The state of Quantum Computing in 2025

Adam Prior.[1]

1. *UrbanFox, Dublin, Ireland*

Abstract
This paper provides an accessible overview of the current state of quantum computing technologies as of September 2025. Aimed at readers with a technical background but no prior experience in quantum computing, it reviews foundational concepts, historical advancements, and leading hardware and software platforms. The paper also discusses practical challenges, potential applications, and the outlook for future developments in the field.

Keywords: Quantum Computing; Qubits; Decoherence; robotics

1. Introduction

Define Quantum Advantage here. Define Some other terms here.

todo: add to historical section that Shor's algorithm is somewhat of a benchmark for quantum computing

2. Historical Advancements in Quantum Computing

The first mentions of quantum computation can be traced as far back as the early as 1980s. In 1980, mathematician Yuri Manin discussed the concept of a quantum computer in his book *Computable and Non-Computable*[1]. Richard Feynman, in 1982, published a paper 'Simulating physics with computers' introducing the idea of simulating quantum systems using quantum computers[2], highlighting the limitations of classical computers at simulating the exponentially growing state space of quantum systems and how quantum systems can be more efficiently simulated.

Given these initial few years of defining work, the field began to gain traction, followed by another *10 years* of foundational theoretical developments. In 1985, David Deutsch further developed the quantum computing theory by rigorously defining the Universal Quantum Turing Machine (the quantum analog of a classical Turing machine) capable of performing any computation that a classical computer can, but more efficiently for certain problems[3]. Further theoretical advancements continued with contributions from, reversible quantum computation by Paul Benioff between 1985-1987[**BBenioff1987**], the first proposal of a physically realizable quantum computer using atoms and photons in 1998[**Yamamoto1998**], and attempts to define quantum complexity classes in 1989[**Bernstein1993**] which would eventually lead to the definition of Bounded-Error Quantum Polynomial Time (BQP) problems, an important class of problems in complexity theory that can be efficiently solved by quantum computers in polynomial time.

It was not until 1992 that the first quantum algorithm was proposed by Deutsch and Jozsa[4], demonstrating that quantum computers could solve certain problems more efficiently than classical computers (quantum advantage). While this algorithm has very limited practical applications, it paved the way for more impactful algorithms. The most famous of these is Shor's algorithm, proposed by Peter Shor in 1994[5]. Shor's algorithm is an efficient quantum algorithm for integer factorization in polynomial time, versus the best-known classical algorithms which run in sub-exponential time. This algoirithm in some sense 'proved' the potential of quantum computing to solve practical problems that are intractable for classical computers, particularly in the context of cryptography, as many encryption schemes rely on the difficulty of factoring large integers. Another significant algorithm is Grover's algorithm, proposed by Lov Grover two years later[6], providing quadratic speedup for unstructured search problems.

While these algorithms demonstrate the theoretical potential of quantum computers, they would be useless without the ability to assemble a physical quantum computer. Around the same time as these algorithmic developments, experimental researchers were developing the first real qubits and quantum gates. The first experimental demonstration of a quantum logic gate was realised in 1995 by a team using an electromagnetic trap to confine ions in place to form a two-qubit system. The energy levels of the ions are used to represent the qubit states ($|0\rangle$ and $|1\rangle$), while laser pulses are used to manipulate these states. The team successfully implemented a controlled-NOT (CNOT) gate, which is a fundamental quantum gate universal quantum gate used in many quantum algorithms[7]. One might define this point in time as the take-off point in the history of quantum computing as it was demonstrated, roughly in parallel, to be both theoretically computationally advantageous (for certain classes of problems) and experimentally realisable and a large acceleration in research and development followed. Over the next decade experiments would demonstrate more complex gates, and increased amount of qubits (8 qubit registers by December 2005)[8], and even new quantum systems that can implement qubits such as superconducting states, nuclear spins, quantum dots and even photon polarization. Theoretical advancements also continued, with the development of quantum error correction codes and various proofs of principle for quantum algorithms that further highlighted feasability and limitations.

Up to this point, it is well known that quantum computers suffer from an extreme scaling problems. Qubits are extremely sensitive to their environment and suffer badly from quantum decoherence which limits the rate at which additional qubits can be added to a system before the entire system becomes unstable highlighting why it took so long to scale beyond only a few qubits. Table 1 summarizes some of the key leaps in qubiit counts over the years.

