Quantum Computing: Foundations, Hardware Platforms, Error Correction, and Applications—A Survey

Dr. M. Reka[1], Dr. S. Vanitha[2], Dr. V. Indhumathi[3], Dr. G. Arutjothi[4]

[1]Assistant Professor, Computer Applications Sona College of Arts and Science, Salem drmreka@gmail.com
[2]Assistant Professor, Computer Science, Sona College of Arts and Science, Salem, vanithas84@gmail.com
[3]Assistant Professor, Computer Applications Sona College of Arts and Science, Salem, sabarishindhu@gmail.com
[4]Assistant Professor, Computer Applications Sona College of Arts and Science, Salem, garutjothi@gmail.com

Abstract:
Quantum computing uses quantum states and operations to process information in ways that differ fundamentally from classical computation. Potential advantages have been established for selected tasks, including integer factorization, unstructured search, and the simulation of quantum systems. However, practical computation remains constrained by hardware noise, limited qubit connectivity, control errors, and the substantial overhead required for fault tolerance. This survey introduces the basic computational model and reviews leading hardware platforms, quantum error correction, representative algorithms, and application areas. It also discusses benchmarking and the gap between algorithmic promise and demonstrated practical advantage. Rather than treating quantum computing as a general replacement for classical computing, the paper emphasizes its likely role as a specialized resource within hybrid computational workflows. Open challenges include scaling reliable logical qubits, reducing error-correction costs, developing useful applications, and establishing fair and reproducible comparisons with classical methods.

Keywords: quantum computing; qubits, quantum algorithms, quantum error correction, quantum hardware, quantum advantage

1. Introduction

Classical computers encode information in bits, which take values conventionally represented as 0 or 1. Quantum computers process quantum states, whose behavior is governed by principles such as superposition, interference, and entanglement. These properties enable quantum algorithms to manipulate probability amplitudes in ways that can provide speedups for particular computational problems.

The field has progressed from theoretical proposals to programmable devices containing physical qubits. Yet the existence of a quantum processor does not, by itself, imply that it can outperform a classical computer on a useful task. Physical qubits are imperfect, and quantum operations are vulnerable to environmental noise and control inaccuracies. In many cases, correcting these errors requires many physical qubits to represent one more reliable logical qubit.

This survey outlines the foundations and current technical landscape of quantum computing. It focuses on four connected questions: how quantum information is represented and manipulated, what kinds of hardware implement quantum operations, how errors can be controlled, and which applications may benefit.

The central argument is that progress should be assessed not only by physical-qubit counts, but also by logical performance, end-to-end resource costs, and comparison with strong classical baselines.

2. Scope and Survey Approach

This paper is a narrative survey, not a systematic review or meta-analysis. It organizes established concepts and representative research directions around quantum computing foundations, hardware, error correction, algorithms, and applications. The cited works are selected as foundational or illustrative sources,the paper does not claim exhaustive coverage of all publications or a fully reproducible literature search.

3. Quantum Information and Computation

3.1 Qubits and quantum states

A qubit is a two-level quantum system. Its pure state can be expressed a

|¥| = α|0| + β|1|

A measurement in the computational basis yields 0 or 1, with probabilities determined by the squared magnitudes of the corresponding amplitudes.

$|\omega| = |\alpha|^2 + |\beta|^2 = 1$

Unlike a classical bit, a qubit cannot be interpreted as simply holding both classical values in an ordinary probabilistic mixture. Quantum states can exhibit interference: operations may reinforce some outcomes and suppress others. Quantum algorithms are designed to exploit this structure before measurement.

For systems of multiple qubits, the joint state belongs to a tensor-product space. Some states cannot be factored into independent states of their constituent qubits; these are entangled states. Entanglement is an important resource in quantum information processing, although its presence alone does not guarantee a computational advantage.

3.2 Quantum gates and circuits

Quantum operations are commonly described using unitary matrices. A quantum circuit applies a sequence of one- and multi-qubit gates, followed by measurement. Since unitary operations are reversible, irreversible tasks such as erasing information must be handled carefully, often by retaining intermediate results or using ancillary qubits.

A universal gate set can approximate arbitrary unitary transformations to a desired precision. In practice, the choice of gate set depends on the hardware platform and affects circuit depth, compilation overhead, and the number of error-prone operations. Connectivity also matters: when two qubits cannot interact directly, additional gates may be needed to move quantum information or to compile an operation into the device's native instruction set.

