THE IN A HURRY SERIES

Quantum Computing

in a Hurry

Computing with the rules of the very small

Max Brocklesby

BOOKSINAHURRY.COM

Contents

The Whole Thing in One Page

The familiar sales pitch says a quantum computer tries every answer at once. It does something stranger and more constrained. A quantum processor creates a physical state whose possible measurement outcomes carry complex amplitudes. Gates change those amplitudes and their relative phases. When computational paths are brought back together, amplitudes can reinforce or cancel. Measurement then converts the surviving pattern into ordinary classical data.

That last step is the catch. A register of n qubits can require up to 2^n amplitudes for an exact generic state-vector description, but reading the register does not print those amplitudes. One shot produces n classical bits. Repeating the experiment gives samples, not a secret exponential spreadsheet. Quantum algorithms work only when they arrange the state so that a small, measurable feature reveals something useful about the problem.

Three quantum rules do most of the computational work. Superposition allows coherent alternatives rather than an ordinary lottery over hidden classical states. Relative phase lets alternatives later interfere. Entanglement makes the state of several qubits inseparable, so useful information can live in relationships that belong to the register as a whole. None of these properties alone proves speed. They are ingredients from which particular algorithms may build an advantage.

Shor's algorithm turns factoring and discrete logarithms into period-finding problems and uses a quantum Fourier transform to expose periodic structure. Grover's algorithm rotates amplitude towards marked answers, giving a quadratic query advantage for unstructured search. Quantum simulation uses one controllable quantum system to reproduce aspects of another without explicitly storing every amplitude on a classical machine. Other algorithms offer more conditional gains. There is no universal quantum speed button, and no known efficient general quantum algorithm for NP-complete problems.

The same physics creates the engineering difficulty. A useful quantum state is valuable because its phase relations remain coherent. Unwanted coupling to the environment leaks those relations away. An arbitrary unknown state cannot be perfectly copied, so engineers cannot protect it by making ordinary backup copies. Fault-tolerant quantum computing instead encodes one logical qubit across many physical systems and extracts information about errors without extracting the logical answer itself. In the most common schemes this is done with repeated syndrome measurements, though coherent, measurement-free fault-tolerant protocols are also possible.

That distinction, physical qubits versus dependable logical computation, is more important than raw machine size. Superconducting circuits, trapped ions, neutral atoms, photons and semiconductor spins make different bargains among fidelity, speed, connectivity, manufacturability and control. Topological designs aim to push some protection into the hardware, but the evidence for practical topological qubits remains incomplete. By 10 August 2026, experiments had shown below-threshold memories, small logical processors, universal logical toolboxes on limited codes and several narrow frontier computations that strain leading classical methods. They had not produced a broad, routine, application-scale fault-tolerant quantum computer.

Quantum computing is therefore best understood as a controlled conversion. The machine uses quantum coherence to manipulate patterns that are hard to represent classically, then must compress whatever matters into a classical answer before the environment, the measurement process or accumulated errors erase the advantage. Its promise and its difficulty come from the same rules.

That is the book.

Why You Should Care

In 1994 Peter Shor described an attack on important public-key cryptosystems using a machine nobody could build. The algorithm was mathematical, not operational: a sufficiently large fault-tolerant quantum computer could factor integers and solve discrete logarithms in polynomial time. Thirty years later, in August 2024, NIST finalised its first three principal post-quantum cryptography standards and urged organisations to begin migrating. The machine capable of threatening modern key sizes still does not exist. The algorithm mattered anyway because some secrets must remain safe for longer than a hardware transition takes.

Security is the most visible consequence, but it is not the best reason to understand the subject. Quantum computing is where the physical meaning of information becomes impossible to ignore. Ordinary digital machines are built from quantum matter, yet their logic is engineered to forget almost all microscopic detail. A bit is treated as 0 or 1, copied freely, stored redundantly and read without changing the abstraction. Quantum computation deliberately preserves details that classical engineering works hard to suppress: coherent phase, superposition and entanglement.

That makes the field a severe test of clear thinking. An exponentially large mathematical state description is not an exponentially large readable memory. A polynomial or quadratic advantage in one complexity model is not automatically a faster product. A device with more physical qubits is not automatically closer to a useful logical computer. A result beyond one classical method may be within reach of a better tensor-network, Monte Carlo, perturbative or specialised simulation. A peer-reviewed logical-gate experiment may be a major scientific milestone while remaining many orders of magnitude from an application.

These distinctions are not rhetorical caution. They are the subject. Quantum computing forces every claim to specify what is being counted: queries, gates, circuit depth, logical qubits, physical qubits, samples, precision, runtime, energy, classical preprocessing or verification. Change the cost

model and a claimed advantage can change with it.

The field also feeds back into physics. The no-cloning theorem says which information cannot be duplicated. Teleportation shows that an unknown quantum state can be transferred using shared entanglement plus classical communication without sending a copy of the original state. Error correction shows that fragile quantum information can be protected without violating no-cloning. Complexity theory asks which physical transformations appear to outrun the best classical descriptions. These are questions about computation, but they sharpen what physicists mean by information, locality, measurement and control.

There is also a practical reason to care about the present rather than an imagined future. Quantum processors are now good enough that some experiments sit in an awkward middle ground. They can perform tasks or physical simulations for which exact classical calculation is difficult, while the quantum result itself may be hard to verify. July 2026 preprints reported hard-circuit sampling with error-detected fidelity certificates and quantum simulations in regimes where leading classical methods struggled to converge. Those are not settled declarations of commercial advantage. They show instead that the frontier problem is becoming epistemic as well as computational: if the answer is beyond straightforward classical calculation, what evidence should make you trust it?

The strongest lesson is therefore not that quantum computers will replace ordinary machines. They almost certainly will not. It is that computation is a physical process, and every physical process grants some operations cheaply while making others expensive or impossible. Classical computing taught us to ignore the underlying physics. Quantum computing makes the physics part of the algorithm.

Once that clicks, the subject becomes less mystical and more demanding. You stop asking whether a qubit is "both 0 and 1" and start asking what amplitudes were prepared, which phases were changed, what interference pattern was engineered, what information the measurement can expose, how errors are suppressed and what classical method the whole system

must beat. That is enough to separate most serious quantum-computing claims from most noise about quantum computing.

The Core Ideas

1. A Qubit Is a State, Not a Tiny Classical Bit

A classical bit is a commitment. Read it and, allowing for hardware failure, it says 0 or 1. A qubit is a controlled quantum state with two reference outcomes, written `|0>` and `|1>`. Its state can be written as

```
|psi> = alpha|0> + beta|1>,
```

where alpha and beta are complex probability amplitudes and their squared magnitudes add to one. If the qubit is measured in this computational basis, the chance of 0 is `|alpha|^2` and the chance of 1 is `|beta|^2`. The measurement produces one of those results, never a visible blend.

This equation is often translated as “the qubit is both 0 and 1”. The phrase is tolerable as a warning that the state is not restricted to two classical points. It becomes misleading when it suggests that two ordinary answers are stored side by side, waiting to be read. A classical random bit can also be 0 with one probability and 1 with another. The difference is that a qubit carries phase relationships which later operations can expose through interference. A probability distribution describes ignorance about a classical value. A pure quantum state describes a physical preparation with more structure than its immediate measurement statistics reveal.

The Bloch sphere gives a better picture for one qubit. Put `|0>` at the north pole and `|1>` at the south. Every pure state lies somewhere on the surface. Quantum gates rotate the state around the sphere, while noise can shrink its description towards the interior. The sphere is not a tiny object spinning inside the chip. It is a map of the possible states and transformations of a two-level quantum system.

Calling the qubit a direction also fixes another common confusion. The labels 0 and 1 are a chosen measurement basis, not two universal

substances. A photon may encode them in two paths or two polarisations. An ion may use two internal energy levels. A superconducting circuit may use its ground and first excited states. An electron spin may use two orientations relative to a field. The physical systems differ, yet each provides a pair of controllable, distinguishable states on which the same abstract qubit mathematics can act.

Change the measurement basis and the same state tells a different story. The state called `|+>`, halfway around the Bloch sphere from `|0>` to `|1>`, gives 0 or 1 with equal probability in the computational basis. Measured in the X basis, it gives the plus result with certainty. Quantum information therefore lives partly in relations among possible measurements, not in a concealed answer that every apparatus is trying to reveal.

This is why counting states needs care. One qubit uses two amplitudes. Two qubits use four. An n-qubit pure state can require 2^n amplitudes to describe. At 300 qubits, a generic state has more amplitudes than there are estimated particles in the observable universe. That is a statement about the size of the mathematical state space, not the memory capacity of a readable quantum hard drive. Measure those 300 qubits once and the machine returns 300 classical bits.

The computational opportunity begins in the gap between those facts. The state is costly for a classical computer to track, while the quantum device evolves it directly. Yet the final readout remains small. Every useful quantum algorithm must use the large internal state space without demanding that the machine print it.

2. Phase Becomes Information Through Interference

Prepare two qubits separately in these states:

```
|+> = (|0> + |1>)/sqrt(2)
```

```
|-> = (|0> - |1>)/sqrt(2)
```

Measure either one immediately in the computational basis and the statistics

are identical: half 0, half 1 over many trials. The minus sign seems to have vanished. It has not. Apply a Hadamard gate first and |+> becomes |0>, while |-> becomes |1>. A relative phase that no direct readout could see has been converted into a deterministic difference.

That small experiment contains the working principle of the whole subject. Quantum gates manipulate amplitudes, including their phases. Amplitudes for alternative computational paths are added before probabilities are calculated. Paths with matching phase can reinforce. Paths with opposite phase can cancel. The algorithm is successful when this interference pattern moves probability towards outcomes that reveal the answer.

A two-path interferometer makes the arithmetic visible. Send single photons towards a beam splitter and the device creates amplitudes for two routes. Recombine the routes at a second beam splitter. When their relative phase is aligned, one detector can click with certainty and the other never clicks, even though each route considered alone would feed both. Change the path length enough to reverse the phase and the bright and dark outputs swap. No photon has to split into two classical halves. The experiment is governed by amplitudes assigned to the complete alternatives and by how those amplitudes recombine.

This is why “quantum parallelism” is an incomplete explanation. A circuit can apply one operation to a superposition of inputs, producing amplitudes associated with many input-output pairs. If the circuit is measured at once, it returns one pair and discards the rest. Parallel evolution becomes useful only when later operations make the alternatives interfere so that a desired global feature affects the probability of the final samples. Without that step, the impressive internal state is computationally mute.

