the shortest program that, when run on some universal machine, reproduces the same mapping from input histories to actions? Similarly, one can ask about the complexity of a physical process: the shortest program that, when interpreted as a simulation or generative procedure, yields sequences of observations statistically indistinguishable from what the process produces.

Two aspects of Kolmogorov complexity are especially relevant here. First, it is defined relative to a universal machine, but the choice of machine only affects the complexity value by an additive constant. Changing the primitive language or substrate does not radically alter which objects are simple and which are complex. Second, Kolmogorov complexity is uncomputable; there is no general method for finding the shortest description of an arbitrary object. It is therefore a conceptual yardstick rather than a practical tool, though approximations and bounds can sometimes be found.

Thinking in these terms shifts attention from the physical size or intricacy of a system to the informational economy of its description. A large neural network with millions of parameters may have a relatively short generative description if it can be produced by a compact training algorithm on a compact dataset. Conversely, a hand-tuned system that hard-codes many individual decisions may have a higher complexity if there is no simple pattern from which those decisions can be derived.

For the purposes of It from Bit, Kolmogorov complexity emphasizes that physical systems and intelligent agents can be viewed as outputs of programs. The physical details of their embodiment matter less, in this perspective, than the length and structure of the descriptions that give rise to them. Minimal computational languages then become candidate reference machines for defining those complexities, anchoring the notion of “shortest description” in an explicit substrate.

4.3 Devices as realizations of descriptions rather than the other way around

In everyday thinking, it is natural to suppose that descriptions are abstractions of devices. We build a machine; then we describe how it works. The machine is primary, the description derivative. From an informational standpoint, especially in engineered systems, this order can be reversed. We start with a description and then construct a device to realize it.

A chip design, expressed in a hardware description language, is not an after-the-fact summary of an existing device; it is the template from which the device is fabricated. A program, expressed in source code, is not merely a commentary on a machine; it is the blueprint from which executable behavior is derived. A model architecture and its learned parameters are not just records of what a system is doing; they are the specifications that determine what it will do when instantiated.

Treating descriptions as primary has two consequences. First, it emphasizes that there may be many different physical devices that realize the same description. A logic gate can be built from

CMOS transistors, from relay contacts, from optical switches, or from molecular structures. A processor implementing a given instruction set can be built in many microarchitectural styles. A neural network model can run on CPUs, GPUs, or specialized accelerators. In each case, what unifies these realizations is not their material composition but their conformity to an abstract description.

Second, it blurs the line between description and implementation. A device is a physical system whose dynamics follow some pattern; to the extent that pattern can be captured by a finite description, the device can be treated as a realization of that description. Conversely, any sufficiently detailed description can be seen as specifying a device, at least in principle. The distinction becomes one of medium: is the description encoded in text, in layout files, in doped silicon, or in electromagnetic states?

From this angle, It from Bit reads as the claim that the most fundamental characterization of physical "it" is informational: a description of state space, transition rules, and boundary conditions. Devices are particular embodiments of such descriptions, not the other way around. Minimal computational languages make this especially clear, because they strip devices down to the repeated application of a few primitive operations. The richer the device, the more it looks like a highly elaborated realization of a relatively compact generative description.

4.4 The role of compilers and interpreters in turning bit into it

If descriptions come first and devices realize them, one still needs mechanisms that connect the two. In computing, those mechanisms are compilers, interpreters, assemblers, and synthesis tools. They are themselves programs—bit strings running on universal machines—that transform one description into another, or into a configuration of physical substrate.

A compiler takes a high-level program and produces machine code. An interpreter reads a program and directly controls a machine according to its instructions. A hardware synthesis tool takes a logic-level description and produces a gate-level netlist. A model loader reads a serialized neural network and allocates memory, binds operations, and sets weights so that the model can be executed. In all of these cases, a bit string is turned into an active process on a device.

Compilers and interpreters are therefore key agents in the practical realization of It from Bit. They operationalize the mapping from abstract description to concrete behavior. Given a universal machine and a compiler, the same high-level description can result in identical observable behavior across different hardware platforms. This is substrate independence in action: bit patterns and transformation rules are enough to carry an "it" from one physical basis to another.

There is also a recursive aspect. The compilers and interpreters themselves are described by bit strings and run on lower-level machines. A compiler can be compiled by another compiler; an interpreter can be interpreted. At the bottom, some minimal language and substrate must exist that need not be further compiled. NAND gates, one-instruction cores, or combinatory systems can serve as such a base. From there, progressively richer description languages can be implemented, layer by layer.

This tower of compilers and interpreters gives a concrete structure to the idea of bit before it. The ultimate behavior of an AI system is the result of a chain of translations from a compact description into successively lower-level codes, ending in sequences of primitive operations on some physical substrate. Each step is itself governed by a description. At no point is it necessary to invoke a special “intelligence substance”; what is carried along the chain is informational structure, encoded in bits.

For Wheeler’s slogan, the lesson is not that hardware is irrelevant, but that hardware can be systematically subordinated to the logic of description and transformation. Once a minimal universal substrate is available, everything else that makes a system intelligent comes from the bit strings we feed into it and the compilers that organize their execution. The It-from-Bit claim thus finds a precise and operational expression: in the realm of artificial intelligence, at least, “it” is what happens when carefully structured bits are pushed through a universal substrate according to the rules of a compiler chain.

5. Minimal Languages as Evidence for Informational Primacy

Minimal universal languages do more than tidy up the theory of computation. They dramatize a particular fact about intelligence and complex behavior: that the same high-level capacities can be realized in wildly different low-level dialects, provided each dialect is universal and we are willing to pay the translation cost. In doing so, they tilt the explanatory balance away from material particulars and toward informational structure.

This section develops that claim along four lines. First, it highlights the fungibility of substrate as one moves from NAND gates through SUBLEQ and lambda calculus to combinators. Second, it argues that universality effectively collapses many hardware specifics into the domain of pure description. Third, it examines how AI architectures remain intelligibly “the same” even when ported across radically different underlying “stuff.” Finally, it interprets minimal cores as witnesses that, in an important sense, nothing more is needed than bit-patterns and their dynamics to host intelligence.

5.1 The fungibility of substrate: from NAND gates to combinators

At the lowest engineered level, a NAND gate is a physical object. It is a particular arrangement of transistors, wires, and materials, with characteristic delays and power dissipation. In a different technology, the same logical function could be realized with relays, optical devices, or molecular switches. The substitutability of these realizations already hints at fungibility: the logical behavior matters more than the particular atoms.

Stepping up one level, a network of NAND gates configured as a CPU is itself a substrate for more abstract operations. Machine instructions, registers, and memory cells are logical concepts that can be realized on any hardware that supports the right state transitions. A program written in machine code does not know whether its instructions are ultimately carried out by silicon transistors or some other medium. As long as the effective behavior of the instruction set remains the same, the program's semantics are preserved.

Minimal formalisms like SUBLEQ, lambda calculus, and SKI combinators push this fungibility further. SUBLEQ treats the machine as an array of integers and a single instruction that subtracts and conditionally branches. Lambda calculus treats everything as function definition and application, with all data encoded as functional objects. SKI combinatory logic eliminates even variables, reducing computation to the repeated application of a few combinators according to fixed rewrite rules.

What is remarkable is not just that these formalisms are equivalent in expressive power, but that one can systematically translate programs between them. Given a program for a NAND-based CPU, one can compile it to SUBLEQ code, transform that into a lambda term, and then further encode it into SKI combinators. The resulting structures look nothing like the original at the surface, but their input–output behavior can be made equivalent up to timing and resource overhead.

As one moves along this chain, the notion of substrate becomes increasingly abstract. The underlying physical device may be the same, but the effective substrate for the program shifts from gates to instructions to higher-order functions to combinators. Each level treats the previous one as a medium for its own patterns. The fact that so many such levels can be interposed, without losing the ability to realize the same computational behavior, is a concrete manifestation of substrate fungibility.

5.2 Universality as the collapse of hardware specifics into pure description

Universality is, in one sense, a negative property: it says that no new primitive is needed beyond those already present to express any effective procedure. In another sense, it is a powerful positive constraint: it ensures that many differences between machines are superficial from the standpoint of what can ultimately be computed.

When a system is universal, its hardware specifics can be summarized, for the purposes of programming, in a finite description: the definition of its instruction set, or the reduction rules governing its terms. Once this description is fixed, the machine becomes a target for compilation. The programmer no longer needs to think about transistor dimensions or wire layouts; those are encapsulated in the fact that the instructions behave as specified. Hardware details have collapsed into the interface.

