Open Access Research Journal of
Engineering and Technology
Journals home page: https://oarjpublication/journals/oarjet/
ISSN: 2783-0128 (Online)

(RESEARCH ARTICLE)

From qubits to general-purpose computing: quantum advantage and its limits

Claude Cockfield [1] and Moses Garuba [2,*]

[1] *Department of Electrical Engineering and Computer Science, Howard University, Washington, DC, United States.*
[2] *Department of Electrical and Computer Engineering, Hampton University, Hampton, Virginia, United States.*

Open Access Research Journal of Engineering and Technology, 2026, 11(01), 023–033

Publication history: Received on 13 June 2026; revised on 20 July 2026; accepted on 22 July 2026

Article DOI: https://doi.org/10.53022/oarjet.2026.11.1.0054

Abstract

Quantum computing has been championed as a ground-breaking technology due to its significant computational advantage over classical computing systems. At present, fault-tolerant, large-scale universal quantum computing remains elusive, and these issues remain central challenges in engineering, physics, and computer science. In theory, quantum advantage offers the potential to make many currently intractable computational problems practically solvable, with profound implications across science, industry, and computing. This paper examines the contemporary case for general-purpose quantum computing and the prevailing challenges that presently prevent its widespread adoption. It surveys the theoretical foundations of computation and quantum mechanics that underpin quantum computers, describes the quantum computational model and a working implementation on IBM's Qiskit platform, and evaluates the persistent barriers of noise, decoherence, and scalability that constrain quantum systems today. We conclude that quantum computing is best positioned, in the near term, as a specialized co-processor rather than a general-purpose replacement for classical computing.

Keywords: Quantum Computing; Qubit; Computer Architecture; Quantum Logic Gates; Qiskit.

1. Introduction

Artificial Intelligence (AI), big data, and Machine Learning (ML) increasingly demand computing capability that scales with dataset size and model complexity for effective analysis, training, and inference [1]. However, the rate of improvement in processor speed and energy efficiency that characterized Moore's Law for five decades has waned, as miniaturization and clock-speed gains have reached physical limits. The industry has responded by exploiting parallelism, improved materials, and novel hardware and software architectures [2].

Quantum computing offers an alternative path to continued performance growth. The laws of quantum mechanics give rise to superposition, the property underlying the exponential algorithmic advantage that quantum computers exhibit over classical systems. Quantum computation was first proposed in the early 1980s by Benioff [3] and Feynman [4]; David Deutsch subsequently proposed the first general-purpose quantum computer in 1985 [5], and rudimentary quantum logic gates were experimentally demonstrated at the National Institute of Standards and Technology (NIST) and the University of Innsbruck in the mid-1990s [6], [7]. Despite these advances, general-purpose quantum computing remains an elusive goal [8].

This paper examines the contemporary case for general-purpose quantum computing and the challenges that presently prevent its widespread adoption. Section 2 reviews the theoretical foundations of computation and quantum mechanics that underpin quantum computers. Section 3 describes the quantum computational model, its available infrastructure, and a working implementation on IBM's Qiskit platform. Section 4 presents results and discussion concerning Shor's algorithm and the open challenges of noise, decoherence, and scalability. Section 5 concludes.

* Corresponding author: Moses Garuba

2. Background and Theoretical Foundations

2.1. Historical Development of Computing and Quantum Theory

The third industrial revolution transformed the global economy from a manufacturing economy to an information economy [9], driving unprecedented economic growth and rising living standards worldwide [10]. In the mid-twentieth century, contributions from mathematicians, engineers, and physicists established computer science as a distinct discipline [11]. Alan Turing formalized computation and its theoretical limits [12], and John von Neumann's EDVAC report described the stored-program architecture that remains dominant today [13], building on work by J. Presper Eckert and John Mauchly. Claude Shannon later demonstrated that all information could be digitized, and that digitized data could be stored and manipulated by computing machines [14].

Subsequent advances in semiconductor design, algorithmics, and digital communications fueled a half-century of compounding gains in processing power. In the 1960s, Gordon Moore observed that the density of integrated-circuit transistors was doubling annually [15], a trend that quickly became both an informal rule and a self-fulfilling industry target. Dennard scaling [16] — the principle that transistor energy requirements fall proportionately as component size shrinks — allowed exponential performance gains at constant energy cost. By the early twenty-first century, however, concerns about the end of Moore's Law intensified [17], as the industry confronted combined scaling [18], thermal, and memory-bandwidth limits [19].