3. Fundamentals of Quantum Computing

In order to properly understand the current state of quantum computing, it is important to first understand some of the fundamental concepts. What exactly is a qubit? What is superposition and entanglement? What is a quantum gate?

3.1. Qubits and their construction

For the average reader without a background in quantum mechanics, but with at least some technical background in computing, a qubit can be thought of as the quantum analog of a classical bit. When we use the term ‘classical’, we specifically mean non-quantum, as in the type of computing that underpins all modern computers today. A classical bit represents the fundamental unit of information in classical computing and is binary in nature, meaning that a bit exists in either one of two states, usually represented as 0 or 1. In classical physics, it is straightforward to encode a bit’s state as either 0 or 1. Usually this is done using voltage levels, where a high voltage might represent a 1 and a low voltage a 0. An important point to state at this point is that a classical bit can only exist in one of these two states at any given time. In classical physics there is no such thing as being both high and low voltage at the same time. It is either one or the other.

Similarly a qubit is also a binary unit of information, but it operates according to the principles of quantum mechanics. There are many quantum mechanical systems that can be used to physically realise a qubit, and generally, any two-level quantum system can technically be used as a qubit. Consider, for example, the spin angular momentum of an electron. This ‘spin’ can take strictly two values: ‘up’ or ‘down’, which can be used to represent the 0 and 1 states of a qubit. Other physical systems that can be used to realise qubits include the polarization states of photons, energy levels of atoms or ions, and superconducting circuits. We are not limited to just electronic spins but a rich variety of physical systems can be used to implement qubits, each with its own advantages and challenges. Such systems include but are not limited to:

- Trapped Ions electronic states
- Superconducting states (charge, flux, or phase)
- Photonic Qubits (polarization states, number of photons, etc)
- Quantum Dots (electron localization states, spin)
- Nuclear Spins (nuclear magnetic resonance spin)

When we consider that the underlying quantum system is being used as just a different way to encode a binary state, it might be tempting to think of a qubit as just a more complex representation of a classical bit. However, the actual power of of quibits lies in their ability to exist in a state called a *quantum superposition*, which is a fundamental principle of quantum mechanics.

3.2. Quantum States, Superposition and Entanglement

In the realm of quantum mechanics we represent the state of a quantum system using a special notation called Bra-ket notation. In this notation, the up state is represented as $|0\rangle$ and the down state as $|1\rangle$. As mentioned previously, however, unlike a classical bit, a qubit can exist in a combination (superposition) of both states simultaneously. For those who are mathematically inclined, this means that a qubit can be in a state that is a linear combination of $|0\rangle$ and $|1\rangle$, represented mathematically as:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{1}$$

where ψ just means ‘the state of the qubit’. The coefficients α and β are complex numbers that represent the probability amplitudes of the qubit being in the $|0\rangle$ and $|1\rangle$ states, respectively. This expression for the superposition, given that the complex coefficients are continuous values, show us that the qubit has an uncountably infinite amount of states it can be in. The probabilities of measuring the qubit to be in either state are given by the squares of the magnitudes of these coefficients. In some sense they represent the ‘weight’ of each state in the superposition, the likelihood that measuring the qubit will give that particular state, as in quantum mechanics, the act of measurement causes the

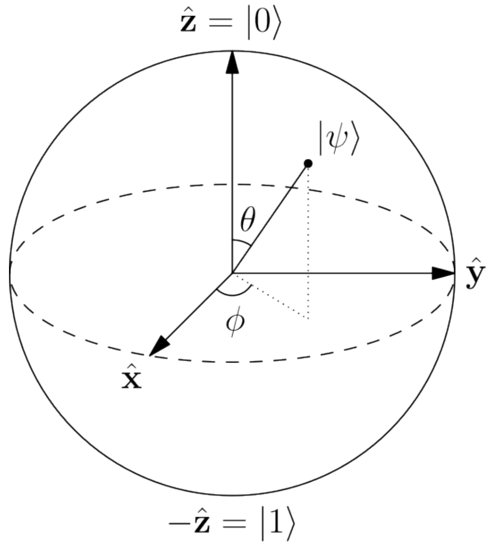

Figure 1: The Bloch Sphere representation of a qubit. Any point on the surface of the sphere represents a valid qubit state, with $|0\rangle$ and $|1\rangle$ at the poles. The angles θ and ϕ determine the specific superposition state.

qubit to 'collapse' into one of the base states (either $|0\rangle$ or $|1\rangle$) with probabilities determined by these coefficients. It is this superposition that enables quantum computers to perform certain computations more efficiently than classical computers as these complex coefficients can interfere constructively or destructively during quantum computations, allowing quantum algorithms to amplify the probabilities of correct answers while suppressing incorrect ones.