Minimal languages accentuate this collapse. A NAND-based machine with a handful of instruction types can be fully characterized by a short manual, defining what each opcode does to the abstract machine state. A SUBLEQ machine can be characterized by an even shorter description, since it has only one primitive instruction. A lambda calculus interpreter needs only to define term syntax and a small set of reduction rules. SKI needs only the rules for how S, K, and I act on their arguments.

From the perspective of universality, all these machines are equivalent modulo encoding. The same algorithm can be represented by different bit strings on each, but its abstract structure—its control flow, data dependencies, and asymptotic behavior—remains the same. The differences between them are pushed down into constant factors and practical concerns such as performance and ease of use.

In this way, universality converts diverse hardware into a smaller number of informationally defined platforms. An AI algorithm written in a high-level language is not targeted at "silicon" in any direct sense; it is targeted at an abstract machine specified by a finite description. As long as the physical hardware realizes that description faithfully, the algorithm's behavior is preserved. The more minimal the description, the clearer it becomes that what matters is the pattern of state transitions, not the details of their embodiment.

5.3 How AI architectures survive radical changes in underlying "stuff"

Artificial intelligence architectures provide especially vivid examples of how high-level structure can survive radical changes in underlying "stuff." A neural network architecture, for instance, is defined by layer types, connectivity patterns, activation functions, and learning rules. When practitioners speak about a particular model—say, a convolutional network or a transformer—they are referring to this abstract structure, not to any specific machine on which it runs.

That structure can be implemented in many different environments. A convolutional network can run on a single CPU, on parallel GPUs, on specialized tensor processing units, or in a highly distributed fashion across many machines. The representations may be stored in different formats; the operations may be fused or reordered for performance; low-precision arithmetic may be used instead of high precision. Yet, within reasonable tolerances, the model is regarded as “the same” if it implements the same mapping from inputs to outputs and can be trained with the same procedures.

Even more dramatically, the same architecture can, in principle, be compiled down to a minimal substrate like SUBLEQ or SKI. The result would be impractical, but conceptually straightforward: arithmetic operations and matrix multiplications would be replaced by long sequences of primitive steps; the control flow of training loops would be encoded in the minimal instruction set or combinator rewrites. The structure of the architecture would be preserved, albeit in a more verbose, lower-level form.

Symbolic AI systems and search-based planners exhibit similar robustness. A planner that operates on state descriptions and action schemas can be implemented in a high-level language, compiled to different machine architectures, or encoded in a minimal core whose primitives bear little resemblance to the domain-level concepts. As long as the logical relationships and search strategies are preserved, the system is treated as the same planner.

This persistence of identity across substrates underwrites a strong sense in which AI architectures are informational entities. Their “essence” lies in how they transform information, not in the particular material or formal substrate used to enact those transformations. Minimal languages serve as extreme test cases: if an architecture can be realized even on an austere substrate, then its dependence on any given kind of “stuff” is revealed to be contingent rather than necessary.

5.4 Minimal cores as witnesses that “nothing more” is needed than bit-patterns

Minimal cores are not just convenient curiosities in the theory of computation. They are witnesses to a striking fact: once a system has a certain minimal informational structure, nothing more is needed at the level of primitives to host any computable process, including those we associate with intelligence.

A NAND-only gate library is enough to build a universal digital machine. A single subtract-and-branch instruction is enough to encode arbitrary algorithms over random-access memory. Three combinators with simple rewrite rules are enough to express any lambda term and thus any computable function. A cellular automaton with a small local rule set is enough to simulate Turing machines and thereby to host any algorithmic pattern.

From the standpoint of AI, this means that the primitive substrate does not need to be “intelligence flavored.” There is no necessity for special-purpose reasoning instructions, concept representations, or learning modules at the lowest level. Everything that makes an AI system intelligent can be layered on top of a core that knows nothing about the domain, nothing about goals, and nothing about semantics beyond the manipulation of bit-patterns or abstract terms according to fixed rules.

This does not mean that minimal cores are the best or only way to build AI, nor that higher-level structures are unimportant. On the contrary, much of the art of AI lies in designing architectures and representations suited to particular tasks. But minimal cores do show that, in the deepest sense, intelligence does not depend on anything more exotic than the existence of a universal pattern-manipulating substrate. Once that substrate is in place, sufficiently rich bit strings, interpreted in the right way, suffice.

Interpreted through Wheeler's slogan, minimal cores thus offer a particularly sharp instantiation of It from Bit. They demonstrate that at least one important class of "it"—artificial intelligence—is generated entirely from bit-level structure and dynamics over a universal substrate. The hardware can vary; the instruction set can be bizarrely minimal; the emergence of intelligent behavior remains possible as long as the informational conditions for universality and sufficient resources are met. In that sense, minimal languages do not just illustrate informational primacy; they make it hard to deny.

6. AI as a Canonical It-from-Bit Construction

If Wheeler's slogan is going to be taken seriously anywhere, artificial intelligence is a natural testing ground. AI systems are not given by nature; they are engineered entities whose entire operational structure can be written down as code and parameters. They stand at the intersection of abstract algorithms and physical devices, making the bit–it relationship unusually visible. In an AI system, every behavior we care about emerges from the interaction of three things: the program that defines an agent's update and decision rules, the data it encounters, and the substrate that executes those rules on those data.

This section treats an intelligent agent in deliberately austere, informational terms. It defines such an agent by its input–output and learning behavior, independent of its internal metaphors or substrate. It then considers how very different implementation styles—neural, symbolic, and hybrid—can nonetheless belong to the same behavioral equivalence class. It interprets training as the construction of informational structure from data, and finally argues that the most interesting properties we call intelligence are emergent invariants that survive changes in underlying "stuff."

6.1 Defining an intelligent agent by its input–output and learning behavior

From a minimal, computational perspective, an intelligent agent can be defined by how it maps histories of interaction to actions, and how that mapping changes in response to experience. At each time step, the agent receives some observation and perhaps a reward signal; it updates an internal state; it chooses an action; and the cycle repeats. The agent's identity, in this view, is exhausted by three ingredients: its state space, its update rule, and its policy for action selection, all seen as functions over bit strings.

Formally, one can imagine the agent's internal state as a finite (but potentially large) binary string. Observations and rewards are likewise encoded as binary strings from some fixed alphabet. An update rule takes the current state plus the latest input and produces a new state. A policy maps the updated state and perhaps the input to an action, again encoded as bits. Learning is present when the update rule and policy jointly allow long-run behavior to adapt to patterns in the data, improving performance on some task or objective.

Crucially, this definition is indifferent to how the state is represented internally. It can be a dense vector of floating-point numbers, a sparse symbolic structure, a graph, or something more exotic. As long as there is some fixed encoding into bits and some fixed decoding of actions back into the environment, the agent is a bit-level dynamical system. Its intelligence is judged by the regularities in its input–output traces over time: prediction accuracy, reward accumulation, stability, and generalization across circumstances.

This input–output-and-learning characterization makes no reference to neurons, symbols, or continuous mathematics. It situates intelligence squarely in the domain of bit patterns and their evolution under fixed rules. For the purposes of It from Bit, this is exactly what is needed: a description of an intelligent process in terms that are fully informational, with the physical implementation treated as a separate question.

6.2 Equivalence classes of implementations: neural, symbolic, and hybrid

Once an agent is defined by its input–output and learning behavior, many different implementation styles can be seen as members of the same equivalence class. A neural implementation realizes the state as activation patterns and synaptic weights, updated by gradient-based rules. A symbolic implementation realizes the state as a database of facts and rules, updated by logical inference and rule induction. A hybrid implementation might combine differentiable components with symbolic ones, passing information back and forth. If all three produce the same observable behavior over the same interaction histories, they are equivalent at the level that matters for intelligence.

This equivalence is not merely conceptual; it is supported by the universality of computational substrates. A symbolic planner can be simulated by a neural network that encodes its state and transition rules in weights and activations, given enough capacity and carefully designed training regimes. Conversely, a neural network's behavior on a discrete set of inputs can be replicated by a symbolic system that stores a sufficiently detailed lookup table or programmatically reconstructs its function. Hybrid systems can be unfolded into either style at the cost of complexity.

From the informational standpoint, what matters is the mapping from histories to actions and the way this mapping changes with ongoing experience. Two agents that share this mapping are indistinguishable to an external observer, even if one is implemented as a deep network on GPUs and the other as a rule-based system on a minimal SUBLEQ machine. They inhabit the same equivalence class of implementations of a single abstract agent.

This notion of equivalence classes makes “it from bit” more precise. The “it” in question is the abstract agent: the function plus the learning dynamics. Each concrete realization—neural, symbolic, hybrid, or otherwise—is a way of knitting bits together so that, under execution, they instantiate that function. Minimal languages and substrates show that the equivalence class is larger than one might have guessed: very sparse, low-level machines can also implement the same agent, confirming that the essential content is informational rather than architectural or material.