In response, the industry shifted toward execution parallelism as the primary means of increasing performance. Quantum computers extend this logic further: they are inherently parallel, processing an exponential number of states simultaneously, in contrast to classical computers, where each storage unit holds a single value at any instant. For n quantum bits (qubits), the total number of representable states is 2^n.

2.2. Computational Theory: Algorithms, Turing Machines, and Complexity

An algorithm is an ordered set of instructions describing how to perform a task; higher-level, platform-agnostic instructions are translated into platform-specific instructions for execution. The Turing machine remains the standard model for analyzing and classifying algorithms in computer science [12], while lambda calculus [20] provides a functionally equivalent, function-centric alternative. Turing machines, based on the stored-memory model of computing [13], remain the most general and powerful model of computation for which physical implementations exist today.

The Chomsky hierarchy classifies computational devices into nested classes of increasing power, where each layer implements a “language” of a given type (Table 1). Turing machines implement Type-0 languages, which are in general undecidable [21] — meaning that not every problem expressible in that class can be resolved computationally. The Busy Beaver problem illustrates this: no computable function can determine whether a given n-state Turing machine is a “busy beaver,” the machine that writes the greatest number of 1s to its tape before halting, for that value of n [22].

Table 1 The Chomsky Hierarchy

Type	Language Class
Type-0	Recursively Enumerable
Type-1	Context-Sensitive
Type-2	Context-Free
Type-3	Regular

An algorithm that is decidable may nonetheless be intractable in practice due to excessive time or space requirements, as captured by complexity classes such as Nondeterministic Polynomial (NP) and Exponential Time (EXPTIME). Beyond even EXPTIME, functions such as Ackermann's are inherently recursive and cannot be expressed as primitive recursive functions. Quantum computers are Turing-equivalent: they possess no computational capacity beyond that of classical Turing machines and remain subject to the same limits for unrecognizable and undecidable problems. However, a potential exponential speedup for certain problem classes [23] means quantum computation could render some classically intractable algorithms tractable. This combinatorial quantum advantage is not general-purpose; it is best

suited to narrow applications such as integer factorization, DNA sequencing, machine learning, optimization, and simulation of protein folding and materials [24].

2.3. Principles of Quantum Mechanics

In the seventeenth century, Sir Isaac Newton laid the foundations of classical mechanics, which models the universe as a deterministic, mechanistic system exhibiting certainty and predictability [25]. Roughly two centuries later, Max Planck established that energy is quantized into discrete packets, earning the Nobel Prize in Physics for this foundational contribution to quantum theory [26]; Albert Einstein received the Nobel Prize in 1921 for his quantum-mechanical description of the photoelectric effect [27]. Their work, later extended by Bohr, Heisenberg, and Schrödinger, established quantum mechanics as the dominant theory of the physical world [28], replacing the determinism of classical physics with probabilistic processes, non-locality, and revised notions of causality.

Quantum theory defines a taxonomy of particles [29]: fermions, the constituents of matter, further divided into quarks and leptons such as electrons; and bosons, the particles that mediate forces, including the Higgs boson, which imparts mass. At quantum scales, several properties underpin the operation of quantum computers, summarized in Table 2.

Table 2 Key Quantum Mechanical Properties Underlying Quantum Computation

Property	Description
Superposition	A particle exists in multiple concurrent potential states simultaneously, each with an associated probability, until measurement induces decoherence to a single classical value. The double-slit experiment demonstrates this wave-particle duality: even solitary photons exhibit wave-like interference when able to traverse multiple paths simultaneously [30].
Entanglement	Particles become bound such that their combined quantum state behaves as a single system regardless of separation distance; Einstein termed this “spooky action at a distance” [31].
Non-Locality	Correlations between entangled particles appear instantaneously across distance. Bell's theorem shows this is almost certainly not explained by local hidden variables, though no information is transferred in the process [32].
Coherence / Decoherence	Coherence is the maintenance of fixed phase relationships within a wavefunction, enabling superposition and entanglement; decoherence occurs when environmental interaction disrupts these relationships.
Quantum Tunneling	Non-zero probability amplitudes at classically “impossible” positions allow a particle to appear on the far side of a barrier; this manifests in modern processors as electrons tunneling through circuit barriers [33].
Uncertainty Principle	The position and momentum of a particle cannot both be measured with arbitrary precision; increasing certainty in one measurement increases uncertainty in the other [34].
Interference	Particles in superposition can be driven toward decoherence, or have their outcome probabilities reshaped, through interference with other particles [35].

