Seminar on Quantum Computing

Introduction

A quantum computer is a computer that exploits quantum mechanical phenomena. On small scales, physical matter exhibits properties of both particles and waves, and quantum computing takes advantage of this behavior using specialized hardware. Classical physics cannot explain the operation of these quantum devices, and a scalable quantum computer could perform some calculations exponentially faster than any modern "classical" computer. Theoretically a large-scale quantum computer could break some widely used encryption schemes and aid physicists in performing physical simulations; however, the current state of the art is largely experimental and impractical, with several obstacles to useful applications.

The basic unit of information in quantum computing, the qubit (or "quantum bit"), serves the same function as the bit in classical computing. However, unlike a classical bit, which can be in one of two states (a binary), a qubit can exist in a superposition of its two "basis" states, a state that is in an abstract sense "between" the two basis states. When measuring a qubit, the result is a probabilistic output of a classical bit. If a quantum computer manipulates the qubit in a particular way, wave interference effects can amplify the desired measurement results. The design of quantum algorithms involves creating procedures that allow a quantum computer to perform calculations efficiently and quickly.

Quantum computers are not yet practical for real-world applications. Physically engineering high-quality qubits has proven to be challenging. If a physical qubit is not sufficiently isolated from its environment, it suffers from quantum decoherence, introducing noise into calculations. National governments have invested heavily in experimental research aimed at developing scalable qubits with longer coherence times and lower error rates. Example implementations include superconductors (which isolate an electrical current by eliminating electrical resistance) and ion traps (which confine a single atomic particle using electromagnetic fields).

In principle, a classical computer can solve the same computational problems as a quantum computer, given enough time. Quantum advantage comes in the form of time complexity rather than computability, and quantum complexity theory shows that some quantum algorithms are exponentially more efficient than the best-known classical algorithms. A large-scale quantum computer could in theory solve computational problems that are not solvable within a reasonable timeframe for a classical computer. This concept of additional ability has been called "quantum supremacy". While such claims have drawn significant attention to the discipline, near-term practical use cases remain limited.

History:

For many years, the fields of quantum mechanics and computer science formed distinct academic communities. Modern quantum theory developed in the 1920s to explain perplexing physical phenomena observed at atomic scales, and digital computers emerged in the following decades to replace human computers for tedious calculations. Both disciplines had practical applications during World War II; computers played a major role in wartime cryptography, and quantum physics was essential for nuclear physics used in the Manhattan Project.

As physicists applied quantum mechanical models to computational problems and swapped digital bits for qubits, the fields of quantum mechanics and computer science began to converge. In 1980, Paul Benioff introduced the quantum Turing machine, which uses quantum theory to describe a simplified computer.

When digital computers became faster, physicists faced an exponential increase in overhead when simulating quantum dynamics, prompting Yuri Manin and Richard Feynman to independently suggest that hardware based on quantum phenomena might be more efficient for computer simulation.

In a 1984 paper, Charles Bennett and Gilles Brassard applied quantum theory to cryptography protocols and demonstrated that quantum key distribution could enhance information security.

Quantum algorithms then emerged for solving oracle problems, such as Deutsch's algorithm in 1985, the Bernstein?Vazirani algorithm in 1993, and Simon's algorithm in 1994.

These algorithms did not solve practical problems, but demonstrated mathematically that one could gain more information by querying a black box with a quantum state in superposition, sometimes referred to as quantum parallelism.

Peter Shor built on these results with his 1994 algorithm for breaking the widely used RSA and Diffie?Hellman encryption protocols, which drew significant attention to the field of quantum computing. In 1996, Grover's algorithm established a quantum speedup for the widely applicable unstructured search problem. The same year, Seth Lloyd proved that quantum computers could simulate quantum systems without the exponential overhead present in classical simulations, validating Feynman's 1982 conjecture.

Over the years, experimentalists have constructed small-scale quantum computers using trapped ions and superconductors.

In 1998, a two-qubit quantum computer demonstrated the feasibility of the technology, and subsequent experiments have increased the number of qubits and reduced error rates.

In 2019, Google AI and NASA announced that they had achieved quantum supremacy with a 54-qubit machine, performing a computation that is impossible for any classical computer. However, the validity of this claim is still being actively researched.

Quantum information processing:

Computer engineers typically describe a modern computer's operation in terms of classical electrodynamics.

Within these "classical" computers, some components (such as semiconductors and random number generators) may rely on quantum behavior, but these components are not isolated from their environment, so any quantum information quickly decoheres.

While programmers may depend on probability theory when designing a randomized algorithm, quantum mechanical notions like superposition and interference are largely irrelevant for program analysis.

Quantum programs, in contrast, rely on precise control of coherent quantum systems. Physicists describe these systems mathematically using linear algebra. Complex numbers model probability amplitudes, vectors model quantum states, and matrices model the operations that can be performed on these states. Programming a quantum computer is then a matter of composing operations in such a way that the resulting program computes a useful result in theory and is implementable in practice.

As physicist Charlie Bennett describes the relationship between quantum and classical computers,

A classical computer is a quantum computer ... so we shouldn't be asking about "where do quantum speedups come from?" We should say, "well, all computers are quantum. ... Where do classical slowdowns come from?"

Communication:

Quantum cryptography enables new ways to transmit data securely; for example, quantum key distribution uses entangled quantum states to establish secure cryptographic keys. When a sender and receiver exchange quantum states, they can guarantee that an adversary does not intercept the message, as any unauthorized eavesdropper would disturb the delicate quantum system and introduce a detectable change. With appropriate cryptographic protocols, the sender and receiver can thus establish shared private information resistant to eavesdropping.

Modern fiber-optic cables can transmit quantum information over relatively short distances. Ongoing experimental research aims to develop more reliable hardware (such as quantum repeaters), hoping to scale this technology to long-distance quantum networks with end-to-end entanglement. Theoretically, this could enable novel technological applications, such as distributed quantum computing and enhanced quantum sensing.

Algorithms:

Progress in finding quantum algorithms typically focuses on this quantum circuit model, though exceptions li