From Bits to Qubits: Comparative Insights into Classical and Quantum Computing Systems

Tariq Jamil
Department of Electrical and Computer Engineering, Sultan Qaboos University, AlKhod 123, Muscat, OMAN

***Abstract*—The rapid development of computing hardware has been driven by an ever-emerging need for high throughput, scalable performance, and computation capabilities to be able to address increasingly complex problems. The paradigm of classical computing, centered on deterministic binary logic and the von Neumann architecture, has long favored modern information processing and still supports a wide range of applications. However, physical limits in scaling transistors, power dissipation, and slowing down Moore's Law have stimulated the consideration of alternative computing paradigms. Quantum computing is now an emerging means that exploits the basic principles of quantum mechanics-such as superposition, entanglement, and quantum interference-to enable new forms of computation. This review study discusses a comparison between the classical and quantum computing systems in operational paradigms, architecture structures, performance characteristics, and application domains. This work is supported by a systematic review of the established theories, currently realized hardware implementations, and representative algorithms. The analysis underlines that classical systems remain very reliable, scalable, and efficient in general-purpose and deterministic workloads, while quantum systems ensure essential advantages in specified problem classes, such as cryptography, quantum chemistry, combinatorial optimization, and selected machine learning tasks. The study concludes that classical and quantum computing are best viewed as complementary technologies. Future high-performance computing platforms will most likely be based on hybrid-classical–quantum architectures in which quantum processors serve as specialized accelerators to help classical systems solve new computational challenges.**

Keywords*—*Classical computing; high-performance computing; quantum computing; qubit

I. Introduction

The pace with which computer hardware has advanced in the past decades has revolutionized science, business, and society. In the midst of all this change is classical computing, a deterministic binary logic model where the basic unit of information—the bit—can be in one of just two states: 0 or 1. Classical computer architecture, which had previously been founded on the von Neumann model, relies on central processing units (CPUs), memory components, and input/output subsystems for executing instructions sequentially [1]. This paradigm has been in a position to maintain a wide scope of applications, ranging from personal computers and business systems to embedded controllers and high-performance computing platforms [2].

Despite significant advances in conventional hardware, largely through transistor scaling in line with Moore's Law, there are inherent bounds that have increasingly gained prominence as computational demands continue to rise. The contemporary challenges, such as simulating computationally intensive quantum systems, solving large combinatorial optimization problems, and performing secure cryptographic analysis, are frequently found to be intractable within conventional systems due to their sequential processing nature and physical miniaturization constraints [2]. Furthermore, the observed slowdown in Moore's Law, combined with increasing power density and thermal dissipation issues, has also heightened the demand for novel paradigms of computation capable of overcoming these computational challenges [3].

Quantum computing is one of those revolutionary paradigms that relies on the counterintuitive principles of quantum mechanics—namely, superposition, entanglement, and quantum interference—to process information in fundamentally new ways. Unlike classical bits, the fundamental unit of quantum information, the qubit, can exist in a superposition of 0 and 1 simultaneously, enabling exponentially parallel computation for certain classes of problems [4]. Moreover, the phenomenon of entanglement allows multiple qubits to become related in such a way that one qubit's state will immediately influence the state of another, regardless of the distance between them, allowing for faster-than-classical information correlation and algorithmic speedup [5]. Several physical implementations of qubits have been proposed and experimentally realized, including superconducting circuits, trapped ions, and photonics, with different strengths and specific engineering challenges [6].

This review article features an orderly comparative assessment of classical hardware computing and quantum hardware computing. It examines their fundamental operating paradigms, structural architectures, performance profiles in computation, and their corresponding domains of application. In consideration of detailed deliberation on each paradigm's advantages, limitations, and technological maturity, this study seeks to put their synergistic roles in addressing existing and anticipated computational problems into perspective. Besides, the review focuses on ongoing research efforts and potential integration strategies of hybrid classical-quantum systems that are predicted to be among the next-generation high-performance computing solutions' primary drivers.