Phase also explains why some physical errors are damaging even when the qubit has not visibly flipped. A bit-flip error exchanges |0> and |1>. A phase-flip error changes the sign between components. Immediate measurement in the computational basis may miss the phase error entirely, yet a later interference step can turn it into the wrong answer. Quantum hardware must therefore preserve distinctions which ordinary digital

hardware is designed to erase.

There are two important cautions. First, an overall or global phase has no observable effect. Multiplying the entire state by the same complex phase does not change any measurement probabilities. Relative phases between components matter. Second, interference does not guarantee improvement. A badly designed circuit can reinforce useless outcomes as efficiently as a good one reinforces useful outcomes. The quantum laws supply the medium, not the algorithm.

The use of complex numbers can make this sound more mysterious than it is. An amplitude has a size and an angle. A gate can rotate that angle. When two contributions arrive at the same outcome, their arrows add. Aligned arrows lengthen the result; opposed arrows shorten it. Squaring the final length gives probability. The geometry is exact, and it is the part of quantum computation most public accounts omit because “all answers at once” is easier to say.

Keep the minus sign. It is often where the computation is hiding.

3. Gates Choreograph Amplitudes, and Must Remain Reversible

A quantum circuit is a schedule of controlled transformations followed by measurements. Lines represent qubits. Time normally runs from left to right. Boxes on one line are single-qubit gates; linked symbols across lines are multi-qubit gates. The diagram looks like an electronic circuit, but no current flows along the drawn wires. They track the state through a sequence of operations.

The standard gate model imposes a demanding rule. Until measurement, an ideal closed-system transformation is unitary, which means it preserves total probability and can be reversed. The X gate swaps `|0>` and `|1>`. The Z gate changes the phase of the `|1>` component. The Hadamard gate moves between the computational basis and equal superpositions. Controlled gates make one operation depend coherently on another qubit. A small universal gate set can approximate any unitary transformation to arbitrary

precision, much as a limited instruction set can express an enormous range of classical programs.

Reversibility changes how algorithms are built. Ordinary code throws information away constantly. An AND gate can map several inputs to the same output, so the output alone cannot recover the inputs. A quantum circuit must retain enough information to reverse its steps. Programmers introduce ancilla qubits to hold working data, compute a function, transfer the useful effect into phase or another register, then run parts of the computation backwards to clear the ancillas. This clearing is called uncomputation. If unwanted working information remains entangled with the answer, it can prevent the intended interference.

Consider a function evaluated on a superposition of inputs. The machine can produce a joint state in which each input is associated with its output. That is the source of the claim that it evaluated every input at once. Yet reading the output register reveals only one result and usually collapses the input with it. The productive move is to design the function call as an oracle whose action changes phases or correlations in a structured way. Later gates then erase information that does not matter and combine amplitudes according to the property being sought.

Grover's search gives a clean example. An oracle reverses the phase of marked states. A second operation, often described as inversion about the mean, converts that phase mark into a larger amplitude. Repeating the pair rotates the state towards the marked subspace. Too few repetitions leave the answer weak; too many rotate past it. The speedup is therefore an engineered oscillation, not a mass inspection of a database.

Compilation connects the abstract circuit to a device. A high-level algorithm may assume arbitrary interactions and ideal gates. Hardware offers a limited native gate set, a particular connectivity graph and errors that vary by qubit, pair and time. The compiler decomposes the algorithm, routes interactions through available connections and schedules operations around calibration data. Each extra swap or idle period adds noise. Two devices with the same qubit count can therefore execute the same logical circuit with radically

different success.

The circuit model is not the only way to compute quantum mechanically. Measurement-based computation consumes a prepared entangled resource through a sequence of adaptive measurements. Adiabatic and annealing approaches evolve an energy landscape. Analogue simulators imitate a chosen Hamiltonian without compiling every step into gates. These models can be equivalent in principle under suitable conditions or useful in narrower roles, but their resource accounting differs. A raw qubit number does not translate cleanly across them.

The gate model remains the clearest mental frame because it exposes the algorithm's obligation. Prepare amplitudes. Attach phases and correlations to the feature of interest. Uncompute distractions. Recombine paths. Measure only after the circuit has made the answer statistically visible.

4. Entanglement Stores Relationships in the Whole

Take two qubits and prepare the state

`(|00> + |11>)/sqrt(2).`

Measure them in the computational basis and the result is either 00 or 11, each with equal probability. You never see 01 or 10. A classical pair of matching random bits could reproduce those statistics, so correlation alone does not identify what is quantum. The distinction appears when the measurement bases change. No assignment of pre-existing local values can reproduce all the correlations predicted for, and observed from, suitable entangled states.

The computational consequence is simpler to state. The state above cannot be written as one state for the first qubit multiplied by one state for the second. Each qubit lacks a complete pure-state description of its own. The information is in the pair. With many qubits, this inseparability can spread across the register, and operations on a few parts can alter global correlations that a classical simulation may find expensive to track.

Entanglement is therefore less like a high-speed cable between qubits than a failure of the parts list. The register must be described as one object. This is why a quantum processor is not well modelled as a collection of independent probabilistic bits. Once entangled, the qubits do not each carry their own answer. They participate in a joint amplitude structure.

It does not follow that more entanglement always means more useful computation. Some highly entangled states have compact classical descriptions. Clifford circuits can create entanglement yet remain efficiently simulable by classical methods under the stabilizer formalism. Tensor-network algorithms can track many quantum systems efficiently when their entanglement has favourable structure. A quantum circuit becomes classically difficult through the combined pattern of entanglement, non-Clifford operations, depth, geometry and required accuracy. Entanglement is a resource and a burden, not a certificate of speedup.

It is also not a method for controllable faster-than-light communication. Suppose two people share an entangled pair and travel apart. One measures her qubit. Her outcome is random. The distant partner's local results remain random, whatever measurement she chooses. Correlations appear only when they later compare records through an ordinary channel. Quantum teleportation makes the same limit explicit: entanglement helps transfer an unknown state, but the protocol still needs two classical bits to complete the transfer. No usable message outruns light.

Why, then, does entanglement help computation? Because algorithms often need correlations among many candidate configurations. A molecule's electronic state may not divide into independent states for each electron. A factoring circuit may need a control register coherently correlated with values of a modular function. An error-correcting code deliberately entangles physical qubits so that local damage leaves a detectable pattern while the logical state remains hidden. In each case, the machine represents a relation directly rather than reconstructing it from separately stored parts.

The cost is control. Entanglement intended by the circuit is useful.

Entanglement with an uncontrolled environment is decoherence. A stray photon, a fluctuating field or a defect in a material can carry away phase information, making the processor's state less coherent when considered alone. The boundary between computation and noise is therefore a boundary between correlations the operator controls and correlations that escape.

A register becomes powerful when its parts stop being separately describable. It becomes fragile for the same reason.

5. Measurement Converts Quantum Information into Classical Evidence

Every useful quantum calculation must cross a boundary. Inside the processor, the state is described by amplitudes, phases and correlations. Outside, a control computer receives ordinary records: voltage levels, photon detections, fluorescence counts or digitised bits. Measurement is the operation that connects those two descriptions.

The Born rule tells us how the quantum state determines outcome probabilities for a chosen measurement. One run of a circuit, usually called a shot, gives one sample. A second run gives another. To estimate a probability, expectation value or correlation, the experiment may need thousands or millions of repetitions. Sampling is therefore part of the computational budget, not an afterthought drawn after the circuit box.

The measurement basis matters. A qubit that looks random when read in the 0/1 basis can be perfectly predictable in another basis. In practice, hardware often has one convenient native readout, so the circuit rotates the state before measurement. That is how an otherwise invisible relative phase becomes a visible change in statistics. The apparatus does not ask "what is the qubit?" in the abstract. It asks one physical question, in one basis, and gets one answer.

This creates a severe output constraint. An arbitrary pure state of n qubits can need 2^n complex amplitudes for its exact state-vector description. Directly measuring the register yields only n classical bits per shot.

Reconstructing a generic unknown state by tomography requires many measurement settings and resources that grow exponentially with system size. A quantum computer is therefore a poor device for loading an arbitrary huge vector and then asking for every entry back. Useful algorithms target compact outputs: a factor, a period, an expectation value, a sample, an eigenphase, a yes/no property.

The no-cloning theorem blocks the obvious workaround. There is no universal physical operation that takes an arbitrary unknown quantum state and produces two perfect independent copies. Known basis states can be copied. A known preparation procedure can be run again. Restricted or approximate copying is possible. What fails is the classical idea of making a perfect backup of any unknown coherent state before experimenting on it.

Measurement is not, however, a single trapdoor placed only at the end of every quantum algorithm. Modern processors measure ancillas midway through circuits, use the results for feed-forward and continue computing on quantum information encoded elsewhere. Teleportation and many error-correction schemes depend on this. Quantum non-demolition measurements can extract a chosen observable while preserving other useful structure. In 2026, trapped-ion researchers also demonstrated a small fault-tolerant universal logical toolbox that avoids mid-circuit measurement entirely by moving syndrome information coherently into auxiliary qubits and feeding it back with gates.

The durable distinction is therefore not "measurement destroys quantum information". It is more precise. Creating a classical record of a quantum observable changes the information available for subsequent coherent interference, and it exposes only the chosen observable rather than a complete hidden state. Good protocols decide carefully what may be learned, what must remain coherent and where the classical record should enter the control loop.

That is why quantum algorithms cannot inspect all their branches and keep going as though nothing happened. The useful structure has to be converted into a small set of observables before readout. Measurement is

the interface where a quantum computation becomes evidence.

6. Speedup Belongs to a Problem, Not to the Processor

Quantum computers are often discussed as though “quantum” were a faster clock speed. It is not. A speedup is a statement about an algorithm for a defined problem, a model of access to the input, a required accuracy and a comparison with the best classical alternative. Change any of those and the advantage may shrink, disappear or become irrelevant.

Shor's algorithm is the strongest famous case. Factoring a large integer can be reduced to finding the period of a modular function. A quantum circuit evaluates that function coherently across a superposition, then applies a quantum Fourier transform whose interference pattern reveals information about the period. Classical post-processing recovers the factors with high probability after enough successful runs. The algorithm uses resources polynomial in the number of input bits, whereas no polynomial-time classical factoring algorithm is known. The related discrete-logarithm algorithm threatens elliptic-curve systems as well as older public-key schemes.

The qualification is physical. Shor's circuit for a cryptographically relevant key would require a large number of low-error logical operations. Each logical qubit and gate may consume substantial physical-qubit, time and decoding resources. Demonstrating small factorizations on present devices does not amount to attacking deployed cryptography, since classical shortcuts can solve those toy instances more cheaply. The algorithm changed security planning because its asymptotic result is serious, not because today's machines are breaking keys.