6.3 Training as the construction of informational structure from data

If an agent is specified by its update and decision rules, training is the process that configures those rules or their parameters so that the resulting behavior is intelligent rather than arbitrary. In informational terms, training is the construction of structure in the agent’s internal state in response to structure in the data. It is a controlled flow of information from environment to agent, mediated by an algorithm that decides what to keep, what to discard, and how to organize what is kept.

Consider a supervised learning scenario. A model architecture defines a space of possible input–output mappings parameterized by weights. A training algorithm processes examples and adjusts weights to reduce a loss function. Each update moves the model to a new point in parameter space. Over time, the sequence of updates builds an internal representation that captures regularities in the data: clusters, manifolds, correlations, or higher-level features. The final weight configuration can be seen as a compressed encoding of the training data, tailored to the tasks the model must solve.

From this perspective, training is a specialized kind of compression and code construction. The raw data may be huge, but regularities allow a much smaller parameter vector to suffice for accurate predictions. Differences between architectures can then be understood as differences in representational language: some architectures make certain patterns easy to express and compress, others make them hard. A good architecture is one that allows a simple, short description of the relevant regularities, given the training procedure.

Symbolic and hybrid systems exhibit the same pattern. A rule learner compresses observed regularities into logical rules. A program synthesizer compresses input–output traces into compact programs. A hybrid system may compress raw sensory data into latent vectors and then compress those vectors into symbolic concepts. In all cases, training is the transformation of a large, high-entropy stream of bits into a smaller, structured internal representation.

Seen through It from Bit, training becomes a paradigmatic example of information giving rise to "it." Before training, the model's parameters are random or uninformative; the agent is effectively a blank computational substrate. After training, those same bits form a highly structured object that, when executed, produces intelligent behavior. The substrate has not changed; only the informational organization of bits on it has. Intelligence is thus not in the silicon but in the specific patterns of bits sculpted by data and learning dynamics.

6.4 Intelligence as an emergent invariant across substrates

The preceding points set the stage for a stronger claim: what we label as intelligence is an emergent invariant across substrates. It is not a property of neurons, transistors, or combinators per se, but a property of particular patterns of information processing that can be realized on any universal machine. Once those patterns appear—through design, learning, evolution, or some combination—the resulting system exhibits behaviors we recognize as intelligent, regardless of the medium.

To say that intelligence is emergent is to acknowledge that it depends on large-scale organization. No single gate, neuron, or instruction is intelligent; intelligence requires coordinated activity across many such elements, governed by architectures and learning rules that shape how information flows and accumulates. To say that it is an invariant is to stress that, when this organization is reproduced on a different substrate, the functional properties are preserved: prediction improves with experience, plans are formed and revised, goals are pursued in changing environments.

Minimal languages sharpen this picture. They show that the underlying substrate can be stripped down to a bare universal core without depriving the system of the ability to host intelligence. An agent implemented in a high-level language on a modern processor can, in principle, be re-implemented as a program in a one-instruction machine, as a lambda term, or as a configuration of a universal cellular automaton. The resource requirements may become unrealistic, but the conceptual statement stands: the same emergent invariant can float atop very different seas of low-level operations.

This does not trivialize embodiment or physical constraints. Real agents are embedded in worlds with particular physics; their sensors and actuators are constrained; their computation costs energy and takes time. Different substrates will favor different forms of intelligence, and some forms may be practically unreachable on certain media. What minimal universality and AI demonstrate, however, is that these constraints do not define intelligence at the deepest level. They shape which emergent invariants are feasible, not what counts as an invariant.

In this sense, AI is a canonical It-from-Bit construction. It provides clear, manipulable examples in which elaborate “its”—agents that predict, plan, and learn—arise from the organization of bits on universal substrates. The bits are not passive labels on preexisting matter; they are active ingredients in the construction of behavior. When those bits are rearranged according to the right training and architecture design principles, the invariant we call intelligence appears and persists, even as the underlying “stuff” changes.

7. Physics, Thermodynamics, and the Cost of Realizing Bit

The argument so far has leaned heavily on logical universality and informational structure. It has shown that very sparse languages and substrates suffice to host intelligence in principle. That is one side of Wheeler’s slogan. The other side lives in physics. Bits are not free to manipulate; they are embodied in systems that obey thermodynamic laws. Gates dissipate heat, memory cells leak, and noise constantly threatens to corrupt stored and transmitted information. Any It-from-Bit picture that ignores these costs risks becoming detached from the world it is meant to describe.

This section puts physical law back into the story, not to undermine informational primacy, but to clarify its limits and proper domain. It begins with Landauer’s and Bennett’s work on the thermodynamics of computation, which links logical operations to entropy changes and energy dissipation. It then considers the broader physical price of information processing in terms of energy, entropy, and noise. Next, it argues that informational primacy does not erase these constraints; instead, it must be understood as layered on top of them. Finally, it contrasts minimal computational substrates with thermodynamically more favorable ones, highlighting that logical minimality and physical optimality need not coincide.

7.1 Landauer, Bennett, and the thermodynamics of computation

Landauer’s principle states that erasing one bit of information in a physical system has an unavoidable thermodynamic cost: at least a certain minimum amount of heat must be dissipated into the environment, proportional to temperature and the logarithm of two. The principle does not say that every bit flip costs this much, only that logically irreversible operations that compress many possible inputs into fewer outputs must, in aggregate, increase entropy somewhere.

This observation ties logical properties directly to physical ones. A purely formal operation like resetting a memory cell to zero has a different thermodynamic status from, say, a reversible permutation of bits. When a logical operation discards information, mapping many distinct logical states onto a single state, the missing information reappears as entropy in the environment. Conversely, logically reversible operations, which map each input state to a unique output state, can in principle be implemented in ways that avoid this minimal dissipative cost, though practical implementations still face other sources of loss.

Bennett extended these ideas by showing that any computation carried out by a conventional, irreversible machine can be embedded into a larger reversible computation. The reversible machine keeps enough history to make every step logically invertible. At the end, the desired output can be copied off and the rest of the machine can be run backward to uncompute intermediate states, returning ancillary bits to their original values. In this way, the core computational work can be arranged so that only the final erasure of unneeded outputs incurs Landauer's minimal cost.

Taken together, Landauer's and Bennett's work demonstrates that information is not just a bookkeeping device; it is woven into the fabric of thermodynamics. Logical irreversibility and entropy increase are linked; recovering full reversibility requires extra structure and careful design. For the purposes of this paper, the key point is that bits live in a physical world where their manipulation has energetic consequences. Even if intelligence can be described entirely in informational terms, its realization must respect these thermodynamic constraints.

7.2 Energy, entropy, and the physical price of information processing

Beyond minimal erasure costs, real information processing systems incur a host of energetic and entropic expenses. Each gate operation in a digital circuit dissipates energy because of resistive losses, charging and discharging of capacitive nodes, and nonideal device behavior. Clock distribution networks consume power just to coordinate the timing of operations. Memory elements draw energy to retain state, refresh dynamic cells, and drive read and write operations. Communication between components adds further costs.

Noise complicates matters. Thermal fluctuations, electromagnetic interference, and imperfections in materials all threaten to corrupt bits. To maintain reliable computation, systems employ error-correcting codes, redundancy, and signal amplification. These techniques, in turn, impose additional overhead, both in the number of physical degrees of freedom devoted to representing a single logical bit and in the energy required to manage them. The more aggressive the noise environment or the tighter the performance constraints, the higher the resource cost.

At larger scales, heat dissipation becomes a limiting factor. High-performance processors and AI accelerators consume significant power and produce substantial thermal loads. Data centers devote considerable infrastructure to cooling and power distribution. The spatial arrangement of devices, the thermal properties of materials, and the ambient environment all influence what kinds of computations are practical. These are not abstract considerations; they directly impact how large and how fast AI systems can be made.

Entropy enters not only through device-level losses but also through the statistics of data and workload. Compressing data to remove redundancy saves storage and transmission energy. Structuring computation to minimize unnecessary operations, avoid idle cycles, and exploit locality reduces entropy production and waste. In this sense, the informational efficiency of algorithms and architectures feeds back into physical efficiency.

All of this underscores that bit manipulation is a thermodynamic activity. The elegance of minimal languages and universal formalisms does not exempt them from energy and entropy bookkeeping. A NAND gate, an instruction in a one-instruction machine, or a combinator reduction is not just a logical event; it is a physical event that must be paid for in power and heat.

7.3 Why informational primacy does not erase the reality of physical constraints

Given these costs, one might suspect a tension between informational primacy and physical realism. If It from Bit is taken to mean that information is somehow more real or more fundamental than matter, does that not downplay the importance of energy, entropy, and material constraints? Minimal languages might seem to encourage such a view by emphasizing that any substrate capable of supporting universal computation will do.