The study is organized as follows: Section II presents a comparative analysis of classical computing and quantum computing with reference to their operational paradigms, structural architectures, computational performance characteristics, and application domains. Section III outlines the challenges and expected future trends in computing, followed by a summary and conclusion in Section IV.

II. Comparative Analysis

A. Operational Paradigms

Traditional computer systems operate on deterministic binary logic such that the fundamental unit of information—the bit—can be one out of two possible states, 0 or 1, at any given moment [2]. The operation model of conventional computing is based on the von Neumann architecture, in which a CPU reads, executes, and applies instructions in RAM and transfers data between I/O devices via a shared system bus [1]. Logical functions are performed through the development of electronic switches known as transistors into logic gates (e.g., AND, OR, and NOT gates) that process in binary states deterministically under the rules of Boolean algebra. Instruction execution one after the other is maintained through control units and registers, which regulate program flow as well as data movement between memory and processing elements. Fig. 1 shows a conceptual block diagram of a typical von Neumann architecture, its key hardware components, and how they are all connected.

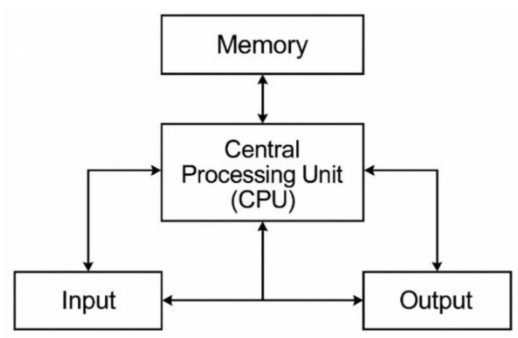

Fig. 1. Block diagram of classical computing architecture showing CPU, memory, I/O, and bus.

Quantum computing, by contrast, relies on the principles of quantum mechanics, wherein the fundamental unit of information—the quantum bit, or qubit—can exist in a superposition state that represents both 0 and 1 simultaneously until observed [4]. This aspect tremendously expands the state space of a quantum system. For example, while a classical n-bit system can represent one of 2^n possible states at any given time, an n-qubit quantum system can represent all 2^n possible states at the same time through superposition. Also, qubits can be entangled, a quantum phenomenon whereby the state of a qubit becomes intrinsically linked with the state of another, regardless of the spatial separation between them, enabling forms of parallelism and information correlation without a classical analogue [5].

Quantum computations are carried out by sequences of reversible, unitary operations known as quantum gates. Unlike classical logic gates, which act deterministically on bits, quantum gates operate on the probability amplitudes of qubit states using linear transformations described by unitary matrices [6]. Standard quantum gates include the Hadamard gate (H), which places qubits in superposition; the Pauli-X gate, which flips qubit states in a manner analogous to a classical NOT gate; and the controlled-NOT (CNOT) gate, which entangles two qubits by conditionally flipping the target qubit based on the state of a control qubit.

The differences in operation between quantum and classical computing systems are profound. The classical systems are founded on sequential, deterministic execution models, with bit operations tracking predictable Boolean logic rules [2]. Their error correction is highly matured, employing redundancy along with error-detecting codes to provide reliable, fault-tolerant computation. On the other hand, quantum systems are probabilistic by nature, and only the result is revealed when measured that collapses the qubit superposition to a definite classical state [4]. Quantum error correction is far more challenging due to the continuous nature of quantum states and their susceptibility to decoherence through interactions with the environment. Further, while classical logic operations in general are not reversible, quantum operations must be reversible because unitary operations preserve the total information content of the system [6]. This fundamental difference in operating principles illustrates both the revolutionary potential and the profound technical challenges of quantum computing relative to conventional classical hardware systems. Table I compares the operational principles of classical computing and quantum computing.