Grover's algorithm offers a different kind of gain. It searches an unstructured set of N possibilities using on the order of the square root of N oracle queries rather than N. That is a quadratic improvement, valuable for some large searches but not an exponential one. The oracle must be implemented, data must be accessible in the assumed way and the candidate answer must be checked. If building the coherent query costs as

much as scanning the data classically, the headline speedup does not survive the complete system.

Quantum simulation is the most native application. A classical computer tracking a generic quantum state can face an exponentially growing description. A controllable quantum processor evolves according to quantum rules without storing every amplitude as a classical number. Feynman's original motivation was that nature might be best simulated by nature-like machinery. Digital simulation algorithms, phase estimation and analogue simulators all pursue this idea. The hard questions are which observables can be prepared and measured, how errors accumulate and whether a classical approximation already captures the required physics.

Variational algorithms such as the variational quantum eigensolver and the quantum approximate optimisation algorithm divide work between a shallow quantum circuit and a classical optimiser. They fit noisy hardware and have produced useful experimental testbeds. They have not established a general practical advantage. Optimisation landscapes can become flat, measurement costs can be large, noise can bias the objective and strong classical heuristics may win. “Hybrid” describes an architecture, not a performance result.

Several theoretical quantum speedups rely on input models that deserve inspection. Amplitude estimation can improve the scaling of some Monte Carlo estimates. Linear-system algorithms can prepare a quantum state related to a solution under conditions on sparsity, conditioning and data access. Quantum machine-learning proposals can assume efficient coherent access to classical data. If loading the data or extracting the output dominates, an elegant subroutine may not accelerate the end-to-end task.

There is no known quantum method that efficiently solves every hard problem. In particular, no general polynomial-time quantum algorithm is known for NP-complete optimisation problems. Quantum computers may improve exact algorithms, sampling, heuristics or structured instances without overturning worst-case complexity. Exponential state space is not the same as exponential speedup.

Speedups also come in different strengths. A quadratic improvement can be decisive when the underlying search is enormous, yet modest once error-correction overhead dominates. An exponential separation in a black-box or sampling model can be intellectually important while relying on an input oracle or output distribution that does not map cleanly to a commercial task. A polynomial improvement can matter more in practice if the constants, data access and verification are favourable. Complexity theory tells you how cost grows; engineering tells you where one curve crosses another. A useful claim needs both.

The comparison must also specify success probability and accuracy. Quantum algorithms are often probabilistic, and some scientific tasks accept approximate answers. Classical competitors can exploit the same tolerance. A speedup proved for exact computation may disappear when approximation is allowed, while a quantum method for sampling may retain an advantage precisely because sampling is the task. "Faster" without a metric is not a technical statement.

The right question is never “Can quantum computing solve this?” With enough time, ordinary computers can solve many of the same finite problems. Ask which quantum algorithm applies, what structure it exploits, what resources the full implementation needs and which classical method is being beaten. The adjective belongs to the machine. The speedup belongs to the proof and the benchmark.

7. Fault Tolerance Builds Reliability Without Copying the Data

A classical memory can fight errors by copying a bit three times. If the copies read 001, majority vote restores 0. Try the same operation on an arbitrary unknown qubit and two obstacles appear. The state cannot be cloned, and measuring each copy would reveal and disturb the logical information. Quantum error correction must create redundancy without making independent replicas and diagnose damage without learning the encoded state.

The solution is to encode one logical qubit into an entangled state of several physical qubits. Different physical patterns represent the logical |0> and |1>, while a superposition of those codewords preserves the unknown amplitudes. Extra ancilla qubits interact with selected combinations of data qubits. Measuring the ancillas reveals a syndrome: information about whether particular parity checks have changed. The syndrome can identify likely errors without revealing whether the logical state was 0, 1 or any superposition between them.

Quantum codes must handle more than bit flips. Phase flips matter too, and general single-qubit errors can be decomposed into combinations of bit and phase errors. Repeated syndrome rounds are needed because the measurements and ancillas can themselves fail. A decoder uses the pattern across space and time to infer the most likely chain of faults and choose a correction, sometimes by updating a classical record rather than physically applying a gate.

The surface code is prominent because it uses local checks on a two-dimensional grid and tolerates comparatively high physical error rates. Its logical operators stretch across a patch. The code distance measures the shortest undetectable error chain; increasing distance gives the decoder more evidence and requires more faults to cause a logical failure. This improvement occurs only below the code's threshold. If physical operations are too noisy or errors are too strongly correlated, a larger patch can perform worse.

A threshold is therefore not a finish line. It means that, under a model and operating conditions, logical error can be suppressed by spending more resources. Useful computation then requires logical gates, state preparation, measurement, routing and decoding to remain fault tolerant together. Non-Clifford gates, needed for universal computation, are especially expensive in many architectures. Surface-code plans often rely on producing and distilling high-fidelity magic states, turning factories for one special resource into a large share of the machine.

The distinction between physical and logical qubits follows. A device may

contain hundreds or thousands of physical qubits yet support only a handful of logical qubits, or none that can run a long universal circuit. Overhead depends on physical error rates, code choice, connectivity, target logical error and algorithm length. There is no fixed conversion ratio. The longer the calculation and the lower the tolerated failure probability, the more protection it demands.

Error mitigation is different. It estimates or cancels some effects of noise by combining data from imperfect runs, stretching noise, discarding detected faults or using a model of the device. It can improve an observable without encoding a fully protected logical state. The sampling overhead may rise sharply as circuits deepen. Mitigation is useful experimental machinery, but it does not supply the scalable guarantee sought from fault tolerance.

The field crossed important experimental thresholds in the middle of the 2020s. Google's 2025 Willow paper reported a distance-7 surface-code memory using 101 physical qubits and a logical error rate that improved as code distance increased. The same system included real-time decoding at distance 5. In 2026, a further Willow experiment used reinforcement-learning control to compensate injected drift and reported an average surface-code logical error per cycle of 7.72 times 10^-4 in its tested regime. The important trend is not the record number by itself. It is that researchers are now improving logical behaviour by controlling the full feedback system, rather than treating each qubit as an isolated component.

Other platforms have reached different pieces of the stack. Neutral-atom experiments have combined repeated correction, logical operations, atom rearrangement and transversal gates in arrays of hundreds of atoms. Colour-code work on superconducting hardware has shown error suppression as code distance increased, together with logical Clifford operations, magic-state injection and lattice-surgery demonstrations. A 2026 silicon-donor experiment demonstrated a universal set of logical operations on a small error-detecting code. Trapped-ion researchers demonstrated a small fault-tolerant universal toolbox without mid-circuit measurement and used it to run an encoded Grover search. Each experiment answers a

different engineering question. None combines enough logical qubits, depth, throughput and universal fault-tolerant operations to run a large useful algorithm.

That variety matters because fault tolerance is not one code on one chip. It is a systems property. A memory that survives longer is useful but may not support a cheap T gate. A platform with excellent gates may pay in slow measurement. A code with convenient logical operations may demand more difficult stabiliser checks. A decoder that works offline may fail a real-time latency budget. A fault-tolerant architecture succeeds only when its weakest necessary component scales with the rest.

Nor is repeated measurement the only possible way to diagnose faults. Most leading architectures extract syndromes into ancillas and measure them, because a classical decoder can then infer corrections. Coherent schemes can instead keep syndrome information in auxiliary quantum systems during the computation and apply conditional quantum feedback before those auxiliaries are reset. The logical principle is the same: acquire information about the error channel without learning the amplitudes that encode the answer. This distinction is important because it separates the information-theoretic requirement from one hardware implementation.

The causal loop now closes. Qubits create computational possibilities because amplitudes, phases and entanglement let the processor manipulate joint quantum structure that can be expensive for a classical machine to track. Those same features forbid ordinary backup copying and make uncontrolled interactions dangerous. Error correction succeeds by adding carefully designed redundancy around the state while preserving what must remain unknown. Quantum computing can scale only when the machine learns enough about its faults to remove them without learning the answer it is trying to protect.

How It Actually Works

A quantum computer is easiest to understand by following one job from the question to the answer. The machine is a stack: mathematical problem, algorithm, compiler, logical operations, physical control, repeated measurements, error handling and classical interpretation. The quantum chip is the unusual component, not the whole computer.

Richard Feynman supplied the original pressure in 1981. Simulating a generic quantum system on a classical machine can require tracking a number of amplitudes that grows exponentially with the number of components. His proposal was to make the simulator obey the same rules as the system. David Deutsch then described a universal quantum computer in 1985. Shor's factoring algorithm in 1994 and Grover's search algorithm in 1996 turned the idea from a simulator into a general computational programme. The practical problem since then has been to make those ideal transformations survive real hardware.

First choose a problem that deserves a quantum computer

The first step is classical: decide what function, distribution or physical quantity is required. This matters because a quantum speedup can disappear before the circuit begins.

Some inputs are naturally compact. The integer in Shor's algorithm is given by a bit string. A quantum simulation can be specified by local interactions rather than by listing the full many-body wavefunction. Other proposals begin with a huge classical data set and assume it can be loaded into amplitudes quickly. Preparing an arbitrary n-qubit state can itself require exponentially many operations. If the entrance to the algorithm costs more than the classical solution, the impressive circuit in the middle is irrelevant.

The output has the same constraint. A circuit may prepare a state corresponding to a long vector, but reading every component defeats the point. Good quantum problems usually ask for something compact: sample

from this distribution, estimate this observable, recover this hidden period, find a marked item, determine this phase, compare these energies. The first resource estimate should therefore include preparation and extraction, not begin at the first colourful gate symbol.

Turn the mathematics into a reversible circuit

The algorithm is written first in ideal logical operations. A compiler then decomposes those operations into the gate set available on a chosen architecture, maps qubits onto physical locations and schedules interactions. Connectivity matters. A machine with nearest-neighbour coupling may need swaps or teleportation to bring distant logical qubits together. A platform with flexible connectivity may pay instead in slower gates or more complex control.

Quantum gates on a closed ideal system are unitary and reversible. Ordinary algorithms often erase temporary information without thinking about it. A quantum circuit cannot do that freely. It keeps enough information to reverse its intermediate work, copies out only information that can be copied consistently and then uncomputes temporary registers. This is why "uncomputation" appears so often in quantum algorithms. Cleaning up computational rubbish is part of restoring the interference pattern the algorithm wants.

On a fault-tolerant machine the compiler has another job. It must turn ideal logical gates into code operations: transversal gates, code deformation, lattice surgery, teleportation, state injection or magic-state consumption. Different decompositions can change the required number of logical qubits, non-Clifford gates and correction cycles by orders of magnitude.

