WHITE PAPER • 2026

Letting Physics Do the Computation

Understanding the Role of Quantum Computers in Modern Computing

Viren Patel

40A2D1B0C6A984378D38C1C2816F5CB59A4BA19970B32A11EED3C79FE5646866

Abstract

Quantum computing is often portrayed as a revolutionary technology capable of replacing **classical computers** across a wide range of computational tasks. While **quantum processors** do offer advantages for certain problem classes, this narrative can obscure a more fundamental insight: computation is deeply tied to the **physical systems** used to implement it.

This paper examines computation from a **physical perspective** and argues that different computational architectures are naturally suited to different types of problems. Through illustrative examples—including a mechanical sorting simulation and a visualization of **Grover's quantum search** algorithm—we demonstrate how some problems can be solved more naturally when the computational process is embedded directly in the physics of the system.

Table of Contents

40A2D1B0C6A984378D38C1C2816F5CB59A4BA19970B32A11EED3C79FE5646866

1. Introduction

The history of computing is inseparable from the **physical systems** used to implement it. Mechanical calculators once performed arithmetic through gears and levers, analog computers modeled differential equations through electrical circuits, and digital computers revolutionized computation through binary logic and programmable architectures.

Each of these systems excelled not because it was universally superior, but because its physical structure aligned naturally with the problems it was designed to solve.

In recent years, **quantum computing** has emerged as a new computational paradigm based on the principles of **quantum mechanics**. Popular discussions often describe **quantum computers** as dramatically faster replacements for **classical machines**. While **quantum processors** can provide advantages for certain tasks, the idea that they will broadly replace **classical computers** is misleading.

Quantum computers are not general-purpose accelerators for all forms of computation. Instead, they represent specialized systems that exploit **quantum mechanical** behavior to efficiently represent and evolve certain classes of problems.

> **KEY INSIGHT**
>
> Quantum computers are not general-purpose accelerators. They are specialized systems that exploit quantum mechanical behavior for specific problem classes.

40A2D1B0C6A984378D38C1C2816F5CB59A4BA19970B32A11EED3C79FE5646866

2. Computation as a Physical Process

Most discussions of algorithms focus on symbolic manipulation and logical operations performed by digital machines. However, computation can also emerge directly from the dynamics of **physical systems**.

Consider the task of sorting objects by size. In a **classical computer**, sorting requires algorithms that repeatedly compare elements and rearrange their positions in memory. Even highly optimized algorithms require a sequence of discrete comparisons and data movements.