3. Materials and Methods: The Quantum Computing Model and Implementation

3.1. Qubits and Quantum Logic Gates

Quantum computers harness the properties above through the qubit, the quantum analogue of the classical bit. Unlike a classical bit, which holds a value of 0 or 1 at any instant, a qubit can represent both values simultaneously. A qubit in state $|1\rangle$, or in an equal superposition, is written $1/\sqrt{2}|0\rangle + 1/\sqrt{2}|1\rangle$, where $1/\sqrt{2}$ is a probability amplitude and $|0\rangle$, $|1\rangle$ are base states. The probability of measuring a given base state is the squared absolute value of its amplitude, and the sum of all such probabilities must equal 1, the Born rule [36]. For n qubits, the register represents 2^n simultaneous states — 32 qubits, for example, span roughly 4.3 billion states via corresponding complex wave amplitudes [37].

Quantum computation models define both quantum and classical operations on qubits, implemented through quantum logic gates analogous to classical logic gates (Table 3).

Table 3 Quantum Logic Gates

Gate	Description	Target	Exploits
Hadamard	Transforms a binary value into a superposition.	Qubit	Superposition
Pauli-X	Flips the qubit state (and its associated probabilities).	Qubit	Amplitude
Pauli-Y	Flips the qubit state and applies a phase flip.	Qubit	Amplitude
Pauli-Z	Applies a phase flip to the $\|1\rangle$ component ($\|1\rangle \rightarrow -\|1\rangle$), altering subsequent interference effects.	Qubit	N/A
CNOT	Controlled-NOT; links two qubits to create entanglement.	Qubit pair	Entanglement

3.2. Reversibility, Linearity, and Redundancy Constraints

Two physical constraints govern quantum computation. First, because quantum mechanics is unitary, operations on qubits in superposition must be reversible [38]; irreversible operations necessitate information loss, which requires environmental interaction and thus induces decoherence. Second, quantum coherence is inherently linear, since superposition requires that quantum states be additive, a property that holds only in linear systems [39], [40]. Consequently, systems in coherence cannot directly execute non-linear operations, such as branching on an intermediate computed value, within the same coherent execution; quantum computers can execute classical operations such as storage and branching, but only outside coherency contexts. Quantum advantage is therefore constrained to problems that can be expressed, at least partially, in effectively linear terms.

A further practical constraint is redundancy: because some qubits must be dedicated to error correction [41], the total number of qubits in a system exceeds the number available for computation. The array of qubits available to a computation — the quantum register — must simultaneously interface with the environment to execute programs while being shielded from unwanted interaction with non-quantum influences and from interference by other qubits. Physical qubits have been realized using ions, electrons, and photons. Because the register stores relative probabilities that must sum to 1, quantum computers are best suited to algorithms that produce a single terminal answer, or to problems whose solution is inferred from the statistical distribution of repeated runs.

3.3. Application Domains and Available Infrastructure

Quantum models of computation are best suited to three broad application areas [42]: large-number factorization, particularly the prime factorization underlying encryption schemes such as RSA; classically intractable combinatorial and optimization problems; and simulation of physical processes relevant to the development of new materials and treatments.

Quantum computers remain expensive: typical long-term research and operational costs are on the order of USD 1 billion [43]. Table 4 lists leading providers of publicly accessible quantum computing infrastructure.

Table 4 Quantum Computing Infrastructure Providers

Company	Access	URL
AWS	AWS Braket	https://aws.amazon.com/braket/
D-Wave	Quantum Leap	https://www.dwavesys.com/
Google	Google Cloud	https://quantumai.google/
Honeywell	Private commercial access to Model H1	https://www.honeywell.com/us/en/company/quantum
IBM	IBM Cloud	https://quantum.ibm.com/
IonQ	AWS Braket / Azure	https://ionq.com/
Microsoft	Azure Cloud	https://quantum.microsoft.com/
Rigetti	—	https://www.rigetti.com/

3.4. Implementation: IBM Qiskit Demonstration

To illustrate the concepts above, this section reproduces a minimal quantum program on IBM's Qiskit platform [44], which generates a Bell-state entanglement between two qubits. Quantum program execution generally follows four phases (Table 5).