TABLE I COMPARATIVE SUMMARY OF OPERATIONAL PRINCIPLES

Feature	Classical Computing	Quantum Computing
Information Unit	Bit (0 or 1)	Qubit (0 or 1 or both)
Operational Basis	Deterministic, Boolean logic operations	Probabilistic, governed by superposition and entanglement
Execution Mode	Sequential	Massively parallel via quantum superposition
Gate Types	Irreversible logic gates (AND, OR, NOT)	Reversible quantum gates (Hadamard, CNOT, Pauli-X, etc.)
Measurement	Direct, non-destructive	Collapses quantum state, probabilistic outcome
Error Correction	Mature, based on redundancy and parity checking	Complex, requires quantum error correction codes (e.g., Shor, Surface codes)

B. Structural Architectures

The conventional computer systems are primarily von Neumann-based, designed by John von Neumann in the mid-20th century. The fundamental organization has a central processing unit (CPU), memory unit, input/output devices, and a system bus to enable them to communicate with each other (see Fig. 1). Inside the CPU, it has an Arithmetic Logic Unit (ALU) and a Control Unit (CU). Instructions and data are pulled from memory, processed by the CPU, and the results are written back into memory or sent out through the I/O interfaces [1]. Data's binary structure and serial processing paradigm, where the information is received in either 0s or 1s, make conventional computers highly deterministic and predictable in a vast number of applications. However, these systems are still restrained by physical constraints such as transistor miniaturization, power dissipation, and thermal dissipation [3].

Quantum computing architectures, on the other hand, draw on the basic principles of quantum mechanics, such as superposition, entanglement, and quantum interference. The simplest unit of quantum information—the qubit—has the possibility of existing in a superposition of states $|0\rangle$ and $|1\rangle$ at the same time [4]. Quantum computers are often made up of several qubits in a lattice or network of qubits, manipulated using sequences of quantum gates. Unlike classical systems, which are designed around a centralized CPU and sequential instruction execution, quantum processors are founded upon a quantum circuit model wherein computations are performed via carefully orchestrated quantum gate operations. These architectures also demand highly specialized hardware, including microwave pulse generators, cryogenic dilution refrigerators for stabilization at near absolute zero temperatures, and sophisticated quantum error correction systems [7].

Architecturally, the computing mechanisms and data flows in quantum and classical systems are very different. In classical systems, information moves between memory, CPU, and I/O subsystems over system buses, as directed by a control unit. Performance bottlenecks tend to occur in the memory hierarchy and instruction execution pipeline, most notably the von Neumann bottleneck, due to limited data transfer between memory and CPU [8]. Conversely, quantum computers take advantage of the inherent parallelism of quantum states, where an n-qubit system can simultaneously represent 2^n possible configurations. This possibility requires meticulous control of qubit coherence and entanglement, typically maintained at millikelvin temperatures in cryogenic setups to mitigate decoherence and environmental noise [7].

To better visualize quantum states, Fig. 2 illustrates a qubit state on the Bloch sphere, demonstrating how quantum states exist in a continuous, multi-dimensional space, as opposed to discrete binary states of classical hardware. While traditional computing has grown into massively scalable, commercially optimized systems, quantum architectures remain in the making, hindered by challenges of qubit fidelity, error rates, and scalability of systems. Yet, quantum architecture flexibility and computational power bring drastic improvements in fields such as cryptography, combinatorial optimization, and complex quantum system simulation [6]. Table II compares the architectural designs of classical computing with quantum computing.

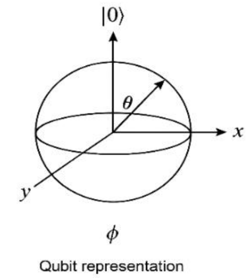

Fig. 2. Qubit representation on the Bloch sphere.

TABLE II COMPARATIVE SUMMARY OF ARCHITECTURAL DESIGNS

Component	Classical Architecture	Quantum Architecture
Processing Unit	CPU (ALU + Control Unit)	No central processor; qubits manipulated by quantum gates
Memory Unit	RAM, Cache, Secondary Storage	Quantum memory (qubit arrays, limited capacity)
Data Bus	Parallel/serial data lines	Quantum interconnects or qubit couplers
Execution Model	Von Neumann sequential model	Quantum circuit model
Operating Environment	Room temperature, conventional hardware	Cryogenic (mK) temperatures for coherence control
Scalability	Highly scalable	Currently limited by decoherence, qubit fidelity