Translate gates into physical control

The same logical gate looks different in different machines. A superconducting processor uses shaped microwave and flux pulses on circuits containing Josephson junctions. An ion trap uses laser or microwave interactions between internal states and shared motion. Neutral-atom devices rearrange atoms in optical tweezers and use Rydberg interactions. Photonic systems create, interfere and detect photons. Spin-qubit devices manipulate electron or nuclear states in semiconductor structures.

All of them require calibration. Frequencies drift. Laser intensities and optical alignments change. Readout classifiers age. Crosstalk depends on which operations run together. A physical circuit is therefore executed by a classical control system that continually estimates device parameters and chooses pulses, timing and feedback. The ideal circuit diagram hides most of the engineering.

Initialisation prepares a known state, commonly 0 for each qubit. Gates then create the superpositions, phase relationships and entanglement required by the algorithm. A simple Bell-state circuit begins with 00, applies a Hadamard gate to the first qubit and then a controlled-NOT. Repeated measurement produces mostly 00 and 11. That experiment is useful because it checks preparation, a single-qubit gate, an entangling gate and correlated readout. It is not useful computation by itself.

To calculate, the circuit needs a pattern whose final statistics reveal an external property. Shor's algorithm coherently evaluates modular exponentiation, creating periodic structure across registers. A quantum Fourier transform converts that structure into peaks from which a classical routine can infer the period. Grover's algorithm alternates a phase-marking oracle with a diffusion operation that rotates amplitude towards the marked subspace. Phase estimation writes an eigenphase into relative phases and then converts them into a bit string estimate. The details differ, but each successful algorithm has an explicit route from hidden quantum structure to a measurable statistic.

Run, measure, repeat and let the classical computer work

A processor rarely runs once. It executes many shots. Each shot prepares the state, applies gates and performs the required measurements. A classical computer aggregates the bit strings, estimates probabilities or observables, checks consistency and may choose the next circuit.

Hybrid algorithms make this loop obvious. A classical optimiser proposes parameters, a quantum processor evaluates an objective through repeated measurements, and the optimiser updates the parameters. Error-mitigation methods may deliberately run related circuits at several effective noise levels or with randomised compilations and combine the results statistically. Fault-tolerant protocols may stream syndrome information into a decoder whose output changes later logical operations.

This classical layer is not embarrassing scaffolding that disappears when quantum hardware matures. Even a large fault-tolerant quantum computer would be a specialised processor embedded in classical computing. The classical system supplies compilation, control, decoding, arithmetic, scheduling, verification and almost all ordinary data handling.

Build the logical machine, not the qubit collection

For long calculations the central resource is a logical qubit: quantum information encoded so that physical faults can be detected and, within limits, corrected. Surface codes are prominent because they use local checks on a two-dimensional layout and tolerate relatively high physical error rates. Colour codes trade more complex checks for useful logical operations. Other families exploit biased noise, erasures, bosonic modes or different connectivity.

A surface-code logical qubit is surrounded by ancillas that repeatedly extract parity information. The resulting syndrome history is sent to a decoder, which infers a likely pattern of faults. If the physical error rate is below the relevant threshold and the assumptions about noise are sufficiently accurate, increasing the code distance can reduce logical error. If the device

is above threshold, enlarging the code can make matters worse.

Threshold is therefore a scaling condition, not a declaration of victory. A complete machine must protect state preparation, memory, measurement, routing and a universal set of logical gates. Non-Clifford gates are especially costly in many architectures. Surface-code resource estimates often devote large fractions of the machine to factories that prepare and distil magic states. Faster logical computation can require more factories, so time and physical-qubit count trade against one another.

The decoder can be a bottleneck too. Syndrome data arrive continuously. If classical decoding falls behind, feed-forward stalls or errors accumulate faster than the control system can interpret them. Recent experiments have demonstrated real-time decoders and low-latency feedback, but a useful large machine must sustain that performance across far larger codes and longer computations.

Not every fault-tolerant design requires measurement at every stage. Coherent schemes can transfer syndrome information into auxiliaries and perform feedback through gates before resetting those auxiliaries. This does not abolish the thermodynamic and information-processing cost of removing entropy, and it does not make error correction optional. It shows that "measure the syndrome" is one important implementation pattern rather than the definition of fault tolerance.

Resource estimates are designs for machines that do not exist yet

An algorithm paper may count ideal logical gates. An engineering plan has to decide how those gates become reliable operations on a physical architecture. That conversion is a resource estimate, and it is where many spectacularly small-looking quantum algorithms become large machines.

Start with the logical circuit. How many logical qubits must remain alive at once? How many two-qubit gates, measurements and non-Clifford operations are required? How much parallelism is possible? What total probability of failure is acceptable? A trillion-operation calculation cannot

tolerate the same logical error per gate as a thousand-operation demonstration. The required reliability follows from the length and structure of the job.

Then choose a code and physical error model. The estimate assigns code distances, syndrome ancillas and routing space. If universal computation relies on magic states, it sizes factories that prepare noisy resource states and distil them until their errors are low enough. Those factories can dominate the physical-qubit count. Running more factories in parallel shortens the calculation while enlarging the machine. Slower production saves qubits while increasing wall-clock time and exposure to memory errors.

Physical gate speed enters next. A processor with nanosecond-scale operations can afford more correction cycles in a second than a machine whose entangling gates take tens or hundreds of microseconds, but faster gates are not automatically better if their error rates force larger codes. Connectivity changes routing. Measurement time changes feedback latency. Leakage can require dedicated removal operations. Atom loss may be easier to diagnose than an unknown Pauli error. The relevant resource is therefore a vector, not one number printed on a slide.

Classical decoding belongs inside the estimate. A surface-code machine generates a stream of syndrome information which must be interpreted quickly enough to support feed-forward and maintain the logical clock. Cryogenic wiring, laser control, microwave generation, memory bandwidth and data movement all set engineering limits that asymptotic gate counts ignore. A quantum computer designed for useful scale is partly a classical real-time system built around a protected quantum core.

Resource estimates are uncertain because the future hardware parameters are guesses. Correlated errors, manufacturing defects, drift, cross-module links and decoder performance may differ from the assumptions. That does not make the exercise decorative. It identifies which parameter controls the cost. A modest reduction in physical error can shrink code distance and save vast numbers of qubits. A better logical circuit can remove entire

magic-state factories. A faster decoder can change the achievable logical clock. Resource estimation turns "one day we will run Shor" into a list of engineering obligations that can be tested one by one.

This is also why there is no universal conversion from physical qubits to logical qubits. The ratio depends on the code, error rates, target reliability, connectivity and algorithm. The correct unit is the complete fault-tolerant computation, not the component count.

Compare the hardware bargains

Superconducting circuits offer fast gates, lithographic fabrication and a mature control ecosystem, but they require deep cryogenics and face frequency crowding, wiring, leakage and fabrication variation. Trapped ions provide identical atomic qubits, long coherence and high-fidelity control with flexible interactions, while gates are slower and scaling large optical and motional systems is difficult.

Neutral atoms can be rearranged into large arrays with useful geometry. Rydberg excitation provides strong temporary interactions, and atom loss can sometimes be detected as an erasure rather than an unknown error. The hard parts include loading, loss, gate fidelity, readout and increasingly elaborate optical systems. Photonic machines gain natural communication and room-temperature transmission, but photons interact weakly and are easily lost; computation therefore relies on interference, measurement, entangled resource states and feed-forward. Semiconductor spin qubits promise extreme density and possible compatibility with industrial fabrication, but uniform manufacture, coupling, readout and cryogenic control must scale together.

Bosonic encodings store a logical degree of freedom in many levels of an oscillator and can build some error bias or correction into one mode. Topological quantum computing goes further in ambition, aiming to encode information non-locally so that local disturbances have less effect. Microsoft has reported increasingly long parity lifetimes and parity-measurement devices in semiconductor-superconductor hybrids. Its 2025 peer-reviewed

parity-measurement paper described progress towards a topological qubit rather than a conclusive demonstration of one. The safe 2026 position is that the architecture remains promising and experimentally unsettled.

Quantum annealers form a different category. They evolve a system towards low-energy configurations of an optimisation problem. They can contain many quantum elements and support useful research, but raw qubit counts are not directly comparable with universal gate-based machines. Performance depends on embedding, noise, schedule, connectivity and the best specialised classical optimiser.

Decide whether an advantage claim survives contact with reality

A claim of quantum advantage needs a named task, a metric, a required accuracy and a best-known classical baseline. Random-circuit sampling asks whether a quantum processor can sample from a distribution that is prohibitively expensive to reproduce classically. Scientific simulation may ask for an observable in a regime where leading tensor-network or perturbative calculations no longer converge. Cryptographic applications ask whether a logical circuit can finish before accumulated error makes the answer useless.

The classical baseline moves. Google's 2019 random-circuit sampling experiment was a landmark demonstration of control, but classical simulation methods improved after publication and narrowed the original runtime gap. IBM's 2023 kicked-Ising work argued for useful computation before full fault tolerance but did not claim proven speedup; later classical methods reproduced the key observables efficiently. Neither sequence invalidated the experiments. Both showed that "beyond classical" is a comparative claim that must be re-tested.

By 10 August 2026 the frontier had moved again. Martiel and colleagues reported a 70-qubit, depth-70 error-detected sampling experiment with a device-dependent fidelity certificate. Leviatan and colleagues reported error-mitigated Floquet-dynamics measurements up to 74 qubits in regimes

where leading tensor-network calculations failed to converge, with several cross-checks. Barron and colleagues proposed a framework for validating observable estimates when exact classical verification is unavailable. These were preprints submitted in late July, not settled universal advantage claims. Their importance is that verification is becoming part of the computational problem.

Logical hardware advanced in parallel. Google's below-threshold surface-code memory used 101 physical qubits for a distance-7 code and reported 0.143 per cent logical error per cycle in the 2025 Nature paper. Later 2026 work on Willow used reinforcement-learning control and reported an average surface-code logical error per cycle of 7.72 times 10^-4 in its tested setting. Neutral-atom work combined key pieces of a universal fault-tolerant architecture in arrays up to 448 atoms. Superconducting colour-code experiments, silicon logical processors and trapped-ion fault-tolerant logical toolboxes broadened the set of demonstrated operations. These are important pieces of a machine. They are not yet the machine implied by application-scale resource estimates.

How we know

Quantum computing has an unusually good evidence stack for a young technology. The mathematical claims behind no-cloning, Shor, Grover, error-correction thresholds and complexity bounds are explicit theorems or algorithms. Hardware claims are tested through calibration data, benchmarking, tomography on small systems, logical error rates, syndrome statistics and repeated experiments. The strongest current claims are published in peer-reviewed papers or exposed as preprints with enough method for independent attack.