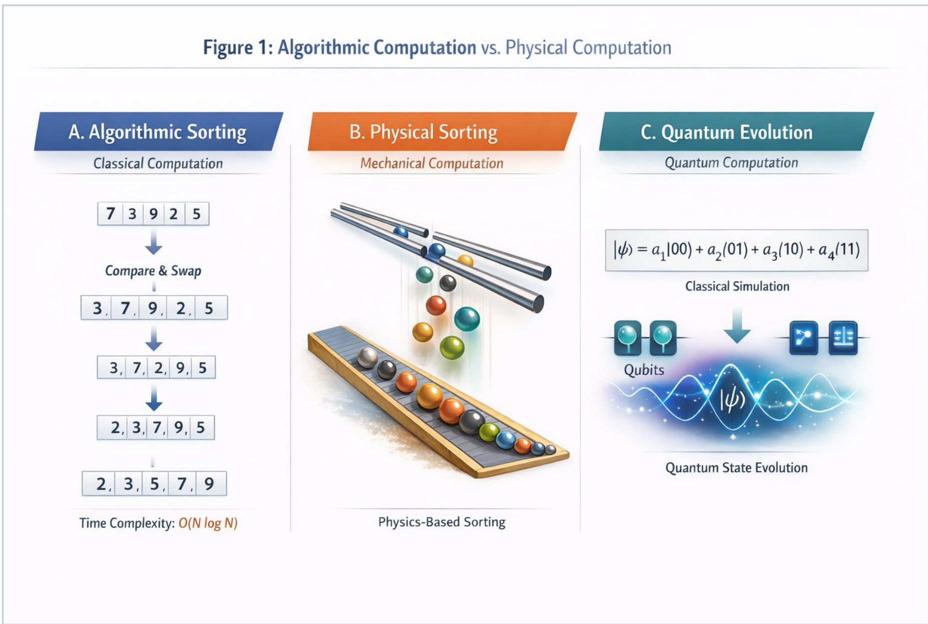

Figure 1: Algorithmic Computation vs. Physical Computation

40A2D1B0C6A984378D38C1C2816F5CB59A4BA19970B32A11EED3C79FE5646866

When the gap becomes slightly larger than the diameter of a marble, that marble falls through the rods and rolls onto a sloped collection tray below. Because all marbles are loaded simultaneously, the smallest marbles fall first, followed by progressively larger ones as the separation increases. The ordering emerges directly from the geometry and motion of the system.

PHYSICAL COMPUTATION

The physics of the system performs the computation — no algorithmic comparisons or measurements are required. Sorting emerges from geometry.

INTERACTIVE DEMONSTRATION

The physical sorting mechanism described above can be explored through an interactive simulation. It models marbles of varying diameters placed between two parallel rods that gradually separate. As the gap increases, smaller marbles fall first, followed by progressively larger ones — demonstrating how computation can arise from physical dynamics rather than symbolic operations.

Interactive simulation: **Physical Sorting Demonstration**

40A2D1B0C6A984378D38C1C2816F5CB59A4BA19970B32A11EED3C79FE5646866

3. The Landscape of Computing Machines

Modern computing systems consist of a diverse **ecosystem** of architectures optimized for different classes of problems. General-purpose **CPUs** are designed to handle sequential logic and control tasks. Graphics processing units (**GPUs**) accelerate highly parallel numerical computations, particularly those used in scientific computing and machine learning.

Specialized processors such as tensor processing units (**TPUs**) have been developed to efficiently execute neural network workloads. **Quantum computers** represent another point in this landscape. Instead of manipulating **classical bits** through deterministic logic operations, **quantum processors** operate on **qubits** that evolve according to the principles of **quantum mechanics**.

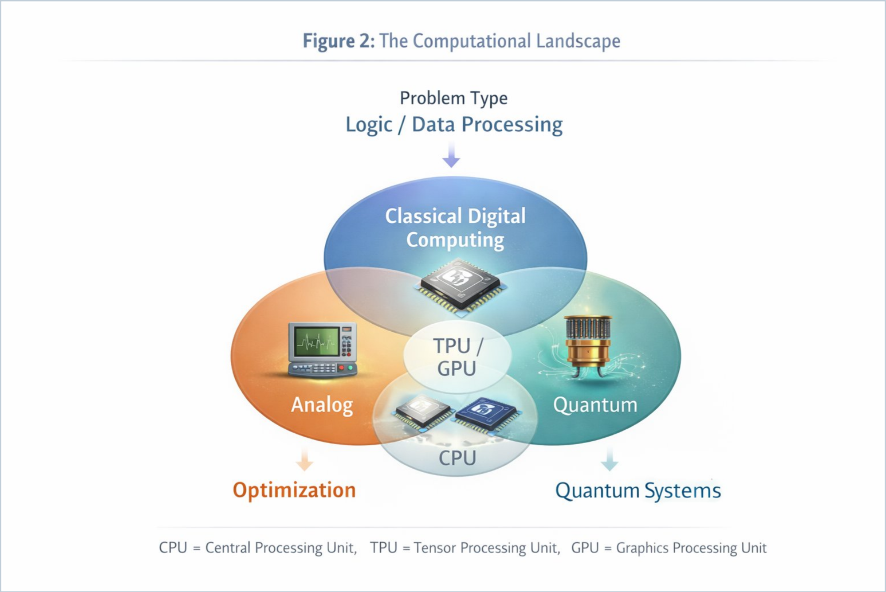

Figure 2: The Computational Landscape

40A2D1B0C6A984378D38C1C2816F5CB59A4BA19970B32A11EED3C79FE5646866

4. The Exponential Wall of Classical Simulation

One of the most challenging problems in **classical computing** is the simulation of **quantum systems**. The state of a quantum system is described by a mathematical object known as the **wavefunction**, which contains the **amplitudes** of all possible configurations of the system.