C. *Computational Performance Characteristics*

The performance in conventional computing systems is generally most accurately measured in terms of clock rate (measured in gigahertz), instruction throughput (instructions per cycle, IPC), and parallel processing capability. Contemporary conventional processors also enhance performance using design techniques like multi-core designs, instruction pipelining, simultaneous multi-threading (hyper-threading), and the utilization of hardware accelerators to perform some operations [2]. High-performance computing (HPC) systems, such as the Frontier and Fugaku supercomputers, utilize very large parallel clusters consisting of thousands of central processing units (CPUs) and graphics processing units (GPUs), providing sustained computing performance at the exaFLOPS rate, or 10^{18} single-precision floating-point computations per second [9]. Classical hardware remains very beneficial for a vast assortment of deterministic, sequential, and general-purpose computational tasks, such as numerical simulation, data analysis, and machine learning prediction.

In contrast, the performance of quantum computer hardware is measured by a different set of values, including the number of qubits, coherence time, gate fidelity, and quantum volume—a multi-dimensional figure of merit devised by IBM that aggregates a system's computational capability in terms of qubit quality, connectivity, and circuit depth [10]. Whereas classical systems tend to exhibit linear or polynomial computational complexity scaling, certain quantum algorithms achieve exponential or quadratic speedup over their best-known classical analogues. These include Shor's algorithm for number factorization and Grover's algorithm for searching an unstructured database, both of which employ quantum mechanical features like superposition and entanglement to explore different computational paths in parallel [11].

One of the key differences between classical systems and quantum systems is in fault tolerance and error control. Classical computing systems have had decades of practice in error detection techniques and correcting mechanisms, with facilities such as parity bits, checksums, and error-correcting code (ECC) memory being used to ensure operational reliability [2]. Quantum systems are inherently prone to decoherence, quantum noise, and gate errors as a result of environmental coupling and hardware defects [12]. As such, existing Noisy Intermediate-Scale Quantum (NISQ) devices are only limited in how far the operation circuit depth they can achieve before inaccuracies of computation begin. More advanced quantum error correction codes, such as Shor's code and surface codes, are in active development but demand large numbers of extra ancillary qubits, which pose severe scalability challenges [12].

Moreover, the regions where classical and quantum computers excel are highly distinct. Classical computers continue to excel in general-purpose computing, transaction processing, and engineering simulations. Quantum computers, however, are bound to excel in niche applications such as quantum chemistry, cryptography, combinatorial optimization, and certain machine learning tasks [6]. Experimental demonstrations of quantum advantage have been provided, such as Google's Sycamore processor, which executed a specific computation problem significantly faster than the fastest-performing classical supercomputers of its time [7]. Nevertheless, demonstration of the widespread, fault-tolerant quantum computing devices to outperform classical hardware consistently on a large majority of problem types remains an ultimate milestone.

D. Application Domains

Classical computing devices have evolved to be a necessity in almost all aspects of contemporary society. From enterprise functions, web applications, and healthcare management to space flight simulations, artificial intelligence (AI), and complex engineering calculations, classical systems provide solid, scalable, and well-established solutions for deterministic and probabilistic computational operations [2]. Supercomputers and high-performance computing (HPC) clusters have significantly impacted fields such as weather forecasting, genome sequencing, and financial modeling. Traditional systems enjoy flexibility through their entrenched hardware and software ecosystems, including standard programming languages, operating systems, and networking protocols [13]. Furthermore, edge computing and cloud infrastructure advancements have enabled classical computing influence expansion into mobile devices, systems of the Internet of Things, distributed services, and made it the technological basis of modern digital infrastructure.

By contrast, quantum computing is directed towards tackling problem classes that are intrinsically difficult or impossible for classical devices due to exponential time or space complexity. A prime example application field is quantum chemistry, wherein the simulation of molecular interactions and reaction mechanisms necessitates computational resources scaling exponentially with system size on classical computers [14]. Quantum computers, by exploiting phenomena such as superposition and entanglement, can more directly simulate complex quantum systems, enabling simulations of large molecules, catalysts, and exotic materials at scales previously unviable. This could revolutionize fields such as pharmaceuticals, materials science, and the creation of sustainable energy.