A more careful reading avoids this conflict. The informational claims made in this paper are primarily about explanatory structure, not about what literally exists. Saying that an AI system is best characterized by its informational patterns is not the same as saying that it does not need hardware. Rather, it says that, for many questions of interest—such as whether two implementations realize the same agent—the hardware can be abstracted away in favor of descriptions that live at the bit level.

Physical constraints then appear as conditions on which informational patterns are realizable, and at what cost. A particular AI architecture may be perfectly definable in a minimal language but practically unrealizable on a given substrate because of energy, space, or time limitations. Another architecture may be both informationally compact and thermodynamically efficient, making it a good choice for implementation. Informational primacy provides the vocabulary for describing architectures and agents; thermodynamics tells us which ones are viable and at what scale.

In this layered view, information and physics are not competing foundations but complementary perspectives. At the bottom, there are physical systems obeying dynamical and thermodynamic laws. On top of that, there are informational descriptions that capture the organization of states and transitions in ways that are useful for design and analysis. Minimal languages and universality clarify the structure of those descriptions. Landauer's and Bennett's results clarify the costs and limits imposed by physics.

For Wheeler's slogan, this suggests a refinement. It from Bit does not mean that bits float free of physical law. It means that, once a suitable physical substrate is in place, the patterns of information it can support become the primary determinants of the complex "its" we care about, such as intelligent agents. The reality of power budgets, heat, and noise does not weaken this claim; it informs it.

7.4 Minimal substrates versus thermodynamically optimal substrates

Minimal computational substrates, such as NAND-only gate libraries, one-instruction machines, or pure combinatory systems, are defined by logical criteria: small primitive sets, simple rules, and universality. Thermodynamically optimal substrates would be defined by different criteria: minimal energy per useful logical operation, minimal entropy production for a given computational task, robustness to noise, and favorable scaling properties.

There is no reason to expect these two notions of minimality to coincide. In fact, they often pull in opposite directions. NAND, for example, is logically appealing because it is functionally complete and uniform, but it is not logically reversible. Every NAND operation discards some information, making it impossible, in principle, to avoid Landauer's minimal erasure costs if NAND is used as the primitive. Reversible gates, such as Toffoli and Fredkin gates, are more natural from a thermodynamic perspective because they can be arranged into computations that are logically invertible, at least in principle. However, they are logically more complex than NAND and awkward to use as the sole primitives in conventional designs.

Similarly, one-instruction set computers are minimal in the syntactic sense, but their implementation may be physically inefficient. Emulating arithmetic, memory access, and control flow through a single primitive may require many more physical operations than using a richer instruction set. The extra operations mean more energy consumption and heat, even though the primitive set is smaller. A more complex instruction set that directly supports common operations can, paradoxically, be more thermodynamically efficient for real workloads.

Lambda calculus and combinatory logic, while beautifully minimal in their formalism, are not particularly well aligned with the physical realities of hardware. Their natural implementations involve frequent creation and manipulation of symbolic structures, which can be expensive in time and energy. In contrast, architectures designed around numerical linear algebra and local memory access patterns can exploit hardware more efficiently, even if their abstract description involves a larger primitive set.

The emerging field of reversible and adiabatic computing explores substrates and design techniques that move toward thermodynamic optimality. These approaches often sacrifice logical minimality and simplicity for reduced dissipation, adding extra degrees of freedom and control to prevent information from being irreversibly lost. At the other extreme, some AI accelerators deliberately introduce redundancy and approximate arithmetic to gain performance and robustness, trading precise logical behavior for favorable thermodynamic and statistical properties.

For the It-from-Bit argument, the lesson is that logical minimality and physical minimality are distinct axes. Minimal languages show that, at the level of pure computation, bit-patterns and universal substrates are all that is needed to realize intelligence. Thermodynamic considerations show that among the many possible substrates and languages, some will support those bit-patterns more gracefully than others, given the constraints of our universe. Informational primacy remains a claim about what structures define an intelligent system; thermodynamic optimality remains a claim about how best to instantiate those structures in matter.

Seen together, they reinforce rather than contradict Wheeler's vision. Bits and their organization determine the high-level character of the "its" we build. The physics of the world determines which patterns are cheap, which are costly, and which are practically unreachable. Minimal substrates testify that no special primitive is needed for intelligence. Thermodynamic substrates remind us that some ways of realizing that intelligence will always be preferable, not because they are more fundamental, but because they fit the physical terms of the universe in which bits must be embodied.

8. From It from Bit to It from Qubit (and Back Again)

So far, the argument has been framed in classical terms: bits, NAND gates, Turing machines, and minimal universal languages. Wheeler's original slogan, however, grew out of quantum physics, and later developments in quantum information theory sharpened the idea: if "it from bit" was a first pass, "it from qubit" is the refinement that takes superposition, entanglement, and non-classical correlations seriously. For questions about intelligence, this raises two related issues. First, how does quantum information alter, complicate, or strengthen an informational reading of reality? Second, does artificial intelligence itself have to go quantum in order for It from Bit—or It from Qubit—to be more than a classical metaphor?

This section approaches those issues in four steps. It starts by sketching how quantum information theory refines Wheeler's picture of bits and measurements. It then considers quantum analogues of classical minimal languages, where universality is achieved with small gate sets or Hamiltonians that manipulate qubits instead of bits. Next, it explores the notion of "AI from Qubit," asking how quantum substrates might host or accelerate intelligent processes, and what that means for informational primacy. Finally, it suggests criteria for when specifically quantum structure matters for the It-from-Bit story, and when classical universality is already enough to support the core claims about intelligence.

8.1 Quantum information as a refinement of Wheeler's picture

Quantum information theory generalizes the classical notion of a bit. A classical bit is either 0 or 1; a qubit is a two-level quantum system that can exist in superpositions of 0 and 1 and can be entangled with other qubits. Measurement in a given basis yields classical bits of outcome, but the underlying state space and the rules for its evolution are richer. Where classical information theory deals with probability distributions over discrete symbols, quantum information theory deals with density matrices, unitary evolutions, and completely positive trace-preserving maps.

For Wheeler's slogan, this refinement is both a challenge and an opportunity. If the universe is fundamentally quantum, then "bit" in It from Bit cannot simply mean classical bit. The fundamental informational degrees of freedom look more like qubits and their entanglement patterns than like simple 0/1 variables. Measurement events still yield discrete outcomes—yes/no answers—but these are now understood as the classical shadows of deeper, non-classical informational structure. The elementary acts of observer-participancy that Wheeler emphasized can be recast as quantum measurements whose outcomes update an observer's classical information about an underlying quantum state.

Quantum information also sharpens the link between physical law and information-processing capabilities. The no-cloning theorem, for example, places constraints on how quantum states can be copied; entanglement monogamy constrains how correlations can be shared. These are informational constraints that have no classical analogue, and they show up directly in what kinds of computation and communication are possible in a quantum world. From this vantage point, It from Bit was prescient but incomplete; It from Qubit better captures the actual structure of our best physical theories.

For AI, the immediate implication is that any truly fundamental informational ontology must accommodate quantum structure. Even if current AI systems are largely classical devices manipulating classical bits, they operate in a quantum world and may eventually be outperformed or qualitatively transformed by quantum-enhanced architectures. The basic move—describing agents and environments informationally—still applies, but the informational substrate now has more room for subtlety.

8.2 Classical minimal languages and their quantum analogues

Just as classical computation has minimal universal formalisms, quantum computation does as well. In the quantum setting, universality typically means that a small set of quantum gates, acting on qubits, can approximate any unitary transformation on a finite number of qubits to arbitrary accuracy. Commons examples include gate sets built from single-qubit rotations and a two-qubit entangling gate such as CNOT. These play a role analogous to NAND in classical digital logic: a small, fixed toolkit from which arbitrarily complex quantum algorithms can be constructed.

There are quantum analogues of minimality in other styles as well. Hamiltonian-based models show that certain simple, local Hamiltonians are universal for quantum computation when appropriately initialized and measured. Measurement-based quantum computation exhibits universality using a fixed entangled resource state (such as a cluster state) and sequences of adaptive single-qubit measurements. In each case, a simple primitive setup suffices to generate any desired quantum evolution, provided one is willing to orchestrate the sequence of operations precisely.

The comparison with classical minimal languages is instructive. On the classical side, NAND, SUBLEQ, lambda calculus, and SKI combinators show that universality is compatible with extreme syntactic austerity. On the quantum side, small universal gate sets and simple Hamiltonians show that the same is true for qubit dynamics. The emphasis shifts from functionally complete Boolean operations to sets of unitaries that densely generate the unitary group on multi-qubit Hilbert spaces. The "language" here is not a Boolean algebra but a Lie group or algebra plus a gate set that spans it.