For a system composed of n **quantum bits**, the complete **quantum state** contains 2^n complex **amplitudes**. A **classical computer** attempting to simulate such a system must explicitly store and manipulate this **exponentially** growing state vector.

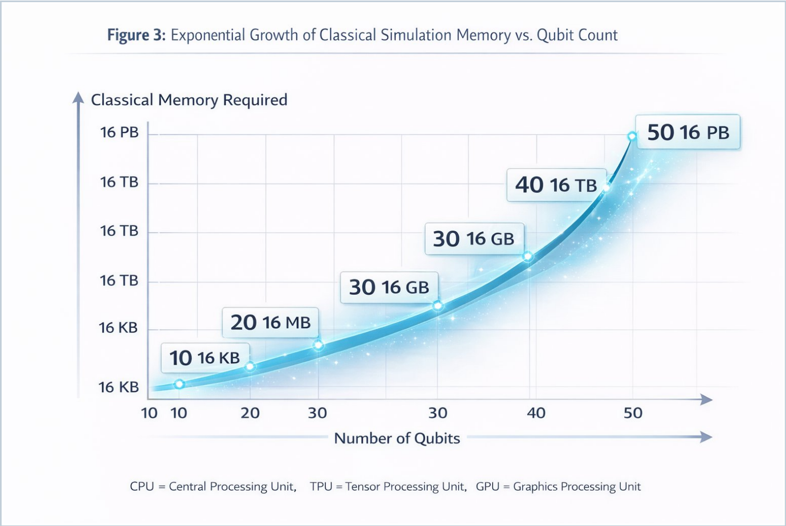

Figure 3: Exponential Growth of Classical Simulation Memory vs. Qubit Count

EXPONENTIAL SCALING

10 qubits: 16 KB | 20 qubits: 16 MB | 30 qubits: 16 GB
40 qubits: 16 TB | 50 qubits: 16 PB — beyond any classical machine

40A2D1B0C6A984378D38C1C2816F5CB59A4BA19970B32A11EED3C79FE5646866

5. Quantum Representation of Information

Quantum computers approach this challenge from a fundamentally different perspective. Instead of storing the full **wavefunction** explicitly in **classical memory**, **quantum processors** encode the state directly within the physical state of **qubits**.

A register of n **qubits** naturally represents a **quantum state** containing 2^n **amplitudes**. Quantum operations manipulate these **qubits** according to the principles of **quantum mechanics**, allowing the system to evolve as a physical **quantum process** rather than as a numerical simulation.