Optimization problems are among the other domains where quantum systems are expected to be more powerful compared to their classical versions. Logistics routing, portfolio optimization, and scheduling problems, classically reliant on approximating or heuristically-based algorithms, are being tried for quantum supremacy with such tools as the Quantum Approximate Optimization Algorithm (QAOA) [15].

Quantum cryptography is also another widely reported application of quantum computing. Classical cryptosystems like RSA and elliptic-curve cryptography (ECC) rely on the computational difficulty of factoring numbers and calculating discrete logarithms—problems that are secure against conventional attacks but susceptible to powerful quantum attacks via Shor's algorithm [11]. The advent of practical quantum technology thus necessitates post-quantum cryptography to secure digital communication infrastructures.

In addition, quantum computing promises to help solve problems in the area of machine learning. QSVM and QPCA are examples of algorithms that even use quantum parallelism to help with data processing and feature extraction among high-dimensional features. This could generate acceleration for more complex AI workloads [16].

While classical systems still rule general-purpose, real-time, and transactional computing, quantum hardware is being developed for problems of high complexity and resource requirements for which classical resources scale unrealistically. The future computing paradigm will most probably be hybrid, with quantum accelerators alongside classical cores in heterogeneous architectures [6]. While quantum computing is now in its Noisy Intermediate-Scale Quantum (NISQ) era, ongoing research and industrial investment point to a path where such systems will become central in niche areas as classical computing remains the basis for standard, mission-critical, and real-time applications.

E. Discussion

It is clear from the above review that classical computing is very efficient, reliable, and scalable with deterministic general-purpose computing, whereas quantum computing behaves in a vastly different manner due to super-positioning and entanglement, providing a substantial advantage in specific problem domains like cryptography, quantum chemistry, and combinatorial optimization, even with the existing hindrances in using quantum computing due to decoherence, qubit accuracy, and scalability. Both approaches to computing have properties which make one no better than the other, rather they exist in a complementary form. The future of computing lies in developing a hybrid classical-quantum system, efficient quantum algorithms, gainful efforts on improved error correction techniques, as well as advanced research on qubit technologies. Future research may also entail developing a fair system of benchmarking both classical and quantum computing on a similar problem set, thus developing next-generation high-performance computing environments incorporating quantum accelerators.

III. Challenges and Future Trends

Despite phenomenal technology advancements, conventional computing is facing inherent constraints as it approaches the physical limits of Moore's law, which historically predicted transistor density doubling approximately every two years [3]. Once the transistor sizes decrease below 3 nanometers, serious issues such as quantum tunneling, increasing thermal dissipation, and power leakage significantly decelerate further miniaturization and limit clock speed increases. Furthermore, the rising complexity of the system and increased energy requirements of massive data centers and high-performance computing (HPC) clusters require sustained innovation in such areas as energy-efficient architectures, three-dimensional stacking of chips, and heterogeneous systems comprising central processing units (CPUs), graphics processing units (GPUs), and field-programmable gate arrays (FPGAs) [17]. Even as classical computing must remain the pre-eminent computation paradigm in general-purpose programs, future trends hold promise of a shift toward hybrid classical-quantum computing systems, special-purpose hardware accelerators optimized to process artificial intelligence (AI), edge computing, and domain-specific tasks to optimize both performance and power efficiency [18].

IV. Summary and Conclusion

In summary, this review has presented a detailed comparative survey of classical and quantum computing systems in terms of their working principles, architectural frameworks, computational performance characteristics, and application domains. While classical computing remains the cornerstone of modern digital infrastructure due to its scalability, reliability, and mature ecosystems, it faces physical and architectural limitations as the miniaturization of transistors reaches theoretical limits. Quantum computing, founded on the principles of superposition, entanglement, and quantum interference, has great potential for addressing intractable problems in cryptography, quantum chemistry, optimization, and artificial intelligence. However, current quantum systems are constrained by decoherence problems, qubit fidelity, and lack of scalability. Looking ahead, the future of computation will be headed in the direction of heterogeneous architectures where classical systems continue to be at the core of general-purpose and mission-critical applications, but quantum accelerators will enhance computational prowess for specialized high-complexity problems. This continued collaboration of the classical and quantum paradigms is the way forward for high-performance computing, representing a fundamental shift in addressing the expanding demands of the data-driven, AI-augmented era.