The nature of superposition also leads to the concept of *entanglement*, another fundamental property of quantum mechanics. When two qubits become entangled, the state of one qubit becomes directly correlated to the state of the other, regardless of the distance between them. This means that measuring one qubit instantly determines the state of the other qubit. Entanglement is a key resource in many quantum algorithms and protocols, enabling phenomena such as quantum teleportation and superdense coding. It is also a crucial component in achieving quantum advantage, as it allows for correlations that are not possible in classical systems as there is no classical analog to entanglement, it is a purely quantum mechanical phenomenon.

As the coefficients α and β are complex numbers, they have both a magnitue and a phase, and it is this phase that allows for interference effects to occur due to relative phase differences between qubit states. Maintining stable relative phases is critical for the proper functioning of quantum algorithms, and is measured by a property called coherence time, which is the time over which a qubit can maintain its quantum state before loss of information to the environment (decoherence) occurs. The global coherence of a multi-qubit system is particularly important, as quantum algorithms often rely on the interference of amplitudes across multiple qubits to achieve their computational advantage. Two important metrics used to quantify the coherence of qubits are τ_1 and τ_2 times. τ_1, or relaxation time, measures how long a qubit can stay in an excited state before it decays to its ground state due to energy loss to the environment. τ_2, or dephasing time, measures how long a qubit can maintain its phase coherence before interactions with the environment cause it to lose its quantum information. Both τ_1 and τ_2 times are critical for the performance of quantum computers, as they determine how long quantum information can be reliably stored and manipulated.

3.3. Quantum Gates

It's all well and good to be able to create qubits and understand their properties, in order to perform computations we need a way to manipulate these qubits and perform quantum gate operations on them. Quantum gates are the quantum analogs of classical logic gates. While classical logic gates could be interpreted as performing operations that flip or combine bits in their restrictied classical space, quantum gates are represented mathematically by matrices operating on the state (vectors) of the qubits, performing flips, rotations, entanglements and other transformations in the much larger quantum state space. This makes more sense when considering the geometric representation of a qubit on the Bloch sphere (see Figure. 1), where quantum gates can be visualized as rotations of the qubit state vector on this sphere.

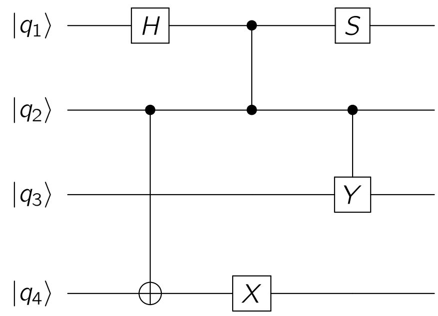

Figure 2: Example of a simple 4-qubit quantum circuit. Each horizontal line represents a qubit, and the symbols show basic operations that can be performed on them such as the Hadamard gate (H), controlled-NOT gate (CNOT), phase gate (S), Pauli-X gate (X), and Pauli-Y gate (Y).

In practice, the experimental implementation of quantum gates depends on the physical system used to realise the qubits. For example, in superconducting qubits, microwave pulses are used to manipulate the energy levels of the qubits, while in trapped ion systems, laser pulses are used to control the internal states of the ions. Common single-qubit gates include the Pauli-X, Y, and Z gates (which correspond to bit-flip and phase-flip operations), as well as the Hadamard gate (which creates superposition). Multi-qubit gates, such as the CNOT (Controlled-NOT) gate, are essential for creating entanglement between qubits and enabling more complex quantum operations. Given that gate operations are actual physical operations, they take a finite duration of time. The time taken to perform a gate operation, known as the gate time τ_g, is important in the context of quantum algorithms, as it must be much shorter than the coherence times (τ_1 and τ_2)(see practical challenges section).