The hardest evidence problem appears exactly where success is most interesting. If a computation is claimed to be beyond reliable classical simulation, exact classical verification is unavailable by definition. Researchers then rely on solvable subcases, internal consistency, cross-platform agreement, statistical certificates, noise models and the failure of several leading classical methods. None is equivalent to knowing

the answer in advance. Frontier advantage claims should therefore be read as evidence packages with stated assumptions, not as permanent borders between quantum and classical computation.

What People Get Wrong

“A quantum computer tries every answer at once”

A superposition can contain amplitudes associated with many computational paths, and one gate can transform all of them coherently. That is the fact behind the slogan. The missing fact is that measurement returns one sample. If the machine merely evaluates a function on every input and stops, the operator learns one input-output pair and gains no mass catalogue of answers.

The useful work comes from later interference. The circuit must attach phases or correlations to a feature, erase irrelevant working information and combine paths so that desired outcomes gain probability. Shor exposes periodicity. Grover amplifies marked states. Neither reads every branch. Many imagined quantum algorithms fail because they create a large superposition and have no economical way to extract the wanted information.

The myth survives because the internal state can be exponentially large, while “parallel worlds” makes an immediate story. It matters because it leads people to expect speedup for every combinatorial problem. A quantum computer can process a superposition coherently. It cannot print the superposition. The output bottleneck is part of the machine, not an implementation detail. Counting branches also says nothing about circuit depth, oracle cost or whether a classical method exploits the same structure more cheaply.

“A qubit is both 0 and 1”

The phrase warns that a qubit is not restricted to the two classical basis states. It fails when taken as a literal storage claim. A qubit in `alpha|0> + beta|1>` has amplitudes and a relative phase. Measurement in the computational basis yields 0 or 1 with probabilities set by the amplitudes. It does not reveal two simultaneous classical values.

A classical random bit can also have a fifty-fifty outcome. What it lacks is coherent phase. The states `(|0> + |1>)/sqrt(2)` and `(|0> - |1>)/sqrt(2)` produce the same immediate 0 and 1 statistics, yet a Hadamard gate distinguishes them with certainty. That difference can drive interference later.

The correction matters because “both” encourages a hidden-bit picture in which the qubit carries extra readable facts. It does not. The Bloch sphere is a better model: a qubit is a direction in a two-dimensional complex state space, and different measurements ask different questions of that state. Superposition is structure in possible outcomes, not two ordinary records occupying one box.

“Entanglement sends messages faster than light”

Entangled measurements can show correlations that no local hidden-variable model can reproduce. The correlations can persist across great distance, and the choice of measurement on one side changes which joint pattern the two observers will later see. None of that gives either observer a controllable remote signal.

Each local outcome is random. One observer cannot choose whether her result is 0 or 1 and therefore cannot encode a message in it. The distant observer's local statistics remain the same whatever measurement she performs. Only after the records are compared through a classical channel does the correlation become visible. Quantum teleportation also requires classical communication before the receiver can reconstruct the state.

The myth is persuasive because “instant correlation” sounds like “instant

communication”, and popular accounts often omit the difference between a joint distribution and a usable local effect. The correction matters for both physics and computing. Entanglement supplies non-classical joint structure, but it does not remove communication latency, allow instantaneous distributed computation or violate relativistic signalling limits.

“The machine with the most qubits is winning”

A qubit count says how many nominal two-level systems a device controls. It says little by itself about how long a useful circuit can run. Gate fidelity, measurement error, coherence, leakage, connectivity, speed, crosstalk, calibration stability and compiler overhead can outweigh the count. A smaller processor with cleaner operations may execute a deeper reliable circuit than a larger one.

The comparison becomes harder across architectures. An annealer's qubits do a different job from universal gate-model qubits. A physical qubit is not a logical qubit. A logical qubit may consume many physical qubits, ancillas, decoder operations and repeated syndrome cycles. The ratio changes with error rate, code, algorithm length and required failure probability.

Raw counts became popular because they are visible, rising and easy to rank. Manufacturers also need simple milestones. The correction matters because it changes procurement, investment and scientific judgement. Ask for logical error per operation, achievable circuit volume, connectivity, gate set, shot rate and the resources needed for the named application. A crowded chip can be an engineering success without being a more capable computer. Architecture determines whether that quantity becomes usable capability.

“Quantum computers will make ordinary computers obsolete”

Quantum processors are poor replacements for the tasks at which classical machines are already excellent. Reading email, rendering a webpage, running a database, adding accounts and controlling a vehicle do not gain from preserving delicate phase relationships at cryogenic temperatures or inside a laser system. Classical bits are cheap, robust, copyable and easy to inspect.

Even quantum algorithms depend on classical computing. Classical software selects the problem, compiles the circuit, generates control signals, decodes errors, aggregates shots and checks results. Fault-tolerant machines would still sit inside data centres beside conventional processors, much as graphics processors accelerate selected numerical work without replacing central processors.

The myth comes from treating “faster on some problems” as a new general generation of computer. The correction matters because realistic value lies in division of labour. Quantum devices may accelerate parts of simulation, algebra, search or sampling when the complete resource case works. Most code will never run on them. A mature quantum computer would be a specialised co-processor attached to an overwhelmingly classical system.

“Shor’s algorithm has already broken encryption”

Shor's algorithm proves that a fault-tolerant quantum computer could factor integers and solve discrete logarithms using polynomial resources. That threatens RSA, finite-field Diffie-Hellman and elliptic-curve cryptography. Present machines cannot run the enormous, low-error logical circuits required for deployed key sizes. Small demonstrations factor numbers that classical computers solve instantly and often use compiled shortcuts specific to the answer.

The algorithm also does not destroy every cryptographic method. Symmetric encryption and hash functions face a different quantum effect, mainly the quadratic search improvement associated with Grover, which can

be countered by larger parameters. Post-quantum public-key schemes use mathematical problems for which no efficient quantum attacks are known. NIST finalised its first three such standards in 2024.

The myth is sustained by collapsing a proved future vulnerability into a current attack. The correction is not permission to delay migration. Stored encrypted data may be collected now and attacked later, and replacing cryptography across long-lived systems takes years. The honest statement is sharper: Shor has changed what prudent security engineering must do, while no cryptographically relevant Shor attack has yet been demonstrated.

“Better physical qubits will make error correction unnecessary”

Better materials, fabrication, shielding, control and calibration can reduce physical error rates. They cannot turn a working quantum processor into a perfectly isolated object. Gates require interaction. Measurement couples microscopic states to amplifiers. Qubits must be controlled, moved or connected. Large systems create more surfaces through which unwanted correlations, heating, drift and rare faults can enter.

Classical computers also experience physical noise, but strong signal margins and cheap copying make correction easy. Quantum states include phase, cannot be cloned and can be damaged without an obvious bit flip. Long computations therefore need an architecture in which error decreases as protection grows. That is the purpose of thresholds, logical qubits and fault-tolerant operations.

The myth persists because early hardware progress is often presented as a race towards “good enough” physical qubits. Good physical qubits are necessary, and they lower error-correction overhead. They are not a substitute for error correction at application scale. The correction matters because it changes the roadmap from one perfect device to a layered system: imperfect components, repeated syndrome extraction, fast decoding and logical operations whose residual failure can be made as small as the algorithm demands.

Use It

Ask where the interference comes from

When someone proposes a quantum speedup, ask which computational paths receive different phases and which later operation makes those paths interfere. A claim that stops at superposition has not explained an algorithm. The machine can prepare many alternatives, but measurement will return one. Something must bias that final distribution.

This question works at several levels. In Grover search, the oracle marks solutions by phase and the diffusion step amplifies them. In phase estimation, controlled evolution writes an eigenphase across a register and an inverse Fourier transform turns it into peaks. In a variational circuit, the answer may be less clean: a classical optimiser adjusts gates until a measured objective improves. If nobody can identify the mechanism connecting quantum structure to a better output, the proposal may be a circuit looking for a problem.

Separate state size from readable information

An n-qubit state can require 2^n amplitudes for an exact classical description. That fact explains why simulation can become hard. It does not mean the device stores 2^n accessible classical numbers. Measurement returns n bits per shot, and estimating many properties can require many preparations.

Use this distinction whenever a claim relies on “exponential information”. Ask what compact quantity is being extracted and how many shots, measurement settings and classical operations are needed. Quantum tomography exposes the trap: reconstructing a generic state grows exponentially. A useful algorithm avoids reconstruction. It estimates a chosen observable, samples a distribution or decides a property. Large internal description is an opportunity only when the output can be compressed before readout. The same test applies to amplitude encoding. A proposal may assume that a long classical vector appears as a quantum

state in negligible time. Ask who prepares it, whether a specialised memory is assumed and how errors in the loading process affect the claimed scaling. Input access is part of the algorithm.

Count the logical machine

Raw physical qubits are a poor unit for application claims. Start with the logical circuit: how many logical qubits, which gates, what depth and what total failure probability does the algorithm require? Then ask how the hardware realises each logical operation. Include code distance, syndrome ancillas, routing, magic-state production, decoder latency and the physical cycle time.

This discipline often reverses an impressive comparison. A device with fast physical gates may spend enormous space on correction. A platform with slower gates may compensate through high fidelity or connectivity. One logical non-Clifford gate can cost far more than a Clifford gate. The meaningful question is not how many qubits fit on a chip. It is how many dependable logical operations the complete system can deliver before the application fails. Then ask whether those operations arrive at a useful rate. A logical processor that succeeds once per day and one that succeeds once per second may use the same algorithm and support different industries. Throughput, queueing and reset time belong beside fidelity.

Demand a named classical baseline

“Beyond classical computers” is incomplete. Which classical algorithm, running on what hardware, at what accuracy and with what memory limit? Brute-force state-vector simulation is rarely the only competitor. Tensor networks exploit limited entanglement. Monte Carlo methods exploit sampling structure. Perturbative methods exploit weak effects. Problem-specific heuristics may beat general algorithms.

The baseline must also be allowed to improve. Quantum demonstrations have repeatedly stimulated better classical methods. That does not erase the experimental achievement; it changes the performance claim. A fair comparison should include data loading, preprocessing, verification and

output extraction on both sides. Quantum versus yesterday's naive classical code is marketing. The contest is between the best complete methods available for the same result.

Distinguish demonstration, advantage and utility

A demonstration shows that a device can perform a chosen operation or circuit. An advantage claim says the quantum method outperforms the best known classical alternative on a defined metric. Utility adds value: the result matters scientifically, operationally or economically enough to justify the complete cost. These categories overlap but do not imply one another.