Table 5 Quantum Algorithm Execution Phases

Phase	Description
Init	Set up resources and initial non-quantum values.
Calculation	Enter superposition and manipulate qubit properties via quantum operations.
Measure	Perform a non-quantum measurement that collapses the waveform into single classical values.
Repetition	Repeat n times when the answer is inferred from a statistical distribution across multiple runs.

```
# Build a 2-qubit circuit and apply Hadamard + CNOT
from qiskit import QuantumCircuit
qc = QuantumCircuit(2)
qc.h(0)        # superposition on qubit 0
qc.cx(0, 1)    # entangle qubits 0 and 1
qc.draw(output='mpl')
```

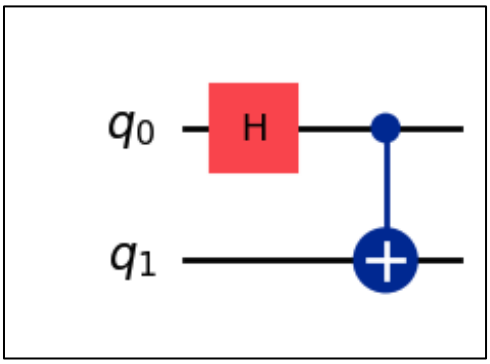

Figure 1 Two-qubit circuit applying a Hadamard gate followed by a CNOT gate [44].

```
# Define observables and estimate expectation values
from qiskit.quantum_info import Pauli
from qiskit_aer.primitives import Estimator
observables = [Pauli(o) for o in ['ZZ','ZI','IZ','XX','XI','IX']]
estimator = Estimator()
job = estimator.run([qc] * len(observables), observables)
values = job.result().values

# Plot expectation values
import matplotlib.pyplot as plt
plt.plot(['ZZ','ZI','IZ','XX','XI','IX'], values, '-o')
plt.xlabel('Observables'); plt.ylabel('Expectation value'); plt.show()
```

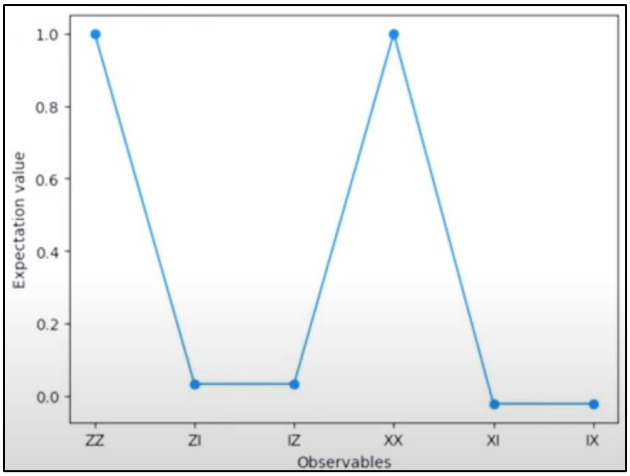

Figure 2 Expectation values of the six Pauli observables measured on the Bell-state circuit [44].

Execution produces the Bell-state superposition $1/\sqrt{2}(|00\rangle + |11\rangle)$ and yields an expectation value of ZZ = 1 (Figure 2), indicating that the two qubits are fully correlated — both measuring 1 or both measuring 0 on any given run. A value of ZZ = 1 indicates correlation but does not, by itself, prove entanglement.

Direct measurement of a quantum circuit causes decoherence; the ZZ observable instead infers correlation without direct measurement. Because each measurement yields only a single classical bit per qubit, circuits must be executed repeatedly to build a result distribution from which the answer is inferred — the repetition-and-sampling approach exemplified by Shor's algorithm, discussed in Section 4.1.

IBM's Qiskit guides describe program construction in four steps [44]: mapping the classical problem to quantum circuits and operators; optimizing those circuits for the target hardware, including qubit routing, conversion to native basis gates, and simulator pre-validation; executing the optimized circuits on hardware via the Sampler or Estimator primitives, with configurable error suppression; and post-processing the resulting outputs — through visualization, readout-error mitigation, or post-selection on physical properties such as spin or parity — to obtain the final result.