3.4. Decoherence, Errors, and Correction

In an actual quantum computer, we are far from the ideal mathematical situation of qubits being perfectly coherent, noise free, and decoupled from the environment. In practice, the quantum states required are incredibly delicate, generally requiring the system to be supercooled to near absolute zero temperatures, and isolated from all external influences. Even still, it is impossible to remove all sources of noise, and as such, the system is subject to decoherence. Decoherence is the proceess by which a quantum system loses its quantum properties due to interactions with its environment. This can occur through various mechanisms, such as thermal fluctuations, electromagnetic interference, or imperfections in the qubit fabrication process. Decoherence leads to the loss of superposition and entanglement, effectively collapsing the qubit states into classical states and destroying the quantum information they carry. This is not a limitation of hardware alone, but a fundamental property of quantum mechanics itself. As stated earlier, the coherence times τ_1 and τ_2 are critical metrics that quantify how long a qubit can maintain its quantum state before decoherence occurs. As we add more qubits to the system, the global coherence of the multi-qubit becomes exponentially more fragile and susceptible to decoherence, making it increasingly challenging to maintain the quantum information across the entire system. Decoherence ultimately results in the occurrence of errors during quantum computations as the global state collapses unpredictably.

Decoherence is not the only source of error in quantum computations. The quantum gate operations themselves are also subject to imperfect operations and thus have some probability of failure. For example, attempting to perform an X gate on a qubit in the $|0\rangle$ state might not always successfully flip it to the $|1\rangle$ state due to imperfections in the control pulses or interactions with the environment during the gate operation. This is known as gate error. Gate fidelity is a measure of how accurately a quantum gate performs the intended operation, and it is typically quantified using metrics such as the average gate fidelity. As a rule of thumb, gate fidelities above 99% are generally considered good, but for practical quantum computing, even higher fidelities (e.g., 99.9% or better) are often required[**99.9percent-required-fidelity**].

To mitigate the effects of decoherence and gate errors, quantum error correction techniques are employed. Error correction involves encoding the quantum information across multiple physical qubits to create logical qubits that are more robust against errors. This is typically done using quantum error-correcting codes, such as the surface code or the Steane code, which can detect and correct certain types of errors without directly measuring the quantum information itself (which would collapse the state). Implementing error correction requires additional qubits and gate operations, which adds complexity to the quantum computer. The overhead for error correction can be significant, often requiring an order of magnitude more physical qubits to create a single logical qubit. However, this is essential for achieving fault-tolerant quantum computing, where computations can be performed reliably even in the presence of noise and errors.

4. Practical Challenges in Quantum Computing

4.11. Decoherence and Noise

As stated in the previous section, the fragility of quantum states plays a huge role in the accuracy of quantum computations. Qubits interact unavoidably with their environments, leading to decoherence that destroys superpositions and entanglement, which are essential for quantum advantage and reducing the error rates of quantum gates. Minimizing decoherence and noise is one of the biggest challenges in building practical quantum computers. The scaling effect of decoherence and noise is particularly problematic as the number of qubits increases, making it exponentially more difficult to maintain coherence across the entire system as we enter the realm of hundreds to thousands of qubits.

4.2. Gate Fidelity and Speed

Given the coherence windows τ_1 and τ_2, these set a hard limit on how many gate operations can be performed before the quantum information is lost. As a real example in trapped ion systems, which can have exceptionally long coherence times of the order of seconds or longer, single-qubit gate times can be on the order of microseconds, while two-qubit gates can take tens to hundreds of microseconds. For a 50 second coherence time and $500\mu s$ gate time, this allows for only 100,000 gate operations before decoherence occurs and destroys the computation. When considering a practical algorithm like Shors algorithm for integer factorization, this number seems far more than enough when considering the trvial example of factoring the number 15 (4-bits) into its factors of 3 and 5. But entering the regime of RSA encryption where we are dealing with 2048-bit numbers to be factorized, we are looking at the order of anywhere from billions to trillions of gate operations. Thus, faster gate operations are desirable to maximize the number of operations that can be performed within the coherence time. However, increasing gate speed often comes at the cost of reduced gate fidelity, as faster gates can be more susceptible to control errors and noise. Thus, there a balance must be struck between gate speed and fidelity that must be carefully managed to optimize the performance of quantum computers.

Gate fidelity itself, an inverse measure of error rate of a gate operation, is a critical factor as even tiny errors in gate operations can accumulate over the course of a quantum computation, leading to incorrect results. Achieving high gate fidelities (e.g., above 99.9%)[**99.9percent-required-fidelity**] is essential for practical quantum computing, but this remains a significant challenge due to various sources of noise and imperfections in the control systems used to manipulate qubits. However we are already approaching and surpassing this threshold with fidelities exceeding 99.9% in some platforms such as trapped ions, and germanium quantum dots[10, 11], and major advancements in 2025 in superconducting qubits have also pushed single gate fidelities reliably beyond 99.997%[12].