From an informational primacy standpoint, both families of minimal formalisms support the same general conclusion. Whether the substrate is classical or quantum, there exist very sparse sets of primitive operations that suffice to express any process allowed by the corresponding theory. That is, minimal qubit languages show that no special "intelligence gate" is needed in the quantum case either; standard primitives for manipulating qubits are enough to implement any algorithm, including those we might use to build AI systems.

8.3 It from Qubit and the possibility of "AI from Qubit" substrates

The natural next question is whether quantum substrates can host forms of AI that change the It-from-Bit story in substantive ways. One possibility, often discussed in theory, is that certain AI-relevant computations—search, optimization, simulation—could be dramatically accelerated by quantum algorithms. Grover-style search offers quadratic speedups; quantum approximate optimization algorithms promise advantages on some combinatorial problems; quantum simulation is naturally suited to quantum hardware. If parts of an AI system's workload are dominated by such tasks, quantum acceleration could make previously infeasible learning or planning strategies practical.

In that sense, one can speak of "AI from Qubit" in a pragmatic way: intelligent agents implemented on quantum hardware, exploiting quantum speedups and entanglement to achieve higher performance. The agent's abstract definition in terms of input–output and learning behavior remains informational. What changes is the class of resource-feasible algorithms available. A quantum agent might, for example, perform more extensive lookahead in a game tree, or learn certain classes of functions with fewer examples, if appropriate quantum subroutines are available.

There is also a more speculative angle. Some have suggested that certain cognitive phenomena might require inherently quantum resources—for instance, exploiting quantum coherence or entanglement in ways analogous to proposals in quantum biology or quantum consciousness theories. Under such proposals, an "AI from Qubit" substrate would not merely accelerate classical algorithms but might realize qualitatively new modes of information processing, perhaps with different generalization properties or representational capabilities.

For the purposes of this paper, it is not necessary to decide how far such speculation can be justified. The key point is more modest: quantum substrates extend the space of possible informational structures and transformations. If one accepts informational primacy, then any advantage conferred by quantum structure is an advantage in the richness of the informational dynamics the universe permits. It from Qubit becomes a statement about the kinds of computational patterns that are physically realizable, and AI from Qubit is one particularly vivid way those patterns might be instantiated in agents.

8.4 When quantum structure matters and when classical universality suffices

Given this landscape, it is important to distinguish two levels of the It-from-Bit (or It-from-Qubit) argument. At the most abstract level, the claim is that complex "its"—including intelligent agents—are best understood as patterns of information processing instantiated in physical substrates. For that claim, classical universality already suffices. Minimal classical languages and substrates show that bit-level descriptions are enough to capture the structure and dynamics of AI systems as we currently build them. Whether the underlying physics is classical or quantum, the equivalence classes of implementations defined by input–output behavior and learning dynamics can be described in purely classical informational terms.

Quantum structure becomes crucial when one asks more fine-grained questions: What are the ultimate limits on computation and communication? Which patterns are physically realizable within given time, energy, and noise budgets? How do entanglement, coherence, and nonlocal correlations change what can be done? At this level, It from Qubit, rather than It from Bit, is the appropriate slogan, because the detailed rules governing information are quantum, not classical. An informational ontology that ignores qubits will miss constraints and possibilities that are real features of our world.

For AI specifically, one can say that classical universality suffices to establish informational primacy in principle, while quantum structure refines what is feasible and efficient in practice. A classical Turing machine or a NAND-based CPU can implement any agent definable in the abstract sense used earlier. No conceptual ingredient is missing. However, a quantum-enhanced substrate may allow some of those agents—or entirely new ones—to operate within resource regimes that would be inaccessible classically. That is a difference in engineering and feasibility, not in the basic nature of intelligence as an informational pattern.

In this light, It from Bit and It from Qubit are not competing slogans so much as nested ones. At the coarsest level, where the focus is on which kinds of organization define agents and other complex systems, bits and their arrangements provide a sufficient language. At the finer level, where the focus is on which arrangements are physically realizable, with what costs and correlations, qubits and quantum laws provide the governing rules. Minimal classical languages demonstrate that intelligence does not depend on any special classical primitive. Minimal quantum gate sets demonstrate that, even in the richer quantum setting, a small repertoire of operations suffices.

Taken together, they reinforce the central thesis of this paper. Whether the substrate is classical or quantum, whether the gate is NAND or a two-qubit entangler, the "it" of intelligence is an emergent invariant of informational structure. Quantum physics deepens and complicates the picture, but it does not reverse the direction of explanation. If anything, it makes It from Bit—or better, It from Qubit—harder to avoid.

9. Objections and Alternative Readings

Any attempt to treat minimal computational languages as evidence for informational primacy invites pushback from several directions. Physicists may insist that bits are derivative properties of underlying matter and fields, not fundamental entities in their own right. Philosophers may worry that computation is far too coarse a notion to function as a metaphysical base. Many will point out that intelligence is not the only thing at stake: consciousness and qualia seem stubbornly resistant to purely computational characterization. Others will suggest that It from Bit is better read instrumentally, as a useful organizing metaphor for science and engineering rather than as a literal claim about what the world ultimately is.

This section collects these objections and sketches ways of responding to them that are compatible with, but do not require, the stronger forms of Wheeler's slogan. The goal is not to eliminate all disagreement but to clarify what exactly the minimal-substrate and AI arguments support, and where they genuinely run out of steam.

9.1 "Bits are just properties of matter": reversing the explanatory arrow

A straightforward objection to It from Bit runs as follows. Bits do not float in a void. They are realized by physical systems: voltages across transistors, spin states, photon polarization, magnetic domains. The easiest way to understand bits is as coarse-grained descriptors of matter and fields. On this view, the explanatory arrow runs from “it” to “bit”: the world is fundamentally made of stuff, and information is a useful abstraction over that stuff. Minimal computational substrates, including NAND-only machines, do not change this; they are just particularly simple arrangements of matter.

In this reading, universality and minimality are facts about how flexible matter can be, not about the independence of information. The fungibility of substrate is interpreted as evidence for the versatility of physical law: the same high-level behavior can be realized by many different arrangements of matter because the underlying laws are permissive. The fact that AI architectures can be ported across substrates simply shows that some patterns of matter are dynamically stable and easy to reproduce, not that those patterns are ontologically primary.

A moderate information-first view can accommodate much of this criticism. It can concede that information always requires a physical carrier, while still insisting that the most fruitful level of description for many purposes is informational. The minimal-substrate story then becomes a claim about explanatory efficiency. When analyzing intelligence, it is often more illuminating to talk about state spaces, transition rules, and learning dynamics than about semiconductor band structures or quantum field modes. Bits are not mystical entities; they are roles that parts of matter can play. The claim is that, for intelligence, those roles capture what is essential.

A stronger information-first view would push further. It would suggest that even matter and fields are best understood as manifestations of underlying informational structure, with physical models functioning as effective theories of that structure. Minimal universal substrates strengthen this suggestion by showing that once certain informational properties are present—discrete states, stable distinctions, controllable transitions—the space of emergent behaviors, including intelligence, becomes largely independent of microscopic details. The explanatory arrow can then run both ways: from matter to bits at the level of realization, and from bits to matter at the level of structural constraints.

9.2 The worry that computation is too coarse to underwrite metaphysics

Another objection targets the use of computation as a metaphysical foundation. Computation, understood as symbol manipulation according to rules, is a human-defined, idealized concept. It abstracts away from many features of physical systems: continuity, chaos, analog variables, stochasticity. To say that “the world is computation” or “it comes from bit” may therefore seem like a category mistake: confusing a modeling tool with the thing being modeled.

Minimal universal languages might make this worry sharper. Turing machines, lambda calculus, and NAND-based CPUs are deliberately austere models; they are valuable precisely because they ignore most of what makes physical systems messy. Using them as evidence about the ontology of reality might look like leaning too heavily on a caricature.

One way to respond is to separate two roles computation plays. As a model, it indeed strips away detail, but it also identifies structural invariants: state spaces, transition relations, compositional rules. These invariants are not arbitrary. They are shared by many physical systems, which is why computational modeling works. From this perspective, computation is not the whole story, but it reveals a layer of organization that is genuinely present in the world.

The minimal-substrate results fit this picture. They do not show that the world is literally a Turing machine or a cellular automaton. They show that a wide class of dynamical systems, including those we use to build AI, have computational skeletons that can be expressed in many equivalent formalisms. When one is asking specifically about intelligence, these skeletons capture the aspects that matter most: how information is stored, updated, and used to guide action.