4. Results

4.1. Shor's Factorization and the Limits of Quantum Speedup

The best known classical algorithm for integer factorization, the general number field sieve, is sub-exponential but super-polynomial, rendering factorization of large integers computationally infeasible classically. This difficulty underlies encryption schemes such as RSA [45]; factoring a 2048-bit RSA key is infeasible for classical computers. Shor's algorithm [46] is a quantum algorithm for integer factorization that offers a theoretical super-polynomial time advantage over the best classical methods, and could in principle be used to factor RSA keys.

Theoretical proofs notwithstanding, Shor's algorithm has not been demonstrated on real quantum hardware for cryptographically relevant values of n, principally because it requires more qubits than current devices provide, and because it is highly sensitive to interference: the algorithm depends on coherently summing an exponentially large number of amplitudes that interfere constructively to build sharp peaks in the output probability distribution, and this 'exponential coherence' is extremely vulnerable to noise [47]–[49]. In anticipation of large-scale quantum factorization, quantum-resistant cryptographic standards based on lattice structures have already emerged [50].

This result crystallizes a broader pattern in the findings above: quantum advantage is real and demonstrable at small scale (Section 3.4), but scaling it to cryptographically or industrially meaningful problem sizes is constrained by the same physical phenomena — interference and decoherence — that make quantum computation possible in the first place.

4.2. Open Challenges: Noise, Decoherence, and Scalability

Quantum algorithms exploit specific phenomena — superposition, interference, entanglement — and are therefore inherently suited only to problems that map onto operations based on those phenomena. Even where an algorithm is

provably sound, a substantial gap remains between idealized mathematical models and what is achievable on physically engineered quantum hardware [8], [23], [51], [52].

Noise and error recovery: Qubits are sensitive to interference; interactions with the external environment, and even with other particles in superposition, can shift the probability distribution that encodes a quantum state and thereby introduce error, analogous to bit-errors in classical computing [53]. Error-correcting qubits mitigate these effects but consume additional qubits that would otherwise be available for computation, and progress remains slow as viable qubit counts stay technologically constrained [54].

Decoherence and stability: Quantum states decay to non-quantum states upon interaction with non-quantum systems, including ambient radiation and cosmic particles [55]; decoherence causes loss of quantum-state information and can fully disrupt processing [56].

Scaling: Real-world quantum advantage requires large qubit counts, yet redundancy for error correction and other overhead reduces net computational qubits, and qubit error rates increase with qubit count [57]. IBM nonetheless plans to build a 100,000-qubit quantum computer by 2033 [58].

Algorithms: The portfolio of viable quantum algorithms remains narrow: real-world problems must be mapped onto the limited class of problems that quantum computers solve directly, and, as Shor's algorithm demonstrates, theoretical correctness does not guarantee practical executability on current hardware. Because quantum mechanics is inherently linear [59], quantum advantage is further diminished for problems that do not admit an effectively linear formulation [60].

5. Discussion: Prospects for General-Purpose Adoption

Taken together, these results indicate that quantum computing, despite decades of active research since the 1980s and a demonstrable concrete advantage in Shor's algorithm [46], is not presently suited to general-purpose computing. The central obstacles are scalability and stability — interference and decoherence — rather than any deficiency in the theoretical case for quantum advantage. Realizing wholesale impact on general-purpose computing tasks will require effective algorithms that render tractable a meaningful subset of NP and EXPTIME problems; the emergence of quantum-resistant, lattice-based cryptography [50] shows that some algorithmic structures already resist quantum exploitation, underscoring that the transition, where it occurs, will be uneven across problem domains.

Consistent with this picture, industry consensus anticipates that quantum computing will initially serve as a remote or local co-processor — analogous to the role GPUs play today — rather than a general-purpose replacement for classical computing [67]. Quantum computing services are already available from major cloud providers, facilitating exactly this hybrid model of adoption [68]–[70].

6. Conclusion

This paper has examined the theoretical foundations, computational model, and present-day limitations of general-purpose quantum computing. Quantum computers demonstrably exploit superposition, entanglement, and interference to achieve algorithmic speedups unavailable to classical machines, and a working implementation on a real quantum cloud platform confirms that these effects can be harnessed in practice at small scale. At the same time, the same physical phenomena that enable quantum advantage — sensitivity to interference and susceptibility to decoherence — currently prevent that advantage from scaling to the qubit counts and stability levels that general-purpose computing would require.