4.3. Error Correction, Fault Tolerance and the Logical-Physical Qubit Gap

Given the probabilistic guarantee of gate faults and decoherence faults, as absolute requirement of quantum computing is to design the computations to be fault tolerant. Acheiving fault tolerance in quantum computers opens its own entire area of research known as error correction with its own domain of difficulties. The goal of error correction is to accept that faults are guaranteed to occur, and aims to detect them, and in turn correct them. In general, this is achieved through the concept of a *logical qubit*.

While actual physical qubits are capable of computation, the probability of error is too high to achieve correct results for realistic problems. A logical qubit is an an collection of qubits that are used for redundancy in the case of errors. We might then say, that one *useful* quibit consists of multiple physical qubits. It is this logical qubit that quantum computation is done with. The logical qubit becomes a fault tolerant qubit via the implementation of error correction codes which use the ensemle to correct errors that occur in the logical qubit.

Many different error correction schemes exist, but leading schemes like surface codes or color codes require many physical qubits per logical qubit, leading to a huge overhead in the number of qubits required for practical quantum computing. In 2023, researchers at google showed experimentally that error rates could be reduced by using increased physical qubit counts[13]. Specifically they showed that increasing the physical qubit count from 17 to 49 per logical qubit, the error rate decreased from 3.0% to 2.9% While this important result shows the potential for improving error rates through increased qubit counts, it also highlights the scale at which we might need to reach to achieve robust error correction. Modern error correction scheme estimates suggested that practical computing would hundreds to thousands of physical qubits per logical qubit, requiring a total number of physical qubits in the order of millions for a practical computation. This is often referred to as the logical-to-physical qubit gap, and it represents one of the biggest challenges in scaling up quantum computers to the sizes needed for practical applications. However, only in the last couple of years, huge strides have been made in developing super efficient error correction codes. 2024 research by IBM reported the implementation of a new error correction code called the Grss code, which protected 12 logical qubits using only 288 physical qubits (24 qubits per logical qubit) for approximately one-million error correction cycles, demonstrating a significant reduction in the logical-to-physical qubit ratio[14] compared to other leading codes like surface code. Surface code, according to the IBM analysis, would require almost 3000 physical qubits to achieve the same task.

Microsoft and Quintinium, also in 2024, announced similar low ratio logical qubits using only 30 physical to encode 4 logical qubits (7.5 physical per logical qubit) using their own error correction techniques. However their quantum system is limited to only 32 physical qubits, so while the ratio is impressive, the absolute number of logical qubits is still very small[15]. However this highlights the rapid pace of advancements in error correction codes and the potential for more efficient schemes to significantly reduce the overhead required for fault-tolerant quantum computing.

While these results are promising, it is important to note that these are still early days in the development of practical error correction codes, and much work remains to be done to fully understand their performance and scalability in real-world quantum computing systems.

4.4. Time & Space resource requirements for practical algorithms

The particularly well known Shor's algorithm is regularly used as the benchmark for quantum computing, as it is one of the most famous algorithms that demonstrates a clear quantum advantage over classical algorithms. It has been suggested in simpler implementations that Shor's algorithm for factoring an n-bit integer can be achieved with as 'few' as $2n + 3$ *logical* qubits, with the number of gate operations required to achieve the factorization scaling as $O(n^3)$[16]. Given RSA encryption 2048-bit integers this suggests RSA encryption breaking could be done with 4099 logical qubits, but in the order of tens-of-billions of gate operations (high gate depth) consisting of modular quantum additions, multiplications and fourier transforms generally performed sequentially. Given the current state of gate speeds and coherence times, this is far beyond the capabilities of current quantum computers as the system would decohere long before the computation could complete.

Further research has investigated trading space for time, and vice versa, i.e using circuits that require more qubits to reduce the overall gate depth, or circuits that use fewer qubits but require more gate operations. However, even with these optimizations, the resource requirements for practical implementations of Shor's algorithm remain extremely high. More recent estimates suggest that factoring a 2048-bit integer using Shor's algorithm would require on the order of 20 million physical qubits when considering error correction overheads and realistic gate fidelities in a computation taking around 8 hours[17]. However no quantum computer with this physical qubit count is anywhere close to being built. On the flip side, trading time for space to minimize qubit counts, a 2024 paper proposed a method to factor a 2048-bit integer using only $n/2$ logical qubits (1730) logical qubits, but at the cost of an extremely high gate depth requiring on the order of tens of trillions of operations[18]. Even with optimistic gate speeds of 1 microsecond per gate and perfect error correction, this computation would take around 230 days to complete.