A logical memory below threshold is a major demonstration, though it may run no application. Random-circuit sampling can support an advantage claim without solving a customer problem. A chemistry calculation can be scientifically useful without beating every classical method in wall-clock time. Keeping the labels separate prevents two opposite errors: dismissing foundational control because it lacks an immediate market, and treating every laboratory milestone as proof that a practical industry has arrived. Add a fourth label when needed: verification. A result can be hard to compute and hard to trust. Ask which internal checks, solvable limits, cross-device comparisons or statistical certificates would detect a plausible failure.

Treat roadmaps as engineering hypotheses

A serious roadmap links future capability to intermediate tests: physical error rates, code performance, modular connections, fabrication yield, decoder speed, logical gate quality and application resource estimates. It is more informative than a date attached to “useful quantum computing”. It remains a hypothesis about many coupled systems.

Read a roadmap by looking for dependencies. What must improve at the same time? Which milestone has been demonstrated, which is projected and which relies on an unbuilt component? Does the plan measure logical performance or substitute raw qubit count? Are application targets based on a published algorithm and resource estimate? Companies need schedules to organise investment. Readers need to preserve the difference between

an engineering target and an observed result.

The limits

Quantum computing does not remove computational difficulty. It changes the cost of selected problems. Many speedups are polynomial rather than exponential. Some require coherent access to data that may be expensive to provide. Some return samples or expectation values rather than complete solutions. Some improve asymptotic scaling at sizes where constants, correction overhead and classical alternatives still dominate.

The theoretical model also abstracts from a difficult stack. Fault-tolerance theorems show that scalable reliable computation is possible under assumptions about noise and control. They do not guarantee that a chosen platform will reach the needed overhead, yield and operating cost. Below-threshold memories and logical gates are evidence that the route is physical, not evidence that the destination is close.

Current advantage claims are narrow by design and can be hard to verify. A classically intractable calculation cannot be checked by straightforward classical recomputation. Certificates, cross-platform tests and solvable limits improve confidence, while new classical methods can still revise the comparison. There will be no single morning on which quantum computing changes from false to true. Capabilities will arrive by task, and many will remain specialised. Hardware can also improve while an application recedes, because the best classical method, required accuracy or economic value changes. Progress on the machine and progress towards a particular use are related measurements, not the same curve.

The one thing to keep

Keep the conversion problem.

A quantum computer can manipulate a state whose exact classical description grows explosively, but it cannot hand that description to you. The algorithm has to turn whatever global structure matters into a small classical signal: a factor, a phase, a sample, an expectation value, a marked item. If it

cannot explain that conversion, superposition and entanglement are decoration rather than computation.

Hardware faces the mirror image of the same problem. It must preserve the parts of the state that are computationally useful while continuously removing entropy, diagnosing faults and communicating with classical control. It cannot protect arbitrary unknown quantum information by ordinary copying. It therefore spends physical qubits, gates, time and decoding effort to create logical behaviour that looks far more reliable than any component underneath it.

This is why quantum computing resists both hype and dismissal. The state space is not imaginary, and interference, entanglement and fault-tolerant encoding are experimentally real. The output bottleneck, noise, verification problem and resource overhead are equally real. Progress is measured by making the conversion from coherent quantum structure to trustworthy classical evidence cheaper, deeper and more reliable.

When you hear a new claim, ask two questions. What quantum structure is being manipulated that the classical competitor finds expensive? And exactly how does the experiment turn that structure into a small answer we can read and trust?

Those questions leave little room for magic.

Terms

Amplitude

A complex number attached to a possible outcome. Amplitudes add and interfere before their squared magnitudes become measurement probabilities, which makes them the working currency of quantum algorithms.

Ancilla

An extra qubit used for temporary work, syndrome extraction, teleportation or gate construction. Ancillas help manipulate information without reading or damaging the logical state directly.

Bloch sphere

A geometric map of every pure state of one qubit. Gates appear as rotations, measurement bases as axes, and decoherence as loss of a sharply defined direction.

Circuit

An ordered sequence of state preparations, gates, measurements and classical controls. The circuit is the abstract program; compilation converts it into operations a particular device can perform.

Coherence time

A characteristic timescale over which a qubit preserves phase or population information. It matters only alongside gate speed, error rates and the structure of the intended circuit.

Computational basis

The reference measurement basis labelled `|0>` and `|1>`. Hardware readout usually distinguishes these states, while other measurements are implemented by rotating the qubit before readout.

Decoherence

Loss of usable quantum coherence through uncontrolled interaction with the environment. It turns intended phase relationships into inaccessible correlations and is a central source of computational error.

Entanglement

A joint quantum state that cannot be divided into independent states for its parts. It supports non-classical correlations, global encodings and error correction, without enabling faster-than-light messages.

Error correction

Encoding logical information across physical systems and repeatedly measuring syndromes to diagnose faults. The aim is to reduce logical error as code size increases below threshold.

Error mitigation

Techniques that estimate cleaner results from noisy circuits without creating fully protected logical qubits. Examples include noise extrapolation, probabilistic cancellation, symmetry checks and post-selection.

Fault tolerance

Designing encoded operations so that a limited number of physical faults does not spread into logical failure. It extends error correction from memory to a complete computation.

Fidelity

A measure of how closely a prepared state, gate or process matches its target. Different fidelity definitions test different failures, so one headline number cannot describe a processor fully.

Gate

A controlled transformation of one or more qubits. Ideal quantum gates are unitary before measurement, and a small universal set can approximate any finite quantum computation.

Grover's algorithm

A quantum search procedure that reduces an unstructured search through N candidates from linear query growth to roughly square-root query growth. Oracle construction and verification remain part of the cost.

Hadamard gate

A single-qubit gate that exchanges computational-basis certainty with equal superpositions. It is widely used to create interference and to convert relative phase into measurable outcome differences.

Interference

The reinforcement or cancellation of amplitudes assigned to alternative computational paths. A quantum algorithm becomes useful when interference concentrates probability on outcomes that reveal the desired property.

Logical qubit

An encoded qubit protected by an error-correcting code. It may require many physical qubits, repeated syndrome measurements, decoding and extra resources for reliable logical gates.

Magic state

A specially prepared non-stabilizer state consumed to implement important non-Clifford operations fault tolerantly. Producing high-fidelity magic states can dominate the cost of large algorithms.

Measurement

A physical interaction that creates a classical outcome with probabilities set by the quantum state. It usually removes the coherence needed for later interference in the measured degree of freedom.

NISQ

Noisy intermediate-scale quantum, John Preskill's term for processors too large for trivial treatment yet lacking full error correction. It describes a hardware era, not a guarantee of useful advantage.

No-cloning theorem

The result that no universal operation can make a perfect copy of an arbitrary unknown quantum state. Quantum redundancy must therefore use encoded correlations rather than independent replicas.

Phase

The angular part of a complex amplitude. Global phase has no observable effect, while relative phase can alter later interference and is therefore computationally significant.

Physical qubit

A controllable two-level quantum system in hardware, such as a superconducting circuit, ion, atom, photon or spin. Physical count alone does not measure useful capability.

Quantum advantage

A demonstrated performance gain over the best known classical method on a defined task and metric. It may be narrow, temporary and unrelated to commercial utility.

Quantum annealing

Computation by evolving a system towards low-energy configurations of an optimisation landscape. Annealers are specialised devices and should not be equated automatically with universal gate-based computers.

Quantum Fourier transform

The quantum analogue of the discrete Fourier transform applied to amplitudes. Its interference pattern is central to period finding and phase estimation, though measurement reveals only sampled information.

Quantum simulation

Using a controllable quantum system to reproduce the dynamics or properties of another. It was an early motivation for the field and remains a leading prospective application.

Qubit

The basic unit of quantum information, described by amplitudes for two reference outcomes and their relative phase. It is a state space, not a classical bit with a third value.

Shor’s algorithm

A polynomial-time quantum algorithm for integer factoring and discrete logarithms. Its security consequences depend on building a sufficiently large fault-tolerant processor, which current machines cannot yet provide.

Surface code

A two-dimensional topological error-correcting code using local parity checks. It offers a comparatively high threshold but can require large physical-qubit and magic-state overheads.

Go Deeper

1. Chris Bernhardt, Quantum Computing for Everyone

MIT Press, 2019. Start here if the book has made you curious but you do not want a physics degree inserted between you and the subject. Bernhardt builds qubits, gates, entanglement and algorithms from clear linear-algebra ideas, with enough mathematics to remove the magic without turning the book into a reference manual. It is especially good on circuit notation and the difference between quantum states and classical probabilities. The warning is the title: “everyone” still needs patience with symbols. Work through the small examples rather than reading them as prose.

2. Eleanor Rieffel and Wolfgang Polak, Quantum Computing: A Gentle Introduction

MIT Press, 2011. This is the structured next step from intuition to method. It covers the circuit model, quantum Fourier transform, search, simulation, error correction and complexity with more formal detail than Bernhardt, while remaining designed for readers entering from computer science or mathematics. Use it when you want to know why an algorithm works rather than retain only its public description. Some hardware discussion has aged because the field moved quickly, but the conceptual and algorithmic chapters remain useful. Keep pencil and paper beside it.

3. Michael A. Nielsen and Isaac L. Chuang, Quantum Computation and Quantum Information

Cambridge University Press, 10th anniversary edition, 2010. This is the standard large reference, often called Nielsen and Chuang as though the authors formed one object. It gives the mathematical framework, algorithms, information theory, noise, error correction and physical implementation at textbook depth. Do not begin at page one and expect a one-hour sequel. Use it to pursue a specific mechanism from this book, check a proof or learn the formal language of the field. Its age shows mainly in the hardware frontier, not in the foundations.

4. Peter W. Shor, “Algorithms for Quantum Computation: Discrete Logarithms and Factoring”

Proceedings of the 35th Annual Symposium on Foundations of Computer Science, IEEE, 1994, pages 124 to 134. Read the primary turning point. The paper is compact, technical and written before a functioning quantum computer could run anything close to its main result. Its value is seeing how the security consequence emerges from period finding rather than from a vague claim of quantum parallelism. The notation assumes comfort with number theory and quantum computation, so use a textbook explanation alongside it. Few papers have changed engineering plans so far in advance of the machine they require.

Notes and Sources

The Whole Thing in One Page and Why You Should Care

The book's central account follows the gate-model treatment in Nielsen and Chuang, Rieffel and Polak, and Bernhardt: a quantum computation manipulates complex amplitudes and relative phases, then extracts limited classical information through measurement. The rejection of “all answers at once” is not a denial that one operation can act coherently across a superposition. It is a correction about output and algorithm design. Interference must make a global property visible before measurement.