References

[1] J. von Neumann, "First draft of a report on the EDVAC," Moore School of Electrical Engineering, University of Pennsylvania, 1945. Published in IEEE Annals of the History of Computing, vol. 15, no. 4, pp. 27–43, 1993.

[2] D. A. Patterson and J. L. Hennessy, Computer Organization and Design: The Hardware/Software Interface, 5th ed. Morgan Kaufmann, 2014.

[3] G. E. Moore, "Cramming more components onto integrated circuits," Electronics, vol. 38, no. 8, pp. 114–117, 1965.

[4] M. A. Nielsen and I. L. Chuang, Quantum Computation and Quantum Information, 10th ed. Cambridge University Press, 2010.

[5] R. Horodecki, P. Horodecki, M. Horodecki, and K. Horodecki, "Quantum entanglement," Rev. Mod. Phys., vol. 81, no. 2, pp. 865–942, 2009, doi: 10.1103/RevModPhys.81.865.

[6] T. Proctor, K. Young, A. D. Baczewski, and R. Blume-Kohout, "Benchmarking quantum computers," Nat. Rev. Phys., vol. 7, pp. 105–118, 2025, doi: 10.1038/s42254-024-00796-z.

[7] Q. A. Memon, M. Al Ahmad, and M. Pecht, "Quantum computing: Navigating the future of computation, challenges, and technological breakthroughs," Quantum Rep., vol. 6, no. 4, pp. 627–663, 2024, doi: 10.3390/quantum6040039.

[8] W. Stallings, Computer Organization and Architecture: Designing for Performance, 10th ed. Pearson, 2015.

[9] J. Dongarra et al., "The Top500 list and the future of supercomputing," Comput. Sci. Eng., vol. 23, no. 4, pp. 24–32, 2021, doi: 10.1109/MCSE.2021.3072860.

[10] E. Giusto et al., "Dependable classical–quantum computing systems engineering," Front. Comput. Sci., vol. 7, p. 1520903, 2025, doi: 10.3389/fcomp.2025.1520903.

[11] P. W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in Proc. 35th Annu. Symp. Found. Comput. Sci., 1994, pp. 124–134, doi: 10.1109/SFCS.1994.365700.

[12] D. Gottesman, "An introduction to quantum error correction and fault-tolerant quantum computation," in Proc. Symp. Appl. Math., vol. 68, pp. 13–58, 2010, doi: 10.1090/psapm/068.

[13] W. Stallings, Operating Systems: Internals and Design Principles, 9th ed. Pearson, 2018.

[14] A. Aspuru-Guzik, A. D. Dutoi, P. J. Love, and M. Head-Gordon, "Simulated quantum computation of molecular energies," Science, vol. 309, no. 5741, pp. 1704–1707, 2005, doi: 10.1126/science.1113479.

[15] V. De Maio, I. Brandic, E. Deelman, and J. Cito, "The road to hybrid quantum programs: Characterizing the evolution from classical to hybrid quantum software," arXiv, Mar. 2025, arXiv:2503.11450.

[16] R. Ballester, J. Cerquides, and L. Artiles, "Quantum federated learning: A comprehensive literature review of foundations, challenges, and future directions," Quantum Mach. Intell., vol. 7, p. 73, 2025, doi: 10.1007/s42484-025-00292-2.

[17] M. D. Hill and M. R. Marty, "Amdahl's law in the multicore era," Computer, vol. 41, no. 7, pp. 33–38, 2008, doi: 10.1109/MC.2008.209.

[18] N. P. Jouppi et al., "In-datacenter performance analysis of a Tensor Processing Unit," in Proc. 44th Annu. Int. Symp. Comput. Archit. (ISCA), 2017, pp. 1–12, doi: 10.1145/3079856.3080246.