Considering the largest quantum computers to date, attention is generally brought to D-Wave's advantage system which boasts 5000+ qubits[19]. However this does not actually operate as a universal quantum computer, specifically designed for optimization problems using quantum annealing. Despite this limitation, it highlights the vast discrepancy between current qubit counts and the requirements of millions for practical applications like breaking RSA encryption.

4.5. Qubit Connectivity, Control, and Engineering Challenges

As quantum computers scale up, several engineering challenges emerge that impact their and scalability.

The physical layout of qubits is dictated by the underlying technology. Whether it's superconducting circuits, trapped ions, neutral atoms, or spin qubits each impose specific geometrical constraints on how qubits can interact. Most current platforms support only nearest neighbor connectivity, limiting the ability to perform entangling operations between distant qubits. While research into long range entanglement and modular interconnects is ongoing, practical quantum networks remain immature. As qubit density increases, cross talk becomes a significant issue, where imperfect isolation, physical proximity, and electromagnetic interference, amogst other effects can couple interactions across the system, degrading overall fidelity and thus increasing the likelihood of errors occuring.

Each qubit in a quantum processor requires precise calibration of its operating parameters, including laser frequencies, pulse shapes, and error mitigation settings. This complexity grows rapidly with the number of qubits, leading to a "complexity explosion" in control systems. The control systems required such as laser sources, and cryogenic infrastructure scale poorly in terms of energy consumption and system complexity. Additionally, hardware drift and environmental fluctuations necessitate frequent recalibration, reducing system uptime and computational efficiency.

The physical requirements for operating quantum hardware are extreme. Superconducting qubits, for example, require dilution refrigerators operating at temperatures near 10 mK, which are difficult to scale to support millions of control lines. Trapped ion and neutral atom systems depend on stable laser and vacuum setups, which are challenging to miniaturize and integrate. Fabrication of quantum devices also presents major hurdles: achieving uniform, and reproducible high fidelity qubits is an open challenge, with device variability causing significant error rates.

Together, these challenges necessitate the need for advances not only in quantum theory and algorithms, but also in engineering, materials science, and systems integration to realize quantum computation at scale.

5. Leading Quantum Hardware Platforms

5.1. Superconducting Qubits

Superconducting qubits are among the most widely used platforms for quantum computing, leveraging Josephson junctions to create two-level quantum systems. These qubits are fabricated using standard microfabrication techniques and operate at millikelvin temperatures using dilution refrigerators. Superconducting qubits offer fast gate speeds and are highly scalable, with leading implementations from IBM, Google, and Rigetti. Challenges include short coherence times and the need for complex cryogenic infrastructure. Google have claimed quantum supremacy with their Sycamore processor with a count of 53 physical qubits in 2019 [**<empty citation>**] though this claim has sparked debate due to the benchmarking used.

Superconducting qubits are among the most widely used platforms for quantum computing, leveraging Josephson junctions to create two-level quantum systems. These qubits are fabricated using standard microfabrication techniques and operate at millikelvin temperatures using dilution refrigerators. Superconducting qubits offer fast gate speeds and are highly scalable, with leading implementations from IBM, Google, and Rigetti. Challenges include short coherence times and the need for complex cryogenic infrastructure. Google have claimed quantum supremacy with their Sycamore processor with a count of 53 physical qubits in 2019 [**<empty citation>**] though this claim has sparked debate due to the benchmarking used. Similarly the more modern 105 qubit google quantum processor, Sycamore, has been developed, with claims of 'below-threshold' error corrections, though demonstrations are limited to specific cases[20]. while IBM has developed a roadmap for scaling to thousands of qubits.

5.2. Trapped Ions

Trapped ion quantum computers use electromagnetic fields to confine ions in vacuum chambers, with qubit states encoded in the internal energy levels of the ions. Laser pulses are used to manipulate and entangle the ions, achieving high-fidelity gate operations and long coherence times. Companies like IonQ and Honeywell have demonstrated significant advances in scaling trapped ion systems. The main challenges are slow gate speeds and difficulties in miniaturizing the control hardware.