The statement about current capability is dated 10 August 2026. Google Quantum AI's 2025 surface-code paper reported below-threshold logical memories, including a distance-7 memory using 101 physical qubits; Sivak and colleagues' 2026 Willow work reported improved logical stability under reinforcement-learning control. Bluvstein and colleagues reported a programmable logical neutral-atom processor in 2024 and key elements of a universal fault-tolerant neutral-atom architecture in the 2026 volume of Nature. Lacroix and colleagues reported colour-code error suppression and logical operations in 2025. Zhang and colleagues reported universal logical operations in a silicon donor processor in 2026. Butt and colleagues demonstrated a small measurement-free fault-tolerant universal logical toolbox and encoded Grover implementation in 2026. These are substantial component and architecture demonstrations. None reports a broad, application-scale, general-purpose fault-tolerant machine.

The July 2026 advantage discussion uses three recent preprints by Martiel and colleagues, Leviatan and colleagues, and Barron and colleagues. Their results are described as task-specific claims under scrutiny because they were less than two weeks old at verification. The book does not infer broad commercial advantage from them.

The post-quantum standards are NIST FIPS 203, FIPS 204 and FIPS 205, approved on 13 August 2024. FIPS 203 specifies ML-KEM for key

establishment. FIPS 204 specifies ML-DSA and FIPS 205 specifies SLH-DSA for digital signatures. The migration is mentioned only to show that a proved quantum algorithm can change present engineering before the required machine exists. Detailed security consequences belong to Cryptography in a Hurry.

Core Idea 1: the qubit

The notation `|psi> = alpha|0> + beta|1>` and the normalisation condition are standard. A pure state of n qubits uses 2^n complex amplitudes in a generic state-vector description. The comparison at 300 qubits is an order-of-magnitude illustration: 2^300 is about 2 times 10^90, above the commonly quoted order of 10^80 particles in the observable universe. It does not claim that nature stores a classical table of those amplitudes or that every 300-qubit state is classically hard to describe.

The Bloch sphere represents pure single-qubit states modulo global phase. Mixed states lie inside the sphere. Physical qubits need not be literal spins; superconducting circuits, atomic levels, photon modes and semiconductor spins can all implement an abstract two-level system. The distinction between a coherent superposition and a classical probability mixture follows Nielsen and Chuang and Rieffel and Polak.

Core Idea 2: phase and interference

The plus and minus states have identical computational-basis probabilities and become distinct after a Hadamard gate. This is the simplest circuit demonstration that relative phase carries operational information. Global phase is unobservable; relative phase can change later probabilities. The two-path interferometer account is a standard amplitude-level description and does not require a claim about a photon following one classical route in secret.

The distinction between adding amplitudes and adding probabilities is central to Feynman's path-based explanations and to modern quantum-algorithm texts. Phase-flip and bit-flip errors are introduced here because both must be corrected in a general quantum code.

Core Idea 3: gates and circuits

Deutsch's 1985 paper supplied the universal quantum-computer model. The gate account follows Nielsen and Chuang. Ideal closed-system gates are unitary and therefore reversible. Classical irreversible functions can be embedded in reversible circuits by retaining input and working information. Bennett's work on logical reversibility supplies the classical foundation; uncomputation is standard in quantum algorithm design.

Grover's original 1996 paper gives the square-root query scaling. The description of phase marking and amplitude amplification uses the usual geometric interpretation. The text keeps oracle construction and checking inside the resource question because query complexity alone is not an end-to-end runtime claim.

The comments on alternative models draw on measurement-based computation, adiabatic computation and analogue simulation. Quantum annealing is kept separate from universal gate computation because device size and performance cannot be compared by raw qubit count across those models.

Core Idea 4: entanglement

The Bell state used in the text is maximally entangled and cannot be factored into independent pure states. Bell's 1964 theorem and later loophole-free tests establish that suitable quantum correlations cannot be reproduced by local hidden-variable assignments. The book does not treat correlation in one basis as sufficient evidence of entanglement.

No-signalling is retained explicitly. Entanglement changes joint statistics but leaves each party's controllable local statistics unable to carry a chosen faster-than-light message. Bennett and colleagues' 1993 teleportation protocol requires two classical bits in addition to shared entanglement.

The warning that entanglement is not a certificate of classical hardness is supported by the Gottesman-Knill result and by tensor-network methods. Clifford circuits can create substantial entanglement while remaining

efficiently simulable. Classical difficulty depends on circuit and state structure, required accuracy and the available classical representation.

Core Idea 5: measurement, output and copying

The Born rule connects squared amplitude magnitudes to outcome probabilities. Repeated shots estimate distributions or expectation values, with ordinary sampling uncertainty often shrinking as the inverse square root of the number of independent samples. The text does not claim that every quantum-estimation method obeys the same scaling; amplitude-estimation algorithms can improve query scaling under stronger coherent-access assumptions.

Wootters and Zurek's 1982 paper is the cited source for the no-cloning theorem. The theorem forbids a universal perfect copier for arbitrary unknown states. It does not prevent copying known basis states, recreating a state from a known preparation procedure or producing approximate and restricted copies.

The tomography statement concerns a generic unknown many-qubit state. Special state families can be learned with fewer measurements when structure is known. Mid-circuit measurement and feed-forward are standard in teleportation and fault-tolerant protocols. Butt and colleagues (2026) demonstrate that some fault-tolerant universal logical operations can instead keep syndrome handling coherent during execution, using auxiliary qubits and later reset; the book uses this to avoid treating mid-circuit measurement as definitional.

Core Idea 6: algorithm-specific speedup

Shor's 1994 paper gives polynomial-time quantum algorithms for factoring and discrete logarithms. The book separates the mathematical result from the physical overhead of a cryptographically relevant implementation. It does not provide a fixed physical-qubit estimate because estimates vary with architecture, physical error, code, gate speed, target failure probability and circuit improvements.

Grover's result is a quadratic oracle-query improvement for unstructured search. The square-root scaling does not remove input, oracle or verification costs. Brassard and colleagues developed amplitude-estimation methods with related quadratic query improvements for estimation tasks.

Feynman's 1982 proposal and Lloyd's 1996 universal-simulation result support quantum simulation as a native application. Phase estimation is treated as a recurring algorithmic primitive. Peruzzo and colleagues introduced the variational quantum eigensolver in 2014; Farhi, Goldstone and Gutmann introduced QAOA in the same period. The text's caution reflects later work on noise, trainability, sampling cost and competitive classical heuristics. “Hybrid” identifies where the work runs, not whether it wins.

The comments on linear systems and data access refer to the Harrow-Hassidim-Lloyd algorithm and to the fine print around state preparation, conditioning and output. Aaronson's 2015 commentary is a useful short warning against converting favourable subroutine scaling into an unsupported end-to-end claim.

No efficient general quantum algorithm is known for NP-complete problems. This is a statement about current algorithmic knowledge, not a proof that no such algorithm can exist.

Core Idea 7: error correction

Shor and Steane supplied early quantum error-correcting codes. The key construction is encoded redundancy: amplitudes are spread across entangled codewords, while ancillas measure parity checks that reveal error syndromes without revealing the logical amplitudes. General single-qubit errors can be represented through the Pauli error basis for correction purposes.

Dennis and colleagues and Fowler and colleagues are the main surface-code sources used. Code distance is the minimum weight of a logical operator or undetectable chain in the relevant setting. The statement that a distance d code corrects up to `(d - 1)/2` independent errors is an

ideal code-distance rule; real circuit-level performance also depends on measurement faults, leakage and correlations.

Threshold means that logical error can be reduced by increasing protection when physical operations and the noise model are sufficiently favourable. It does not mean the device is application-ready. Campbell, Terhal and Vuillot review routes to fault-tolerant universal computation. Bravyi and Kitaev are the source for magic-state distillation as a route to non-Clifford operations. Temme, Bravyi and Gambetta support the distinction between mitigation and correction.

The Google 2025 numerical claims use the corrected Nature article: a 101-physical-qubit distance-7 memory, a reported logical error of 0.143 per cent per error-correction cycle and a suppression factor of 2.14 when distance increased by two. The paper also reported real-time decoding for distance 5. The body rounds only where the exact precision would distract.

Operating sequence and history

Feynman's lecture was delivered in 1981 and published in 1982. Deutsch's universal model appeared in 1985, Shor's algorithms in 1994 and Grover's search paper in 1996. The chronology is used to show the sequence from physical motivation to universal model to specific complexity results, not to claim that no related quantum-computation ideas existed earlier.

The operating stack follows standard hardware and software practice: choose a problem and encoding; compile an ideal circuit to native gates and connectivity; calibrate the device; initialise; run gates; measure repeated shots; post-process classically; and, for hybrid methods, update parameters and repeat. Logical machines add encoding, syndrome extraction, decoding and fault-tolerant gate construction.

The Bell-state example is exact in the ideal circuit model. Real devices show preparation, gate and measurement errors, so experimental frequencies need not be precisely half 00 and half 11. The Shor, Grover and phase-estimation descriptions are conceptual operating summaries rather than executable circuits.

The resource-estimation discussion follows the fault-tolerance literature and published factoring estimates, including Gidney and Ekerå. Their well-known RSA-2048 estimate is not quoted numerically because it is architecture-specific and has already been revised by later circuit and code work. The general lesson is stable: logical gate composition, magic-state supply, physical error, code distance and cycle time jointly determine the machine.

The decoder discussion is supported directly by the Google 2025 experiment, which treated real-time decoding as an operational requirement and reported latency alongside code cycles. Decoder throughput is part of the classical control stack, not an external bookkeeping task.

Hardware approaches

The superconducting account draws on Kjaergaard and colleagues. The trapped-ion account draws on Bruzewicz and colleagues. Neutral-atom trade-offs draw on Morgado and Whitlock and the logical-processor experiments by Bluvstein and colleagues. The photonic account uses Slussarenko and Pryde. Semiconductor-spin claims use Burkard and colleagues. These sources describe platform strengths and scaling obstacles without supporting a claim that one platform has won.

Topological quantum computation is described from Nayak and colleagues' theoretical review. Microsoft's 2025 parity-measurement paper reported observations consistent with its proposed route. Nature news coverage in 2025 and 2026 documented continuing external scepticism about whether the evidence established topological qubits. The body therefore calls the approach promising in principle and the specific experimental claims disputed.

The annealing distinction follows Albash and Lidar. Adiabatic quantum computation can be universal under appropriate conditions, while commercial quantum annealers are specialised implementations whose performance must be compared with classical optimisation methods on the same embedded problem.