The evidence therefore suggests that, in its present state, quantum computing is not a general-purpose computing technology, but a specialized accelerator best suited to narrow problem classes such as factorization, combinatorial optimization, and physical simulation. Continued progress in error correction, qubit fabrication, and algorithm design will determine how much of that narrow advantage can be broadened over time. In the interim, the most realistic path to practical impact is a hybrid model in which quantum processors serve as co-processors to classical computing infrastructure.

Compliance with ethical standards

Acknowledgments

The author declares that no external funding was received in support of this research.

Disclosure of conflict of interest

The author declares no conflict of interest.

Statement of informed consent

The present research does not contain any studies performed on animal or human subjects by the author.

References

[1] J. Rane, S. K. Mallick, O. Kaya and N. L. Rane, "Artificial intelligence, machine learning, and deep learning in cloud, edge, and quantum computing: A review of trends, challenges, and future directions," In Future Research Opportunities for Artificial Intelligence in Industry, no. 4.0 and 5.0, pp. 1-38, 2024.

[2] S. Naffziger, J. Warnock and H. Knapp, "SE2 when processors hit the power wall (or 'when the CPU hits the fan')," 2005 IEEE International Digest of Technical Papers, pp. 16-17, February 2005.

[3] P. Benioff, "The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines," Journal of Statistical Physics, vol. 22, p. 563–591, May 1980.

[4] R. P. Feynman, "Simulating physics with computers," International Journal of Theoretical Physics, vol. 21, p. 467–488, 7 May 1981.

[5] D. Deutsch, "Quantum theory, the Church-Turing principle and the universal quantum computer," Proceedings of the Royal Society of London A, vol. 400, pp. 97-117, 1985.

[6] C. Monroe, D. M. Meekhof, B. E. King, W. M. Itano and D. J. Wineland, "Demonstration of a Fundamental Quantum Logic Gate," Physical Review Letters, vol. 75, no. 25, p. 4714–4717, 18 December 1995.

[7] J. I. Cirac and P. Zoller, "Quantum Computations with Cold Trapped Ions," Physical Review, vol. 74, no. 20, p. 4091–4094, 1 March 1995.

[8] A. D. Corcoles, A. Kandala, A. Javadi-Abhari, D. T. McClure, A. W. Cross, K. Temme, P. D. Nation, M. Steffen and J. M. Gambetta, "Challenges and Opportunities of Near-Term Quantum Computing Systems," Proceedings of the IEEE, vol. 108, no. 8, p. 1338–1352, Aug 2019.

[9] J. Rifkin, The third industrial revolution, Palgrave Macmillan, 2011.

[10] S. Andrew, Exploring the Linkages Between Productivity and Social Development in Market Economies, Centre for the Study of Living Standards, 2004.

[11] P. E. Ceruzzi and W. Aspray, A History of Modern Computing, second edition (History of Computing), MIT Press, 2003.

[12] A. M. Turing, "On computable numbers, with an application to the Entscheidungsproblem," Proceedings of the London Mathematical Society, vol. 42, pp. 230-265, 1936-1937.

[13] J. Von Neumann, "First Draft of a Report on the EDVAC," 1945.

[14] C. E. Shannon, "A mathematical theory of communication," The Bell System Technical Journal, vol. 27, no. 3, pp. 379-423, 1948.

[15] G. E. Moore, "Cramming More Components onto Integrated Circuits," Electronics, pp. 114-117, 19 April 1965.

[16] R. H. Dennard, F. H. Gaensslen, H.-N. Yu, V. L. Rideout and A. R. LeBlanc, "Design of ion-implanted MOSFETs with very small physical dimensions," Journal of Solid-State Circuits, vol. 9, no. 5, p. 253, 1974.

[17] T. N. Theis and H. S. Wong, "The End of Moore's Law: A New Beginning for Information Technology," Computing in Science & Engineering, vol. 19, no. 2, pp. 41-50, 2017.

[18] S. H. Fuller and L. I. Millett, "Computing Performance: Game Over or Next Level," Computer, vol. 44, no. 1, pp. 31-38, 2011.

[19] S. Borkar and A. A. Chien, "The Future of Microprocessors," Communications of the ACM, vol. 54, no. 5, pp. 67-77, 1 May 2011.

[20] A. Church, "An unsolvable problem of elementary number theory," American Journal of Mathematics, vol. 58, no. 2, pp. 345-363, 1936.

[21] M. Sipser, Introduction to the Theory of Computation (3rd edition), Cengage Learning, 2012, p. 165–209.

[22] P. Michel, "The Busy Beaver Competition: a historical survey," 2022.