3.3 Sources of computational speed up

Quantum algorithms do not accelerate every computation. Known speedups are associated with particular mathematical structures or computational models. Shor's algorithm provides a polynomial-time method for factoring integers on an ideal fault-tolerant quantum computer, whereas the best-known classical methods have sub-exponential complexity. Grover's algorithm offers a quadratic query advantage for unstructured search, not an exponential one.

Quantum simulation is another important motivation. Because quantum systems can be difficult to represent efficiently on classical computers, quantum processors may offer a natural platform for modeling molecular and material behavior. Whether a specific simulation is practically advantageous depends on the system, required precision, circuit resources, and the capabilities of classical alternatives.

4. Quantum Hardware Platforms

No single physical implementation currently satisfies every desirable requirement. Platforms differ in qubit coherence, gate performance, connectivity, operating temperature, control complexity, and prospects for manufacturing at scale.

4.1 Superconducting circuits

Superconducting qubits are electrical circuits operated at cryogenic temperatures. They support relatively fast gates and have become a widely used platform for programmable processors. Their challenges include frequency variation, crosstalk, limited connectivity in many designs, and the engineering demands of cryogenic control and readout.

4.2 Trapped ions

Trapped-ion systems use internal energy levels of ions as qubits and electromagnetic fields to control them. They can achieve high-fidelity operations and support interactions between qubits that are not physically adjacent. Gate speeds are generally slower than those of superconducting devices, and scaling large ion collections while maintaining control and connectivity remains an engineering challenge.

4.3 Neutral atoms

Neutral-atom systems trap atoms, often in optical arrays, and use controlled interactions to implement quantum operations. Their flexible array geometries and potential for large-scale qubit placement are notable strengths. Current research addresses gate fidelity, atom loss, state preparation, and reliable operation across larger devices.

4.4 Photonic systems

Photonic quantum computing encodes information in properties of light, including polarization, path, or time bins. Photons are well suited to communication and can operate without cryogenic matter-based qubits in some settings. The difficulty of generating, storing, and interacting photons deterministically is a major challenge; approaches use measurement, optical components, and specialized resource states.

4.5 Comparing platforms

Hardware comparisons should use more than qubit count. Relevant measures include gate and measurement error, coherence time, connectivity, operation speed, device stability, and the ability to execute a target workload. Metrics such as quantum volume or algorithm-specific benchmarks can provide useful information, but no single score captures all aspects of practical utility.

5. Quantum Error Correction

5.1 Error-correcting codes

Quantum error-correcting codes encode logical qubits into entangled states of physical qubits. Stabilizer codes are a broad family in which operators are measured to identify error syndromes. The surface code is especially influential because it is compatible with local interactions on a two-dimensional array and can tolerate a nonzero error rate under suitable conditions.

A threshold theorem states, under specified assumptions, that sufficiently low physical error rates can permit arbitrarily long computation through error correction. The theorem does not mean that fault-tolerant computation is inexpensive. Practical costs depend on the physical error rates, architecture, target algorithm, desired failure probability, and decoding performance.

5.2 Fault tolerance and logical qubits

A logical qubit is an encoded unit of quantum information designed to be more reliable than its physical components. Fault-tolerant computation requires logical operations, state preparation, measurement, and error correction to be performed so that individual faults do not spread uncontrollably.

For many algorithms, the number of physical qubits required can greatly exceed the number of logical qubits. Some operations, including non-Clifford gates in common fault-tolerant schemes, can carry significant resource costs. Consequently, estimating an algorithm's practical feasibility requires an end-to-end resource analysis rather than a count of abstract logical operations alone.

6. Representative Quantum Algorithms

6.1 Shor's algorithm

Shor's algorithm reduces integer factoring to period finding and uses the quantum Fourier transform to identify the relevant periodic structure. Its theoretical significance is substantial because large-scale fault-tolerant implementations could threaten widely used public-key cryptographic systems based on factoring or related number-theoretic problems. However, factoring cryptographically relevant integers requires far more reliable resources than current noisy devices provide.

6.2 Grover's algorithm

Grover's algorithm searches an unstructured space of NNN possibilities using on the order of

$$Q_{\text{quantum}} = O(\sqrt{N}), \qquad Q_{\text{classical}} = O(N)$$

oracle queries, compared with order NNN classical queries in the black-box setting. The advantage is quadratic, and real applications must account for the cost of implementing the oracle, loading data, and verifying the output.

6.3 Quantum simulation

Quantum simulation seeks to model quantum systems using controllable quantum devices. Approaches include digital simulation with gate sequences and analog simulation using a system whose natural dynamics approximate the target model. Potential applications include electronic structure, condensed matter, and chemical reaction modeling. Practical performance depends on accuracy requirements and the efficiency of state preparation and measurement. The below figure shows the simulation of Quantum State Evolution.