Benchmarks and the current frontier

Arute and colleagues' 2019 random-circuit sampling experiment is used as an example of a narrow advantage demonstration followed by strong classical response. The book does not repeat the original runtime comparison because later simulation improvements changed it.

Kim and colleagues' 2023 paper reported evidence for utility before fault tolerance on a 127-qubit processor and explicitly stopped short of claiming established quantum speedup. Begušić, Gray and Chan later gave fast converged classical simulations for the relevant kicked-Ising observables. Other classical methods also addressed the experiment. This sequence supports the claim that classical baselines move.

The 2026 sampling preprint by Martiel and colleagues reported a 70-qubit, depth-70 circuit using 97 physical qubits, syndrome post-selection and a device-dependent fidelity lower bound. Leviatan and colleagues reported error-mitigated Floquet-dynamics measurements up to 74 qubits with tensor-network, Pauli-path and cross-platform checks. Barron and colleagues proposed a framework for observable estimation without exact classical verification. The text labels these as recent preprints rather than settled field-wide advantage.

What People Get Wrong and Use It

The seven misconceptions rest on distinctions established earlier: superposition versus readable output; coherent state versus classical mixture; correlation versus signalling; physical count versus logical capability; accelerator versus replacement; algorithmic threat versus current cryptographic attack; and physical improvement versus scalable fault tolerance.

The six lenses are resource-accounting tools rather than predictions. Input preparation and output extraction can erase a subroutine advantage. Logical operations, throughput and failure budgets matter more than one chip number. Classical comparison must name an algorithm and accuracy. Demonstration, advantage, verification and utility answer separate

questions. Roadmaps are assessed as dependency structures and targets, not recorded as achievements.

Bibliography

Primary papers, experiments and standards

Aaronson, Scott, and Daniel Gottesman. “Improved Simulation of Stabilizer Circuits.” Physical Review A 70 (2004): 052328.

Arute, Frank, et al. “Quantum Supremacy Using a Programmable Superconducting Processor.” Nature 574 (2019): 505-510.

Barron, Samantha V., et al. “Observable Estimation in the Absence of Classical Verification.” arXiv:2607.25998, 2026.

Bell, John S. “On the Einstein Podolsky Rosen Paradox.” Physics Physique Fizika 1 (1964): 195-200.

Bennett, Charles H. “Logical Reversibility of Computation.” IBM Journal of Research and Development 17 (1973): 525-532.

Bennett, Charles H., et al. “Teleporting an Unknown Quantum State via Dual Classical and Einstein-Podolsky-Rosen Channels.” Physical Review Letters 70 (1993): 1895-1899.

Bluvstein, Dolev, et al. “A Fault-Tolerant Neutral-Atom Architecture for Universal Quantum Computation.” Nature 649 (2026): 39-46.

Bluvstein, Dolev, et al. “Logical Quantum Processor Based on Reconfigurable Atom Arrays.” Nature 626 (2024): 58-65.

Brassard, Gilles, Peter Høyer, Michele Mosca, and Alain Tapp. “Quantum Amplitude Amplification and Estimation.” Contemporary Mathematics 305 (2002): 53-74.

Bravyi, Sergey, and Alexei Kitaev. “Universal Quantum Computation with Ideal Clifford Gates and Noisy Ancillas.” Physical Review A 71 (2005): 022316.

Deutsch, David. “Quantum Theory, the Church-Turing Principle and the Universal Quantum Computer.” Proceedings of the Royal Society A 400 (1985): 97-117.

Farhi, Edward, Jeffrey Goldstone, and Sam Gutmann. “A Quantum Approximate Optimization Algorithm.” arXiv:1411.4028, 2014.

Feynman, Richard P. “Simulating Physics with Computers.” International Journal of Theoretical Physics 21 (1982): 467-488.

Google Quantum AI and Collaborators. “Quantum Error Correction below the Surface Code Threshold.” Nature 638 (2025): 920-926. Corrected 2026.

Grover, Lov K. “A Fast Quantum Mechanical Algorithm for Database Search.” In Proceedings of the 28th Annual ACM Symposium on Theory of Computing, 212-219. New York: ACM, 1996.

Harrow, Aram W., Avinatan Hassidim, and Seth Lloyd. “Quantum Algorithm for Linear Systems of Equations.” Physical Review Letters 103 (2009): 150502.

Hensen, Bas, et al. “Loophole-Free Bell Inequality Violation Using Electron Spins Separated by 1.3 Kilometres.” Nature 526 (2015): 682-686.

Kim, Youngseok, et al. “Evidence for the Utility of Quantum Computing before Fault Tolerance.” Nature 618 (2023): 500-505.

Lacroix, N., et al. “Scaling and Logic in the Colour Code on a Superconducting Quantum Processor.” Nature 645 (2025): 614-619.

Leviatan, Eyal, et al. “Resolving Structure in Prethermal Floquet Dynamics with Precision Quantum Computation.” arXiv:2607.24937, 2026.

Lloyd, Seth. “Universal Quantum Simulators.” Science 273 (1996): 1073-1078.

Martiel, Simon, et al. “Sampling Hard Circuits with Verifiably High Fidelity.” arXiv:2607.25941, 2026.

Microsoft Azure Quantum, Morteza Aghaee, Alejandro Alcaraz Ramirez, et

al. “Interferometric Single-Shot Parity Measurement in InAs-Al Hybrid Devices.” Nature 638 (2025): 651-655.

National Institute of Standards and Technology. FIPS 203: Module-Lattice-Based Key-Encapsulation Mechanism Standard. Gaithersburg, MD: NIST, 2024.

National Institute of Standards and Technology. FIPS 204: Module-Lattice-Based Digital Signature Standard. Gaithersburg, MD: NIST, 2024.

National Institute of Standards and Technology. FIPS 205: Stateless Hash-Based Digital Signature Standard. Gaithersburg, MD: NIST, 2024.

Peruzzo, Alberto, et al. “A Variational Eigenvalue Solver on a Photonic Quantum Processor.” Nature Communications 5 (2014): 4213.

Raussendorf, Robert, and Hans J. Briegel. “A One-Way Quantum Computer.” Physical Review Letters 86 (2001): 5188-5191.

Shor, Peter W. “Algorithms for Quantum Computation: Discrete Logarithms and Factoring.” In Proceedings of the 35th Annual Symposium on Foundations of Computer Science, 124-134. Los Alamitos, CA: IEEE Computer Society Press, 1994.

Shor, Peter W. “Scheme for Reducing Decoherence in Quantum Computer Memory.” Physical Review A 52 (1995): R2493-R2496.

Steane, Andrew M. “Error Correcting Quantum Code.” Physical Review Letters 77 (1996): 793-797.

Temme, Kristan, Sergey Bravyi, and Jay M. Gambetta. “Error Mitigation for Short-Depth Quantum Circuits.” Physical Review Letters 119 (2017): 180509.

Wootters, William K., and Wojciech H. Zurek. “A Single Quantum Cannot Be Cloned.” Nature 299 (1982): 802-803.

Butt, Fabian, et al. “Demonstration of Measurement-Free Universal Logical

Quantum Computation.” Nature Communications 17 (2026): 995.

Sivak, Volodymyr, et al. “Reinforcement Learning Control of Quantum Error Correction.” Nature 655 (2026): 879-884.

Zhang, C., et al. “Universal Logical Operations in a Silicon Quantum Processor.” Nature Nanotechnology 21 (2026): 635-641.

Modern works and reviews

Aaronson, Scott. “Read the Fine Print.” Nature Physics 11 (2015): 291-293.

Albash, Tameem, and Daniel A. Lidar. “Adiabatic Quantum Computation.” Reviews of Modern Physics 90 (2018): 015002.

Begušić, Tomislav, Johnnie Gray, and Garnet Kin-Lic Chan. “Fast and Converged Classical Simulations of Evidence for the Utility of Quantum Computing before Fault Tolerance.” Science Advances 10 (2024): eadk4321.

Bernhardt, Chris. Quantum Computing for Everyone. Cambridge, MA: MIT Press, 2019.

Bruzewicz, Colin D., John Chiaverini, Robert McConnell, and Jeremy M. Sage. “Trapped-Ion Quantum Computing: Progress and Challenges.” Applied Physics Reviews 6 (2019): 021314.

Burkard, Guido, Thaddeus D. Ladd, Andrew Pan, Jason M. Nichol, and Jason R. Petta. “Semiconductor Spin Qubits.” Reviews of Modern Physics 95 (2023): 025003.

Campbell, Earl T., Barbara M. Terhal, and Christophe Vuillot. “Roads towards Fault-Tolerant Universal Quantum Computation.” Nature 549 (2017): 172-179.

Dennis, Eric, Alexei Kitaev, Andrew Landahl, and John Preskill. “Topological Quantum Memory.” Journal of Mathematical Physics 43 (2002): 4452-4505.

Fowler, Austin G., Matteo Mariantoni, John M. Martinis, and Andrew N. Cleland. “Surface Codes: Towards Practical Large-Scale Quantum

Computation.” Physical Review A 86 (2012): 032324.

Gidney, Craig, and Martin Ekerå. “How to Factor 2048 Bit RSA Integers in 8 Hours Using 20 Million Noisy Qubits.” Quantum 5 (2021): 433.

Kjaergaard, Morten, et al. “Superconducting Qubits: Current State of Play.” Annual Review of Condensed Matter Physics 11 (2020): 369-395.

Morgado, M., and S. Whitlock. “Quantum Simulation and Computing with Rydberg-Interacting Qubits.” AVS Quantum Science 3 (2021): 023501.

Nayak, Chetan, Steven H. Simon, Ady Stern, Michael Freedman, and Sankar Das Sarma. “Non-Abelian Anyons and Topological Quantum Computation.” Reviews of Modern Physics 80 (2008): 1083-1159.

Nielsen, Michael A., and Isaac L. Chuang. Quantum Computation and Quantum Information. 10th anniversary ed. Cambridge: Cambridge University Press, 2010.

Orús, Román. “A Practical Introduction to Tensor Networks: Matrix Product States and Projected Entangled Pair States.” Annals of Physics 349 (2014): 117-158.

Preskill, John. “Quantum Computing in the NISQ Era and Beyond.” Quantum 2 (2018): 79.

Rieffel, Eleanor G., and Wolfgang H. Polak. Quantum Computing: A Gentle Introduction. Cambridge, MA: MIT Press, 2011.

Slussarenko, Sergei, and Geoff J. Pryde. “Photonic Quantum Information Processing: A Concise Review.” Applied Physics Reviews 6 (2019): 041303.