[23] T. Hoefler, T. Häner and M. Troyer, "Disentangling hype from practicality: On realistically achieving quantum advantage," Communications of the ACM, vol. 66, no. 5, pp. 82-87, 2023.

[24] V. Hassija, V. Chamola, A. Goyal, S. S. Kanhere and N. Guizani, "Forthcoming applications of quantum computing: peeking into the future," IET Quantum Communication, vol. 2, no. 1, pp. 35-41, 2020.

[25] M. W. McCall, Classical Mechanics: From Newton to Einstein: A Modern Introduction, John Wiley & Sons.

[26] M. J. Klein, "Max Planck and the beginnings of the quantum theory," Archive for History of Exact Sciences, vol. 1, pp. 459-479, 1961.

[27] S. Klassen, "The photoelectric effect: Reconstructing the story for the physics classroom," Science & Education, no. 20, pp. 719-731, 2011.

[28] M. Le Bellac, Quantum physics, Cambridge University Press, 2011.

[29] A. Sudbery, Quantum mechanics and the particles of nature. An outline for mathematicians., Cambridge University Press, 1986.

[30] Y. Aharonov, E. Cohen, F. Colombo, T. Landsberger, I. Sabadini, D. C. Struppa and J. Tollaksen, "Finally making sense of the double-slit experiment," Proceedings of the National Academy of Sciences, vol. 114, no. 25, pp. 6480-6485, 2017.

[31] R. D. Mattuck, "Ghost-like action-at-a-distance in quantum mechanics: An elementary introduction to the Einstein, Podolsky, Rosen paradox," European Journal of Physics, vol. 3, no. 2, p. 107, 1982.

[32] J. S. Bell, "On the Einstein Podolsky Rosen Paradox," Physics, vol. 1, no. 3, p. 195–200, 4 November 1964.

[33] M. Razavy, "Quantum theory of tunneling," World Scientific, 2013.

[34] J. Hilgevoord and J. Uffink, "The uncertainty principle," Stanford Encyclopedia of Philosophy, 2012.

[35] Z. Ficek and S. Swain, "Quantum interference and coherence: theory and experiments," Springer Science & Business Media, vol. 100, 2005.

[36] M. Born, "Zur Quantenmechanik der Stoßvorgänge," Zeitschrift für Physik, vol. 37, no. 12, pp. 863-867, 1926.

[37] M. A. Fahdil, A. F. Al-Azawi and S. Said, "Operations algorithms on quantum computer," IJCSNS, vol. 10, no. 1, p. 85, 2010.

[38] M. Swathi and B. Rudra, "Implementation of Reversible Logic Gates with Quantum Gates," IEEE 11th Annual Computing and Communication Workshop and Conference, pp. 1557-1563, 2021.

[39] P. A. M. Dirac, The Principles of Quantum Mechanics (International Series of Monographs on Physics), Clarendon Press, 1982.

[40] M. A. Nielsen and I. L. Chuang, Quantum Computation and Quantum Information, Cambridge University Press, 2010.

[41] D. Gottesman, "An introduction to quantum error correction and fault-tolerant quantum computation," In Quantum information science and its contributions to mathematics, Proceedings of Symposia in Applied Mathematics, vol. 68, pp. 13-58, 2010.

[42] Y. Kanamori and S. M. Yoo, "Quantum computing: principles and applications," Journal of International Technology and Information Management, vol. 29, no. 2, pp. 43-71.

[43] "Quantum computing: progress and prospects (Appendix A)," National Academies of Sciences, Engineering, and Medicine.

[44] "IBM online quantum resources | Hello World," IBM, [Online]. Available: https://docs.quantum.ibm.com/guides/hello-world.

[45] V. Bhatia and K. R. Ramkumar, "An Efficient Quantum Computing technique for cracking RSA using Shor's Algorithm," 5th International Conference on Computing Communication and Automation (ICCCA), pp. 89-94, 2020.

[46] P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," Proceedings 35th Annual Symposium on Foundations of Computer Science, pp. 124-134, 1994.

[47] R. Landauer, D. H. Feng and B. L. Hu, "Is quantum mechanically coherent computation useful?," in Proceedings of the Drexel-4 Symposium on Quantum Nonintegrability—Quantum Classical Correspondence, 1994.