A more cautious metaphysical conclusion, then, is that computation provides a privileged language for describing the organizing patterns of certain complex systems, not that it exhausts their reality. Minimal languages highlight the robustness and generality of this language. They make a strong case that if one wants to talk about what an AI system is, in a way that transcends particular hardware, computational description is not optional. Whether that elevates computation to a fundamental ontological category is a further step—one that some may take, but that the AI and minimality arguments alone do not force.

9.3 Consciousness, qualia, and whether computation alone is enough

A third objection focuses not on intelligence but on consciousness and qualia. An AI system might behave intelligently without being conscious. If It from Bit is meant to be a comprehensive metaphysical thesis, it must address not only the functional, externally observable aspects of systems but also whatever inner, subjective aspects exist. Critics will argue that computation, especially as formalized in minimal languages, has nothing to say about qualia. It describes what systems do, not what it is like to be them.

Minimal computational substrates are ill-equipped to answer this objection. They show that intelligence, as defined by learning and behavior, can be realized on many kinds of hardware. They say nothing about whether different realizations share the same experiential properties, or whether any of them have experiences at all. The substrate-independence of functional patterns is clear; the substrate-(in)dependence of consciousness is not.

There are at least three positions one might take here. A functionalist position would treat consciousness as another emergent invariant of certain complex informational patterns. On this view, if two systems share the right computational organization, they share conscious states, regardless of substrate. For the functionalist, the minimal-substrate argument for intelligence naturally extends to qualia.

A substrate-dependent position would insist that certain physical features—continuous dynamics, particular quantum effects, specific biological properties—are essential for consciousness. On this view, two systems can be functionally equivalent as information processors yet differ in their experiential properties. Minimal substrates might be sufficient for intelligence but still lack whatever is needed for consciousness. In this case, It from Bit might be true for behavior but false or incomplete for experience.

A third, more agnostic position would leave consciousness out of the It-from-Bit argument entirely. It would treat the informational analysis as a powerful tool for understanding the public, functional aspects of systems, and regard questions about subjective experience as orthogonal or currently beyond reach. Minimal substrates would then be evidence about the structure of intelligence, not about the nature of mind in the richest sense.

This paper aligns most naturally with the third stance. Its claims concern intelligence as an informational phenomenon instantiated in universal substrates. They provide a strong case that no special primitive is needed for intelligent behavior. Whether this extends to consciousness is an additional, nontrivial question. The arguments here can be combined with functionalist views to support a stronger thesis, but they do not, by themselves, resolve the issue.

9.4 Instrumental versus ontological readings of It from Bit

Finally, there is a more general objection about how to read Wheeler's slogan and related claims. One can treat "It from Bit" instrumentally, as a guiding metaphor for how to organize scientific theories and engineering practice: focus on information flows, measurement outcomes, and computational structure because those are what can be quantified and controlled. Alternatively, one can treat it ontologically, as a literal claim about what the world is fundamentally made of.

The instrumental reading sits comfortably with the minimal-substrate and AI arguments. It says: for building and understanding artificial intelligence, it is overwhelmingly useful to treat systems as information-processing entities. Minimal languages show that this treatment can be pushed down to very sparse substrates. The success of this perspective in AI, along with its success in thermodynamics and quantum information, justifies giving information a central role in our conceptual toolkit. There is no need to commit to any strong metaphysical claim.

The ontological reading is bolder. It says: the reason information is such a powerful organizing concept is that the world really is informational at its core. Bits (or qubits) and their structure are what exist; matter and fields are emergent. Minimal substrates then become not just engineering curiosities but glimpses of how little is needed, in informational terms, to generate the rich “its” we observe, including intelligent agents.

The arguments in this paper support the instrumental reading decisively. They show that treating AI as an informational phenomenon is not only possible but necessary for cross-substrate understanding, and that minimal languages provide especially clear testbeds for this treatment. They offer the ontological reading a boost by removing one class of objections: no special “intelligence stuff” is needed at the primitive level; bits and their dynamics suffice for the aspects of mind that AI currently captures.

However, they do not force the ontological reading. One can accept everything said here and still regard matter, fields, or something else as ontologically primary, with information as a powerful but derivative abstraction. Conversely, one can embrace an informational ontology but disagree about whether computation, as formalized in current theory, fully captures it.

A reasonable conclusion is that minimal substrates for intelligence narrow the space of plausible views. They make it difficult to maintain that intelligence depends on some irreducible material essence beyond informational organization. They strengthen the case for treating information as at least co-fundamental with more traditional physical categories. Whether one steps from there to a full “It from Bit” metaphysics is, in the end, a philosophical choice. The evidence from AI and minimal languages does not decide the choice, but it makes one side of it more attractive than it might otherwise have been.

10. A Logico-Informational Case for It from Bit

By itself, the mathematical theory of computation is neutral about what the world is made of. It shows that certain abstract machines can emulate others, that small sets of primitives can be universal, and that algorithms can be encoded as finite strings over finite alphabets. What this paper has argued is that, when those facts are combined with the empirical reality of artificial intelligence, they begin to look like more than mathematical curiosities. They become evidence that, at least for a large and important class of “its” in the world—namely artificial agents—what is essential is informational pattern, not material substrate.

The core of the argument runs through three steps. Minimal substrates show that many details of matter can be collapsed, for the purposes of hosting intelligence, into a thin layer of computational roles. AI systems, when viewed through the lens of input–output behavior and learning, reveal themselves as patterns that can be instantiated across a wide range of underlying “stuff.” And information, in the precise sense of bit-level structure and transformation, emerges as the stable invariant that unifies these heterogeneous realizations.

Taken together, these points make a logico-informational case for reading Wheeler's slogan as more than poetry, even if they stop short of a full metaphysical proof.

10.1 Minimal substrates and the collapse of material specificity

Minimal universal substrates perform a kind of conceptual compression on the space of possible physical devices. A NAND-only gate library, a one-instruction set machine, a lambda calculus interpreter, a combinatory system, or a universal cellular automaton can all be implemented in many kinds of matter: different semiconductor processes, optical setups, mechanical relays, even exotic media. Yet from the standpoint of computation, each substrate is fully characterized by a short description: its primitive operations and how they compose.

This collapse of material specificity has two aspects. First, it shows that many microscopic differences between realizations can be ignored when one is interested in the abstract computational behavior. Whether a NAND gate is built from a particular transistor topology or from a different nanotechnology does not matter to the logic of a program, as long as the truth table is preserved. Second, it shows that even quite different primitive sets are interchangeable at the level of what can be computed. A system defined in terms of NAND can be simulated by one defined in terms of SUBLEQ, and both by a combinatory logic, up to systematic encoding.

In this sense, minimal substrates testify that much of what distinguishes one piece of hardware from another is incidental to its ability to host complex processes. Universality strips away the particulars and leaves a thin interface described in informational terms. When intelligence is layered on top of this interface, the material bases of the underlying machines become spectators rather than protagonists in the story of what the system can, in principle, do.

10.2 AI as a test case for pattern over substance

Artificial intelligence sharpens this collapse into a test case for pattern over substance. Unlike many physical systems, AI systems are explicitly designed and specified. Their architectures are written as code; their training procedures are algorithms; their learned models are arrays of parameters. When practitioners say that two implementations are "the same model," they mean that they realize the same mapping from inputs to outputs, the same training dynamics, and the same performance characteristics, even if they run on different hardware.

This practical notion of sameness is inherently informational. A neural network architecture is defined by its connectivity and activation functions, not by the particular logic cells that implement its multiply–accumulate operations. A planning algorithm is defined by its search strategy and heuristic, not by the specific memory technology used to store its frontier. In each case, the pattern of computation is the identity condition; the substrate is a changeable support.

Minimal computational languages stress-test this identity condition. If the same AI architecture can, in principle, be compiled down to a one-instruction machine, a lambda term, or a cellular automaton configuration, then whatever makes it that architecture must be independent of any of those particular embodiments. The architecture is a pattern that floats above the substrate. Intelligence, defined as the ability to learn, generalize, and act effectively, becomes a property of such patterns rather than of any one kind of physical machinery.

AI thus offers a concrete domain in which pattern over substance is not just a philosophical slogan but an operational practice. Systems are moved from one hardware platform to another without loss of identity; their essence is carried by bit strings and transformation rules. Minimal universality shows that this could be taken even further than current engineering practice actually does.

10.3 Information as the stable invariant across heterogeneous implementations

Once one focuses on pattern rather than substance, information appears as the natural language in which those patterns are expressed. Architectures, algorithms, models, and training procedures can all be written as finite descriptions over a fixed alphabet. Their execution traces can be captured as sequences of states and transitions. Differences between implementations can be described as different encodings of the same abstract process on different universal machines.