6.4 Variational algorithms

Variational quantum algorithms use a parameterized quantum circuit together with a classical optimization loop. The quantum device evaluates a cost function, and a classical optimizer updates circuit parameters. These methods are compatible with relatively shallow circuits, making them attractive for noisy devices. However, their performance can be hindered by measurement overhead, optimization instability, noise, and the difficulty of demonstrating an advantage over classical heuristics. Their usefulness must therefore be established for specific problem instances.

7. Application Areas and Evidence of Utility

7.1 Optimization and machine learning

Quantum algorithms have been proposed for combinatorial optimization, sampling, and machine-learning tasks. Claims of benefit vary widely because results depend on problem encoding, data access, hardware

noise, and the strength of classical baselines. A quantum method should be compared against well-optimized classical approaches on the same task and under comparable resource assumptions.

7.2 Cryptography and security

Large-scale fault-tolerant quantum computers would affect public-key cryptography based on integer factorization and discrete logarithms. This motivates the development and deployment of post-quantum cryptographic standards. The security transition is a present-day engineering and policy concern, even though a machine capable of breaking relevant deployed keys is not currently available.

8. Practical Quantum Advantage

A claim of quantum advantage should specify the task, input size, accuracy, resource assumptions, and classical comparison. Demonstrating that a quantum device produces a result that is difficult to reproduce classically is not by itself proof of practical usefulness. The task may have no direct application, or a classical algorithm may later reproduce the result more efficiently.

A meaningful evaluation should report the full workflow: data preparation, circuit compilation, execution, error mitigation or correction, measurement, and post-processing. It should also distinguish between a theoretical asymptotic speedup and a measured advantage at a particular scale. Reproducible benchmarks and transparent resource estimates are essential for progress.

9. Open Challenges

Several challenges cut across hardware and algorithms:

Scalable reliability: Increasing qubit count while preserving low error rates and stable operation.

Error-correction overhead: Reducing the physical resources and time needed to maintain logical qubits.

Useful workloads: Identifying problems where quantum algorithms offer a measurable benefit over strong classical methods.

Data and measurement costs: Managing state preparation, repeated measurements, and classical information transfer.

Benchmarking: Developing metrics that are application-relevant, reproducible, and resistant to misleading comparisons.

Co-design: Matching algorithms, error-correction codes, hardware connectivity, and control systems.

Security readiness: Preparing cryptographic infrastructure for the possibility of future large-scale quantum attacks.

Conclusion

Quantum computing has developed into a diverse research field spanning physics, computer science, engineering, and applied mathematics. Its strongest prospects are specialized quantum devices may outperform classical computers on selected problems, particularly where quantum structure can be exploited or where quantum systems themselves are the object of study. Realizing these prospects depends on building reliable logical qubits, lowering fault-tolerant resource costs, and validating applications against competitive classical methods.

The field should therefore be evaluated through careful, task-specific evidence rather than broad claims based on qubit counts alone. Continued progress will depend on joint advances in hardware, error correction, algorithm design, and reproducible benchmarking.

REFERENCES:

1.Feynman, R. P. “Simulating Physics with Computers.” International Journal of Theoretical Physics, 21, 467–488 (1982).
2.Shor, P. W. “Algorithms for Quantum Computation: Discrete Logarithms and Factoring.” Proceedings of the 35th Annual Symposium on Foundations of Computer Science, 124–134 (1994).
3.Grover, L. K. “A Fast Quantum Mechanical Algorithm for Database Search.” Proceedings of the 28th Annual ACM Symposium on Theory of Computing, 212–219 (1996).
4.Nielsen, M. A., and Chuang, I. L. Quantum Computation and Quantum Information. Cambridge University Press (2000).
5.Preskill, J. “Quantum Computing in the NISQ Era and Beyond.” Quantum, 2, 79 (2018).
6.Arute, F., et al. “Quantum Supremacy Using a Programmable Superconducting Processor.” Nature, 574, 505–510 (2019).
7.Bravyi, S., and Kitaev, A. “Quantum Codes on a Lattice with Boundary.” arXiv:quant-ph/9811052 (1998).
8.Aharonov, D., and Ben-Or, M. “Fault-Tolerant Quantum Computation with Constant Error Rate.” SIAM Journal on Computing, 38(4), 1207–1282 (2008).
9.Preskill, J. “Reliable Quantum Computers.” Proceedings of the Royal Society A, 454, 385–410 (1998).
Cerezo, M., et al. “Variational Quantum Algorithms.” Nature Reviews Physics, 3, 625–644 (2021).