[48] D. Beckman, A. N. Chari, S. Devabhaktuni and J. Preskill, "Efficient networks for quantum factoring," Physical Review A, vol. 54, no. 2, p. 1034, 1996.

[49] J. Y. Cai, "Shor's algorithm does not factor large integers in the presence of noise," Science China Information Sciences, vol. 67, no. 6, p. Article 3961, 2024.

[50] J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler and D. Stehlé, "CRYSTALS-Kyber: A CCA-Secure Module-Lattice-Based KEM," IEEE European Symposium on Security and Privacy (EuroS&P), pp. 353-367, 2018.

[51] "Evidence for the utility of quantum computing before fault tolerance," Nature, no. 618, p. 500–505, 2023.

[52] D. Bultrini, M. H. Gordon, P. Czarnik, A. Arrasmith, M. Cerezo, P. J. Coles and L. Cincio, "Unifying and benchmarking state-of-the-art quantum error mitigation techniques," Quantum, vol. 7, p. 1034, June 2023.

[53] A. M. Steane, "Overhead and noise threshold of fault-tolerant quantum error correction," Physical Review A, vol. 68, no. 4, p. 042322, 2003.

[54] Google Quantum AI, & Collaborators, "Quantum error correction below the surface code threshold," Nature, vol. 638, p. 920–926, 9 December 2024.

[55] J. M. Martinis, "Saving superconducting quantum processors from decay and correlated errors generated by gamma and cosmic rays," npj Quantum Information, vol. 7, no. 1, p. 90, 3 June 2021.

[56] H. E. Brandt, "Qubit devices and the issue of quantum decoherence," Progress in Quantum Electronics, vol. 22, no. 5-6, pp. 257-370, 1999.

[57] J. Preskill, "Reliable quantum computers," Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences, vol. 454, no. 1969, pp. 385-410, 1998.

[58] IBM, "100K-qubit supercomputer: The next frontier in quantum computing," IBM Quantum, 3 October 2023. [Online]. Available: https://www.ibm.com/quantum/blog/100k-qubit-supercomputer.

[59] J. Von Neumann, The mathematical foundations of quantum mechanics by Hilbert and von Neumann, Princeton University Press, 1955.

[60] S. Aaronson, "The limits of quantum," Scientific American, vol. 298, no. 3, pp. 62-69, 2008.

[61] E. Gibney, "The quantum gold rush," Nature, vol. 574, no. 7776, pp. 22-24, 2019.

[62] M. G. Raymer and C. Monroe, "The US national quantum initiative," Quantum Science and Technology, vol. 4, no. 2, p. 020504, 2019.

[63] C. Chakraborty, M. Bhattacharya, S. Pal and G. Agoramoorthy, "India's Quantum Move: From Budget Allocation, Action and Future Challenges," Molecular Biotechnology, vol. 66, pp. 3449-3461, 2023.

[64] P. Knight and I. Walmsley, "UK national quantum technology programs," Quantum Science and Technology, vol. 4, no. 4, p. 040502, 2019.

[65] E. B. Kania, "China's quest for quantum advantage—Strategic and defense innovation at a new frontier," In Comparing Defense Innovation Around the World, pp. 47-77, 2022.

[66] E. B. Kania and J. K. Costello, "Quantum technologies, US-China strategic competition, and future dynamics of cyber stability," In 2017 International Conference on Cyber Conflict (CyCon US), pp. 89-96, November 2017.

[67] L. Riesebos, X. Fu, A. A. Moueddenne, L. Lao, S. Varsamopoulos, I. Ashraf, J. van Someren, N. Khammassi, C. G. Almudever and K. Bertels, "Quantum Accelerated Computer Architectures," IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1-4, 2019.

[68] G. S. Ravi, K. N. Smith, P. Gokhale and F. T. Chong, "Quantum computing in the cloud: Analyzing job and machine characteristics," IEEE International Symposium on Workload Characterization (IISWC), pp. 39-50, November 2021.

[69] J. Barzen, F. Leymann, M. Falkenthal, D. Vietz, B. Weder and K. Wild, "Relevance of near-term quantum computing in the cloud: A humanities perspective," International Conference on Cloud Computing and Services Science, pp. 25-58, May 2020.

[70] M. Golec, E. S. Hatay, M. Golec, M. Uyar, M. Golec and S. S. Gill, "Quantum cloud computing: Trends and challenges," Journal of Economy and Technology, vol. 2, pp. 190-199, 2024.