5.3. Photonic Qubits

Photonic quantum computers encode qubits in the polarization, path, or number states of photons. These systems operate at room temperature and are naturally immune to many sources of decoherence. Photonic platforms are well-suited for quantum communication and networking, with companies like Xanadu and PsiQuantum leading development. Challenges include efficient photon generation, detection, and scalable entanglement.

5.4. Quantum Dots

Quantum dot qubits use the spin or charge states of electrons confined in semiconductor nanostructures. These platforms benefit from compatibility with existing semiconductor manufacturing and offer potential for high-density integration. Notable advances have been made in silicon and germanium quantum dots, but challenges remain in achieving uniformity and long coherence times across large arrays.

5.5Neutral Atoms

Neutral atom quantum computers trap individual atoms using optical tweezers, encoding qubits in atomic energy levels. These systems offer flexible connectivity and long coherence times, with companies like QuEra and ColdQuanta making progress in scaling up atom arrays. Challenges include precise control of atom positions and maintaining stability over large systems.

5.6Quantum Annealers

Quantum annealing devices, such as those developed by D-Wave, use large arrays of qubits to solve optimization problems by evolving the system toward a ground state. While not universal quantum computers, quantum annealers have demonstrated practical applications in optimization and machine learning. Limitations include restricted problem types and lack of universal gate operations.

Quantum computing hardware has rapidly evolved over the past decades, with several distinct physical platforms emerging as leaders in the race to build scalable, high fidelity quantum processors. Each platform leverages unique quantum phenomena and engineering approaches, offering different advantages and facing specific challenges. The most prominent quantum hardware platforms include:

- Superconducting Qubits
- Trapped Ions
- Photonic Qubits
- Quantum Dots
- Neutral Atoms
- Quantum Annealers