Figure 4: Classical vs. Quantum Representation of States

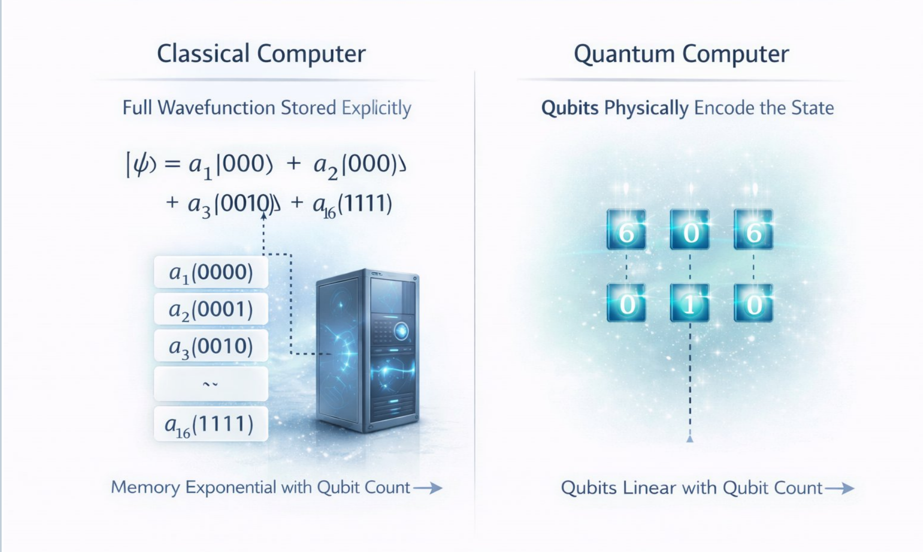

Figure 4: Classical vs. Quantum Representation of States

CLASSICAL VS. QUANTUM

Classical: Memory grows exponentially with qubit count (2^n amplitudes stored).

Quantum: Physical qubits scale linearly — n qubits encode n-qubit states.

40A2D1B0C6A984378D38C1C2816F5CB59A4BA19970B32A11EED3C79FE5646866

6. Quantum Search and Amplitude Amplification

While **quantum computers** do not accelerate every computational task, they can provide advantages for certain types of problems. One example is **Grover's search algorithm**, which enables searching an unsorted database of N elements in approximately √N steps rather than the N steps required by **classical algorithms**.

The key mechanism behind this speedup is **amplitude amplification**. Rather than sequentially checking each candidate solution, **Grover's algorithm** prepares a **quantum superposition** of all possible states and iteratively amplifies the probability **amplitude** of the correct solution through **constructive interference**.

GROVER'S SPEEDUP

Classical search: O(N) steps | Quantum search: O(√N) steps
For N = 1,000,000: Classical needs ~1M checks, Quantum needs ~1,000

■■ INTERACTIVE DEMONSTRATION

To illustrate the dynamics of amplitude amplification, an interactive visualization of Grover's algorithm has been developed. The simulation allows readers to observe how a uniform superposition is prepared, an oracle marks the correct state through phase inversion, and the diffusion operator amplifies the target's probability amplitude. Through successive iterations, constructive interference increases the likelihood of measuring the correct result.

Interactive simulation: **Grover's Algorithm Visualization**

40A2D1B0C6A984378D38C1C2816F5CB59A4BA19970B32A11EED3C79FE5646866

6.1. Quantum Oracles and the Role of Classical Functions

A common misconception about **Grover's algorithm** is that it somehow evaluates a classical function across all possible inputs simultaneously. In reality, a purely classical black box cannot operate on a **quantum superposition**. Interacting with a classical system would collapse the **quantum state**, destroying the **superposition** that makes the algorithm work.

$$U_f |x\rangle = (-1)^{f(x)} |x\rangle$$

where f(x) equals 1 for the target state and 0 for all others. This phase flip does not produce a directly observable change, but it sets up the conditions for **constructive interference** in subsequent steps of the algorithm. Figure 5 illustrates the distinction between a classical black-box function and a quantum **oracle**.

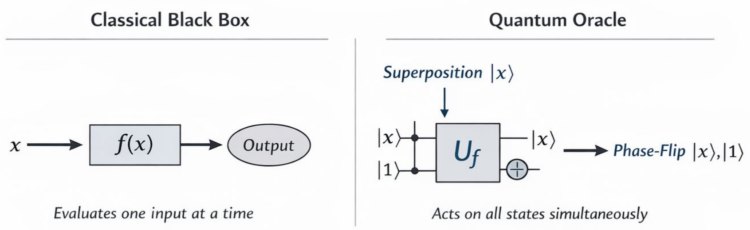

Figure 5: *Comparison between a classical black-box function and a quantum oracle. A classical function evaluates one input at a time, while a quantum oracle acts linearly on a superposition of states and marks the correct solution by flipping its phase.*

Figure 5: Comparison between a classical black-box function and a quantum oracle.