What remains stable across these encodings is informational. The structure of a computation can be represented as a control flow graph, a transition system, or a function from histories to actions. Learning changes the informational content of an agent's internal state in lawful ways, moving it from unstructured randomness toward structured regularities that reflect the environment. Even thermodynamic considerations enter through informational quantities such as entropy, error rates, and redundancy.

In this framework, information is not an incidental label attached to underlying "stuff." It is the glue that allows us to recognize the same agent, the same algorithm, or the same model across different embodiments. If two systems are regarded as implementing the same AI, it is because there exists an informational mapping between their states and operations that preserves the relevant structures. The choice of substrate matters only insofar as it constrains which informational invariants can be realized with what efficiency.

Minimal languages clarify this by stripping away any features that are not strictly necessary to support such invariants. They reveal a floor at which everything that makes an AI system what it is can be captured in bit-level patterns and their dynamics. Whatever changes below that floor is implementation detail; whatever persists across it is, by construction, informational.

10.4 Why this does not prove, but strongly motivates, Wheeler's view

Strictly speaking, none of this proves that Wheeler's It from Bit is the correct metaphysical picture of reality. The existence of minimal universal substrates, the portability of AI architectures, and the stability of informational invariants across implementations are all compatible with a more traditional view in which matter and fields are ontologically basic, and information is a powerful but derivative abstraction.

What the logico-informational case does, however, is shift the burden of explanation. It makes it difficult to maintain that complex entities like artificial intelligences are best understood in material terms. Their design, behavior, and identity conditions are captured far more naturally and completely in informational terms. Their dependence on particular substrates is loose; their dependence on informational structure is tight. When a framework captures the essence of a phenomenon so well, it is natural to ask whether that framework might be getting at something fundamental.

Moreover, the same informational concepts that clarify AI also play central roles in thermodynamics, communication theory, and quantum information. Entropy, channel capacity, error correction, and entanglement are all informational quantities that constrain physical processes. The fact that such concepts bridge engineering and fundamental physics strengthens the suspicion that bits and qubits are not just convenient bookkeeping devices but reflect deep features of how the world is organized.

A cautious conclusion, then, is that minimal substrates for intelligence make It from Bit an empirically and theoretically attractive stance. They show that no special "intelligence substance" is needed beyond universal information-processing capabilities. They reveal information as the stable invariant across diverse physical realizations of complex behavior. They connect naturally with existing informational structures in physics. From there, one can reasonably entertain the idea that information is not merely a way we describe the world, but one of the primary kinds of "stuff" out of which the world, and the minds within it, are made.

11. Outlook: Physics with AI in the Loop

If minimal substrates and artificial intelligence support a stronger reading of Wheeler's slogan, they also suggest a shift in practice: from physics as a human-only endeavor that occasionally borrows computational tools, to physics as a discipline in which AI systems are themselves active participants in the search for informational structure. In such a setting, the distinction between "using information to describe the world" and "the world arising from information" is no longer purely philosophical. It becomes a live, iterative process in which new informational models are proposed, tested, refined, and sometimes executed by machines whose own operation embodies the principles under study.

This final section sketches an outlook along four lines. First, it considers how AI and minimal languages can be used to probe informational foundations more systematically than has been possible by hand. Second, it outlines experimental and theoretical directions in which AI tools, minimal substrates, and It-from-Bit ideas can reinforce one another. Third, it extends the discussion to It from Qubit, Life from Qubit, and broader informational cosmology, suggesting that AI may serve as a bridge between these domains. Finally, it closes with reflections on bit, it, and future intelligences, human and artificial, as co-inhabitants of an informational universe.

11.1 Using AI and minimal languages to probe informational foundations

One immediate application of the perspective developed in this paper is methodological. If information and computation play central roles in physics, then AI systems running on minimal or near-minimal substrates can be used as experimental probes of informational structure. Instead of treating AI as a black-box tool for curve fitting, one can treat it as a tunable, inspectable realization of complex information processing in a controlled environment.

At a basic level, this might involve using AI to explore the space of minimal universal formalisms themselves. Automated theorem provers and program synthesizers can search for new one-instruction sets, new combinatory bases, or new cellular automata with universal behavior, subject to constraints on simplicity, symmetry, or physical plausibility. The results could feed back into our understanding of how sparse the primitives of computation can be while still supporting rich emergent behavior, which in turn informs It-from-Bit discussions about how little structure is needed at the bottom.

More ambitiously, AI systems can be trained to discover or compress physical laws in informational form. Given data from simulations or experiments, an AI tasked with finding compact, predictive models amounts to an agent searching for short descriptions in the sense of Kolmogorov complexity and Minimum Description Length. When such systems succeed in recovering known laws, or in proposing new candidate laws, they offer concrete examples of information structuring itself in ways that mirror the organization of the physical world.

Minimal languages are useful here because they make the mapping from description to device transparent. An AI system that discovers a compact law or algorithm can itself be expressed in a minimal formalism, making explicit how the informational pattern is realized. Experiments can then vary the substrate while keeping the informational content of the AI fixed, probing how physical constraints and noise shape the expression of informational structure without changing its essence.

In this way, AI and minimal substrates become instruments for "informational microscopy": tools that let us vary implementation details while holding patterns constant, and thus observe how robust those patterns are as candidates for fundamental structure.

11.2 Possible experimental and theoretical directions linking AI and It from Bit

Several more concrete directions suggest themselves at the intersection of AI, minimal computation, and informational physics.

One line of work would involve building AI systems that operate as meta-modelers: agents that search over spaces of candidate dynamical laws, expressed in minimal languages, to explain given data. By constraining the hypothesis space to programs in a one-instruction set machine, a small quantum gate set, or a universal cellular automaton rule space, one can ask how often the best explanatory models exhibit features reminiscent of known physics: locality, conservation laws, symmetries, or specific scaling behaviors. If such features repeatedly emerge as informationally efficient structures, this would strengthen the case that they are not arbitrary accidents of our universe, but natural attractors in the space of compact descriptions.

Another direction is experimental and architectural. One can design physical devices that are intentionally close to minimal universal substrates, then embed AI agents within them and study their behavior under thermodynamic and noise constraints. For instance, a reversible computing platform built around a small reversible gate set could host learning algorithms designed to minimize information erasure. Observing how such agents adapt their strategies under strict energetic budgets could illuminate trade-offs between informational efficiency and physical cost, making tangible the link between bit-level organization and thermodynamic law.

A third direction involves using AI to explore the boundaries between classical and quantum informational structure. Hybrid classical–quantum models that attempt to learn or simulate quantum systems can be analyzed from an It-from-Qubit perspective: which parts of the system's behavior can be captured by classical informational patterns, and which resist such reduction? Minimal quantum gate sets and Hamiltonians can serve as testbeds for such explorations, with AI systems searching for compact classical surrogates where possible and identifying irreducibly quantum patterns where not.

Finally, there is a conceptual direction: using AI-generated models and simulations as laboratories for thinking about observer-participancy. An AI agent interacting with a simulated environment and updating its internal model of that environment is, in effect, enacting Wheeler's picture of bits arising from questions and answers. Studying the informational trajectories of such agents may yield insights into how observer-dependent descriptions cohere into stable, shared "its," both in artificial worlds and, by analogy, in the universe at large.

11.3 Implications for It from Qubit, Life from Qubit, and informational cosmology

The It-from-Bit argument, especially in its quantum-refined form, naturally extends beyond computation and AI to questions about life and cosmology. If information plays a central role in physics and in intelligence, it is tempting to ask whether biological organization and cosmic structure might also be best understood in informational terms: as "Life from Qubit" and "Cosmos from Qubit" themes.

Artificial intelligence can contribute here in two ways. First, as a modeling tool. AI systems trained on biological data can search for compact, predictive models of evolutionary dynamics, gene regulation, or neural processing. When such models succeed, they provide candidate informational structures that may underlie living organization. Expressing these models in minimal languages or quantum-inspired formalisms could reveal whether biological complexity admits particularly simple generative descriptions, and whether those descriptions have an intrinsically quantum flavor.

Second, AI can act as a generator of synthetic "worlds" in which informational cosmology can be explored experimentally. By designing artificial environments with different underlying rules—some classical, some quantum-like, some entirely novel—and then observing how agents evolve, learn, and organize, one can ask which kinds of informational substrate give rise to robust, scalable "life-like" phenomena. Does complex adaptive behavior require specific kinds of locality or entanglement? How do different informational geometries shape the emergence of structure?

Such experiments would not prove that our universe is an instance of a particular informational cosmology, but they would map out the space of possibilities. If, for example, complex adaptive behavior repeatedly emerges in environments governed by simple, local, qubit-like rules, and struggles to emerge in radically different informational settings, that pattern of results would be suggestive. It would indicate that the combination of It from Qubit and Life from Qubit is not arbitrary, but occupies a privileged region in the space of informationally parsimonious, life-supporting worlds.