6. Future Outlook

References

1. Manin YI. Computable and Non-Computable. Moscow: Sovetskoe Radio, 1980 :128
2. Feynman RP. Simulating physics with computers. International Journal of Theoretical Physics 1982; 21:467–88
3. Deutsch D. Quantum theory, the Church–Turing principle and the universal quantum computer. Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences 1985; 400:97–117. DOI: 10.1098/rspa.1985.0070
4. Deutsch D and Jozsa R. Rapid solution of problems by quantum computation. Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences 1992; 439:553–8. Available from: https://api.semanticscholar.org/CorpusID:121702767
5. Shor P. Algorithms for quantum computation: discrete logarithms and factoring. *Proceedings 35th Annual Symposium on Foundations of Computer Science*. 1994 :124–34. DOI: 10.1109/SFCS.1994.365700
6. Grover LK. A fast quantum mechanical algorithm for database search. 1996. arXiv: quant-ph/9605043 [quant-ph]. Available from: https://arxiv.org/abs/quant-ph/9605043
7. Monroe C, Meekhof DM, King BE, Itano WM, and Wineland DJ. Demonstration of a Fundamental Quantum Logic Gate. Phys. Rev. Lett. 1995 Dec; 75(25):4714–7. DOI: 10.1103/PhysRevLett.75.4714. Available from: https://link.aps.org/doi/10.1103/PhysRevLett.75.4714
8. Häffner H, Hänsel W, Roos CF, Benhelm J, Chek-al-kar D, Chwalla M, Körber T, Rapol UD, Riebe M, Schmidt PO, Becher C, Gühne O, Dür W, and Blatt R. Scalable multiparticle entanglement of trapped ions. Nature 2005; 438:643–6
9. Monroe C, Meekhof DM, King BE, Itano WM, and Wineland DJ. Demonstration of a Fundamental Quantum Logic Gate. Physical Review Letters 1995; 75. National Institute of Standards and Technology, Boulder, Colorado:4714–7. Available from: https://tf.nist.gov/general/pdf/140.pdf
10. Srinivas R, Burd SC, Knaack HM, Sutherland RT, Kwiatkowski A, Glancy S, Knill E, Wineland DJ, Leibfried D, Wilson AC, Allcock DTC, and Slichter DH. High-fidelity laser-free universal control of trapped ion qubits. Nature 2021 Sep; 597:209–13. DOI: 10.1038/s41586-021-03809-4. Available from: http://dx.doi.org/10.1038/s41586-021-03809-4
11. Zhou YC, Ma RL, Kong Z, Li AR, Zhang C, Zhang X, Liu Y, Jiang HT, Wu ZT, Wang GL, Cao G, Guo GC, Li HO, and Guo GP. High-fidelity geometric quantum gates exceeding 99.9% in germanium quantum dots. Nature Communications 2025; 16:7953
12. Rower DA, Ding L, Zhang H, Hays M, An J, Harrington PM, Rosen IT, Gertler JM, Hazard TM, Niedzielski BM, Schwartz ME, Gustavsson S, Serniak K, Grover JA, and Oliver WD. Suppressing Counter-Rotating Errors for Fast Single-Qubit Gates with Fluxonium. PRX Quantum 2024; 5(4):040342. DOI: 10.1103/PRXQuantum.5.040342. Available from: https://link.aps.org/doi/10.1103/PRXQuantum.5.040342
13. AI GQ. Suppressing quantum errors by scaling a surface code logical qubit. Nature 2023; 614:676–81. DOI: 10.1038/s41586-022-05434-1
14. Bravyi S, Cross AW, Gambetta JM, Maslov D, Rall P, and Yoder TJ. High-threshold and low-overhead fault-tolerant quantum memory. Nature 2024; 627:778–82
15. Paetznick A, Silva MP da, Ryan-Anderson C, Bello-Rivas JM, III JPC, Chernoguzov A, Dreiling JM, Foltz C, Frachon F, Gaebler JP, Gatterman TM, Grans-Samuelsson L, Gresh D, Hayes D, Hewitt N, Holliman C, Horst CV, Johansen J, Lucchetti D, Matsuoka Y, Mills M, Moses SA, Neyenhuis B, Paz A, Pino J, Siegfried P, Sundaram A, Tom D, Wernli SJ, Zanner M, Stutz RP, and Svore KM. Demonstration of logical qubits and repeated error correction with better-than-physical error rates. 2024. arXiv: 2404.02280 [quant-ph]. Available from: https://arxiv.org/abs/2404.02280
16. Beauregard S. Circuit for Shor's algorithm using 2n+3 qubits. 2003. arXiv: quant-ph/0205095 [quant-ph]. Available from: https://arxiv.org/abs/quant-ph/0205095
17. Gidney C and Ekerå M. How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits. Quantum 2021 Apr; 5:433. DOI: 10.22331/q-2021-04-15-433. Available from: http://dx.doi.org/10.22331/q-2021-04-15-433
18. Chevignard C, Fouque PA, and Schrottenloher A. Reducing the Number of Qubits in Quantum Factoring. *Advances in Cryptology – CRYPTO 2025*. Ed. by Tauman Kalai Y and Kamara SF. Cham: Springer Nature Switzerland, 2025 :384–415
19. King AD, Lanting T, Harris R, Amin MH, Andriyash E, Isakov SV, Johnson MW, Bunyk PI, Rich C, Thom MC, et al. Quantum critical dynamics in a 5,000-qubit programmable spin glass. Nature 2023; 618:500–5. DOI: 10.1038/s41586-023-05867-2. Available from: https://www.nature.com/articles/s41586-023-05867-2
20. Acharya R, Abanin DA, Aghababaie-Beni ea, AI GQ, and Collaborators. Quantum error correction below the surface code threshold. Nature 2025; 638:920–6

Year	**Qubit Count**	**Description**
1995	2	First experimental demonstration of a quantum logic gate using trapped ions[9]
2001	7	Implementation of Shor's algorithm on a 7-qubit NMR quantum computer[**Vandersypen2001**]
2005	8	Demonstration of an 8-qubit register using trapped ions[**https://www.nature.com/articles/nature0427**
2011	5	D-Wave Systems announced a 512-qubit quantum annealer[**DWave2011**]
2016	16	IBM unveiled a 16-qubit superconducting quantum processor[**IBM2016**]
2017	20	Google announced a 20-qubit superconducting quantum processor[**Google2017**]
2019	53	Google's Sycamore processor achieved quantum supremacy with 53 qubits[**Arute2019**]
2020	65	Honeywell announced a 65-qubit trapped-ion quantum computer[**Honeywell2020**]
2021	127	IBM unveiled a 127-qubit superconducting quantum processor[**IBM2021**]
2022	433	IonQ announced a 433-qubit trapped-ion quantum computer[**IonQ2022**]
2023	1000+	Various companies announced plans for quantum processors with over 1000 qubits[**Various2023**]
2024	2000+	Continued advancements with processors exceeding 2000 qubits[**Various2024**]
2025	5000+	Leading companies project quantum processors with over 5000 qubits[**Various2025**]

Table 1: Key advancements in qubit counts from 1995 to 2025 and their corresponding platforms.