40A2D1B0C6A984378D38C1C2816F5CB59A4BA19970B32A11EED3C79FE5646866

Why the Oracle Must Be Reversible

All quantum operations must be reversible, meaning no information can be destroyed during the computation. A classical function that simply outputs 0 or 1 is inherently irreversible because the input cannot be recovered from the output alone.

The Real Cost of Grover's Algorithm

Grover's algorithm provides a quadratic reduction in the number of **oracle** evaluations required to find a solution: $O(\sqrt{N})$ compared to $O(N)$ for classical search. However, the total runtime is not determined solely by the number of **oracle** calls.

The actual computational cost becomes $O(\sqrt{N} \times C)$, where C is the cost of executing the **oracle** circuit once. In many practical applications, constructing an efficient **oracle** is itself a significant engineering challenge. For problems where the function being evaluated involves complex logic, the **oracle** circuit may require many quantum gates, increasing the overall resource requirements.

Classical search: $O(N)$ oracle evaluations
Grover search: $O(\sqrt{N})$ oracle evaluations
Actual runtime: $O(\sqrt{N} \times$ cost of oracle circuit$)$

KEY INSIGHT

Grover's algorithm does not 'search all possibilities simultaneously' in the classical sense. Instead, it manipulates the quantum wavefunction so that interference gradually concentrates probability on the correct answer.

This reflects a broader theme in quantum computing: rather than computing answers directly, algorithms shape the physical evolution of the system so that the correct result becomes the most likely measurement outcome.

40A2D1B0C6A984378D38C1C2816F5CB59A4BA19970B32A11EED3C79FE5646866

6.2. Geometric Interpretation of Grover's Algorithm

The dynamics of **Grover's algorithm** can be understood through an elegant geometric interpretation. The entire computation takes place within a two-dimensional subspace spanned by two orthogonal vectors: the target state and the uniform **superposition** of all non-target states.

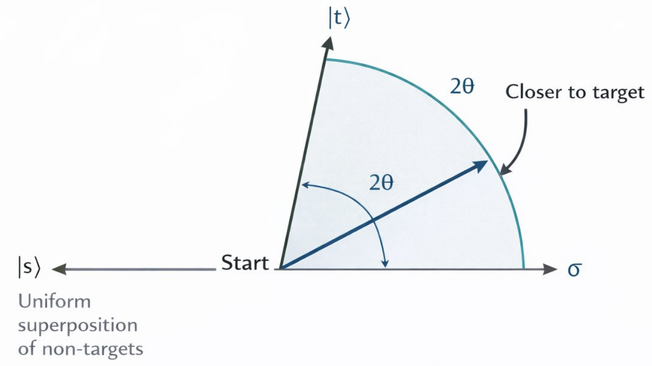

Figure 7: *Geometric interpeetation of Grover's algorithm. The quantum state evolves in a two-dimensional space spanned by the target state and the uniform superposition of non-target states. Each iteration rotates the state vector closer to the correct solution.*

Figure 7: Geometric interpretation of Grover's algorithm. Each iteration rotates the state vector closer to the target.

> **GEOMETRIC INSIGHT**
>
> The oracle reflects the state about the non-target subspace (phase flip).
> The diffusion operator reflects about the mean amplitude.
> Together, these two reflections produce a rotation toward the target state.

40A2D1B0C6A984378D38C1C2816F5CB59A4BA19970B32A11EED3C79FE5646866

Another way to visualize **Grover's algorithm** is through an energy landscape analogy. The probability distribution across all computational states can be imagined as a surface, where the correct solution sits at the bottom of a funnel-shaped basin.