In this speculative but disciplined way, AI becomes a bridge between minimal computation, quantum information, biology, and cosmology. It allows us to ask, not in the abstract but through concrete constructions, which informational substrates are fertile ground for minds and life, and which are barren. The answers may not be definitive, but they can sharpen our intuitions about what It from Bit or It from Qubit really entails when extended beyond physics into the full tapestry of the universe.

11.4 Closing reflections on bit, it, and future intelligences

The logico-informational case developed in this paper has a modest structure and an ambitious implication. Modestly, it observes that minimal universal substrates, classical and quantum, suffice to realize artificial intelligence; that AI architectures travel easily across substrates; and that information is the most stable invariant across heterogeneous implementations. Ambitiously, it suggests that these facts make Wheeler's It-from-Bit slogan more compelling than it might appear when considered only in the context of measurement theory and black holes.

The picture that emerges is layered. At the base are physical systems with their thermodynamic and quantum constraints. On top of that are minimal computational substrates: small sets of primitive operations that can realize any pattern of information processing allowed by those constraints. Above that are architectures, programs, and training procedures, written as finite descriptions and enacted by compilers and interpreters. At the top are agents: systems that learn, predict, plan, and act in ways we recognize as intelligent. At each layer, bits and qubits are not optional bookkeeping devices but the currency in which structure is expressed and transformed.

Future intelligences—whether human, artificial, or hybrid—will likely inhabit and refine this layered picture. They will design new minimal languages, discover new informational structures in physics and biology, and implement agents on substrates we have not yet imagined. Some may take It from Bit as a working hypothesis; others may treat it as a metaphor. In either case, their practice will continue to test whether information truly deserves the central role this paper has sketched.

If the arguments here are on the right track, then the long-term trend will be toward increasing recognition of pattern over substance. As more "its" are built, understood, and translated across substrates, the grip of any one material picture will loosen, and the appeal of informational primacy will grow. Whether that culminates in a full informational metaphysics or in a more nuanced pluralism remains to be seen. What is already clear is that bits—and their quantum generalizations—will not disappear from our deepest accounts of the world. They are woven into the fabric of the devices we build, the theories we propose, and the minds that ponder both.

12. References

The references below are grouped by theme. They are not exhaustive, but they gather key sources on computation and minimal universality, information and complexity, thermodynamics of computation, quantum information and "It from Qubit," artificial intelligence and philosophy of mind, and informational cosmology, along with the companion preprint that motivated this paper.

12.1 Computation, universality, and minimal formalisms

- Church, A. (1936). An unsolvable problem of elementary number theory. *American Journal of Mathematics*, 58(2), 345–363.
- Turing, A. M. (1936). On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, 42(2), 230–265.
- Minsky, M. (1967). *Computation: Finite and Infinite Machines*. Prentice Hall.
- Shepherdson, J. C., & Sturgis, H. E. (1963). Computability of recursive functions by means of register machines. *Journal of the ACM*, 10(2), 217–255.
- Shannon, C. E. (1938). A symbolic analysis of relay and switching circuits. *Transactions of the AIEE*, 57, 713–723.
- Cook, M. (2004). Universality in elementary cellular automata. *Complex Systems*, 15(1), 1–40.
- Wolfram, S. (2002). *A New Kind of Science*. Wolfram Media.

(Additional technical sources on one-instruction set computers, combinatory logic, and universal cellular automata can be drawn from the theoretical computer science literature as needed.)

12.2 Information theory, Kolmogorov complexity, and Minimum Description Length

- Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, 27, 379–423, 623–656.
- Kolmogorov, A. N. (1965). Three approaches to the quantitative definition of information. *Problems of Information Transmission*, 1(1), 1–7.
- Solomonoff, R. J. (1964). A formal theory of inductive inference, Parts I and II. *Information and Control*, 7(1–2), 1–22, 224–254.

- Chaitin, G. J. (1969). On the length of programs for computing finite binary sequences. *Journal of the ACM*, 16(1), 145–159.
- Li, M., & Vitányi, P. (1997). *An Introduction to Kolmogorov Complexity and Its Applications* (1st ed.). Springer.
- Rissanen, J. (1978). Modeling by shortest data description. *Automatica*, 14(5), 465–471.

12.3 Thermodynamics of computation and physical limits of information processing

- Szilard, L. (1929). On the decrease of entropy in a thermodynamic system by the intervention of intelligent beings. *Zeitschrift für Physik*, 53(11–12), 840–856.
- Landauer, R. (1961). Irreversibility and heat generation in the computing process. *IBM Journal of Research and Development*, 5(3), 183–191.
- Bennett, C. H. (1973). Logical reversibility of computation. *IBM Journal of Research and Development*, 17(6), 525–532.
- Bennett, C. H. (1982). The thermodynamics of computation—A review. *International Journal of Theoretical Physics*, 21(12), 905–940.
- Feynman, R. P. (1996). *Feynman Lectures on Computation*. Addison–Wesley.
- Leff, H. S., & Rex, A. F. (Eds.). (2003). *Maxwell's Demon 2: Entropy, Classical and Quantum Information, Computing*. Taylor & Francis.

12.4 Quantum information, It from Qubit, and minimal quantum universality

- Deutsch, D. (1985). Quantum theory, the Church–Turing principle and the universal quantum computer. *Proceedings of the Royal Society A*, 400(1818), 97–117.
- Wheeler, J. A. (1990). Information, physics, quantum: The search for links. In W. H. Zurek (Ed.), *Complexity, Entropy and the Physics of Information* (pp. 3–28). Addison–Wesley.
- Nielsen, M. A., & Chuang, I. L. (2000). *Quantum Computation and Quantum Information*. Cambridge University Press.
- Zeilinger, A. (1999). A foundational principle for quantum mechanics. *Foundations of Physics*, 29(4), 631–643.
- Lloyd, S. (2006). *Programming the Universe: A Quantum Computer Scientist Takes on the Cosmos*. Alfred A. Knopf.

(For minimal universal quantum gate sets and Hamiltonian models, standard quantum computing texts and specialized articles on universal gate constructions and measurement-based quantum computation apply.)

12.5 Artificial intelligence, learning, and pattern-based accounts of agency

- Russell, S., & Norvig, P. (2010). *Artificial Intelligence: A Modern Approach* (3rd ed.). Prentice Hall.
- Sutton, R. S., & Barto, A. G. (2018). *Reinforcement Learning: An Introduction* (2nd ed.). MIT Press.
- Hutter, M. (2005). *Universal Artificial Intelligence: Sequential Decisions Based on Algorithmic Probability*. Springer.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
- Pearl, J. (2009). *Causality: Models, Reasoning, and Inference* (2nd ed.). Cambridge University Press.

12.6 Philosophy of computation, mind, and physical information

- Putnam, H. (1967). Psychological predicates. In W. H. Capitan & D. D. Merrill (Eds.), *Art, Mind, and Religion* (pp. 37–48). University of Pittsburgh Press.
- Searle, J. R. (1980). Minds, brains, and programs. *Behavioral and Brain Sciences*, 3(3), 417–457.
- Dennett, D. C. (1991). *Consciousness Explained*. Little, Brown.
- Chalmers, D. J. (1996). *The Conscious Mind: In Search of a Fundamental Theory*. Oxford University Press.
- Piccinini, G. (2015). *Physical Computation: A Mechanistic Account*. Oxford University Press.

12.7 Information, gravity, and cosmological perspectives

- Bekenstein, J. D. (1973). Black holes and entropy. *Physical Review D*, 7(8), 2333–2346.
- Hawking, S. W. (1975). Particle creation by black holes. *Communications in Mathematical Physics*, 43(3), 199–220.
- ’t Hooft, G. (1993). Dimensional reduction in quantum gravity. In *Salamfestschrift* (pp. 284–296). World Scientific.

- Susskind, L. (1995). The world as a hologram. *Journal of Mathematical Physics*, 36(11), 6377–6396.
- Lloyd, S. (2000). Ultimate physical limits to computation. *Nature*, 406(6799), 1047–1054.

12.8 Companion and related work by the present author

- Youvan, D. C. (2025). *Minimal Substrates for Intelligence: NAND, Universality, and the Lowest-Level Languages for AI*. ResearchGate preprint, November 2025. DOI: 10.13140/RG.2.2.18547.80166.

The author has published over 4,600 AI-assisted papers at: https://www.researchgate.net/profile/Douglas-Youvan/research . Google Scholar has indexed these preprints, and if you want a particular reference, the AI mode of a Google search (Gemini) has efficiently catalogued these preprints.