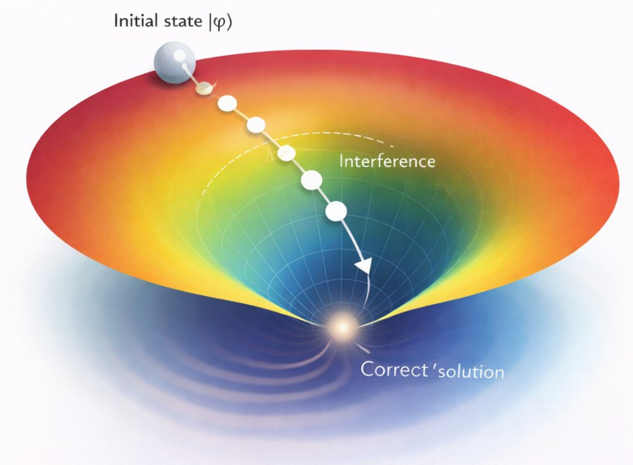

Figure 9: *Grover's algorithm can be visualized as manipulating a probabiity distribution, shaped like an energy landscape. Through controlled interference, the probability of measuring the correct solution increases. This visualization parallels the marble-sorting analogy*

Figure 8: Grover's algorithm visualized as an energy landscape where interference guides probability toward the solution.

40A2D1B0C6A984378D38C1C2816F5CB59A4BA19970B32A11EED3C79FE5646866

7. Interactive Simulations and Visualization

To develop physical intuition for the mechanisms behind **Grover's algorithm**, an interactive simulation was built that visualizes three key representations simultaneously: the **amplitude** distribution across all basis states, the geometric rotation in the two-dimensional Grover subspace, and the evolving probability distribution over successive iterations.

The simulation displays a quantum circuit at the top, showing the **oracle** and **diffusion** operations being applied. The left panel presents a bar chart of **amplitudes** for all computational basis states, with the target state highlighted. The center panel shows the geometric rotation interpretation, where each Grover iteration rotates the state vector by 2θ toward the target. The right panel tracks the measurement probability of the target state as it grows through successive iterations.

Figure 9: Interactive Grover simulation showing quantum circuit, amplitudes, geometric rotation, and probabilities.

40A2D1B0C6A984378D38C1C2816F5CB59A4BA19970B32A11EED3C79FE5646866

Insights from Building the Simulation

Several observations became clearer through the process of building the simulation. The geometric interpretation of **Grover's algorithm** is far more intuitive when visualized as a rotation in a two-dimensional subspace than when described purely through matrix algebra.

The **oracle** does not find the answer. It only flips the phase of the marked state, which by itself produces no observable change. The **diffusion** operator is the component responsible for amplification through **interference**, reflecting all **amplitudes** about their mean so that the marked state gains **amplitude** at the expense of the others.

SIMULATION INSIGHTS

The oracle marks — it does not find. The diffusion operator amplifies. Together they produce a rotation: each iteration brings the state closer to the target. After $\pi/4\sqrt{N}$ steps, measurement yields the answer.

INTERACTIVE DEMONSTRATION

Explore the full interactive simulation with adjustable parameters, step-by-step oracle and diffusion controls, and real-time visualization of amplitude, geometric rotation, and probability evolution.

Interactive simulation: **Grover's Algorithm Visualization**

40A2D1B0C6A984378D38C1C2816F5CB59A4BA19970B32A11EED3C79FE5646866

8. Hybrid Computational Architectures

The most realistic vision for the future of computing is not one in which **quantum computers** replace **classical machines**. Instead, future systems will likely combine multiple specialized processors within a unified architecture.

Classical processors will orchestrate the overall computation and handle general-purpose logic. **GPUs** and classical accelerators will continue to serve large-scale numerical workloads. **Quantum processors** will be invoked selectively for operations that exploit **superposition** and **interference**, acting as specialized accelerators rather than replacements.

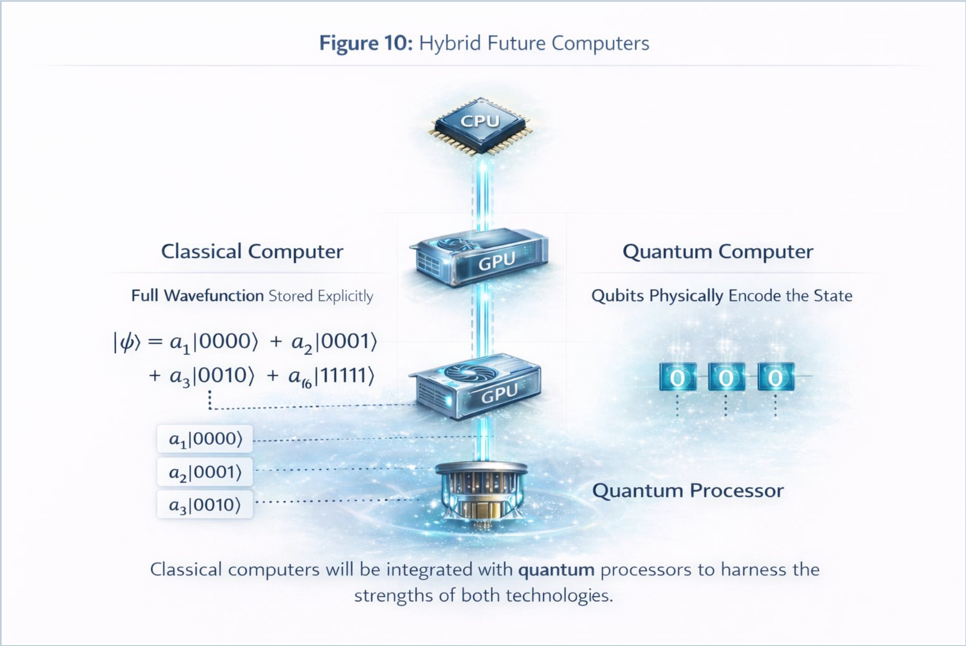

Figure 10: Hybrid architectures combining classical CPUs, GPUs, and quantum processors.

40A2D1B0C6A984378D38C1C2816F5CB59A4BA19970B32A11EED3C79FE5646866

9. Conclusion

Quantum computers represent an important extension of the computational landscape, but they should not be viewed as universal replacements for **classical machines**.

By examining computation from a **physical perspective**, we see that different computational architectures arise from different underlying **physical process**es. Mechanical systems can perform sorting through geometry, **classical digital** computers perform symbolic manipulation through logical operations, and **quantum processors** exploit **interference** and **superposition** to evolve complex state spaces.

CENTRAL THESIS

The future of computing is not about one architecture replacing another.
It is about an ecosystem of specialized machines, each leveraging the
physical principles best suited to the problems they solve.

Acknowledgment

The author would like to acknowledge the IQM School of **Quantum Computing** for providing a structured and rigorous introduction to **quantum computing** concepts. The lectures and educational resources offered through the program were instrumental in shaping the foundational perspective presented in this paper. The clarity of the curriculum helped establish a solid conceptual understanding of **quantum systems**, quantum algorithms, and the physical principles that distinguish **quantum computation** from classical approaches. This acknowledgment reflects sincere appreciation for the quality and accessibility of the program.

40A2D1B0C6A984378D38C1C2816F5CB59A4BA19970B32A11EED3C79FE5646866

References

[1] **Feynman, R. P.**
Simulating Physics with Computers
International Journal of Theoretical Physics, 21(6/7), 467–488, 1982.

[2] **Grover, L. K.**
A Fast Quantum Mechanical Algorithm for Database Search
Proceedings of the 28th Annual ACM Symposium on Theory of Computing, pp. 212–219, 1996.

[3] **Nielsen, M. A. & Chuang, I. L.**
Quantum Computation and Quantum Information
Cambridge University Press, 10th Anniversary Edition, 2010.

[4] **Aaronson, S.**
Quantum Computing Since Democritus
Cambridge University Press, 2013.

40A2D1B0C6A984378D38C1C2816F5CB59A4BA19970B32A11EED3C79FE5646866