An Introduction to Quantum Computing

Frank Rioux

Emeritus Professor of Chemistry

College of St. Benedict | St. John’s University

The goal of this presentation is to demonstrate how a quantum computer exploits quantum mechanical effects such as *superpositions*, *entanglement* and *interference* to perform new types of calculations that are impossible on a classical computer. Quantum computation is therefore nothing less than a distinctly new way of harnessing nature. Adapted from David Deutsch’s, *The Fabric of Reality*, p. 195.

Whereas classical computers perform operations on classical bits, which can be in one of two discrete states, 0 or 1, quantum computers perform operations on quantum bits, or *qubits,* which can be put into any superposition of two quantum states, |0> and |1>. Peter Pfeifer, McGraw-Hill Encyclopedia of Science and Industry.

Superpositions have no classical interpretation. They are *sui generis*, an intrinsically quantum-mechanical construct... N. David Mermin

The nature of the relationships which the superposition principle requires to exist between the states of any system is of a kind that cannot be explained in terms of familiar physical concepts. One cannot in the classical sense picture a system being partly in each of two states and see the equivalence of this to the system being completely in some other state. There is an entirely new idea involved, to which one must get accustomed and in terms of which one must proceed to build up an exact mathematical theory, without having any detailed classical picture. P. A. M. Dirac, *The Principles of Quantum Mechanics*, p. 12.

Entanglement is a special type of superposition in which two or more *quons* behave as a single entity no matter how great their separation. They are always in contact via a non-local interaction that is *unmediated*, *unmitigated* and *immediate* (Nick Herbert, *Quantum Reality,* p. 214). Entanglement is a unique resource that quantum computers exploit which enables them to outperform classical computers.

Using the *Quirk Quantum Simulator* (algassert.com/quirk), the examples that follow will show how a quantum computer calculates, teleports, searches, factors and simulates using superpositions, entangled states and interference effects. The **Appendix** provides mathematical detail about the quantum states and quantum gates that appear in the quantum circuits that follow.

**Parallel Quantum Calculation**

A quantum computer can calculate like a classical digital computer, but it can also calculate in a parallel mode that is not possible with a classical computer that operates only on 0s and 1s.

Haroche and Raimond (*Exploring the Quantum*, p. 96) describe this process as follows: "By superposing the inputs of a computation, one operates the machine 'in parallel', making it compute simultaneously all the values of a function and keeping its state, before any final bit detection is performed, suspended in a coherent superposition of all the possible outcomes." See below on the right and in the Quirk circuit.



<http://www.users.csbsju.edu/~frioux/q-intro/QuantumComputerIntro.pdf>



**Deutsch’s Algorithm**

However, as Haroche and Raimond note, on a practical level only one result can be realized for each operation of the circuit because on measurement the superposition created by the circuit collapses to one of the states forming the superposition. David Deutsch created the following algorithm to determine a fundamental characteristic of f(x), blue in both circuits, more efficiently than a classical computer.



<http://www.users.csbsju.edu/~frioux/q-intro/DeutschProblemBrief.pdf>



**Simple Teleportation**

Quantum teleportation is a uniquely quantum method for transferring a state from one location to another. It does, however, require the assistance of a classical communication channel.

<http://www.users.csbsju.edu/~frioux/q-intro/SimpleTeleportationSoln.pdf>





**A Quantum Search**

Searching and finding the prime factors of large integers are two areas where quantum computers, if of sufficient size and stability, will excel with respect to classical computers. Here’s a simple quantum search algorithm which successfully finds the red arrow.

<http://www.users.csbsju.edu/~frioux/q-intro/FourCardMonte.pdf>





**Shor’s Factorization Algorithm**

Shor’s factorization algorithm is presented in the following link and Quirk circuit, but will not be discussed further here because of its greater complexity. If Shor’s algorithm is implemented on a sufficiently large quantum computer all existing security codes will be vulnerable.

<http://www.users.csbsju.edu/~frioux/q-intro/ShorAlgorithmSummary.pdf>



However quantum mechanics also works the other side of the street. As demonstrated in the following link, it is possible to create unbreakable secret keys using entangled quantum particles (quons).

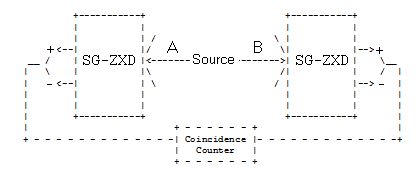
<http://www.users.csbsju.edu/~frioux/q-intro/EkertSecretKey.pdf>

**Quantum Simulation**

Richard Feynman was the first to demonstrate that classical computers are not capable of adequately simulating all natural phenomena. “I want to talk about the possibility that there is to be an *exact* simulation, that the computer will do *exactly* the same as nature. I'm not happy with all the analyses that go with just the classical theory, because nature isn't classical, dammit. And if you want to make a simulation of nature, you'd better make it quantum mechanical, and, by golly, it's a wonderful problem because it doesn't look so easy.” Simulating Physics with Computers, a lecture at MIT in 1981.

"Quantum simulation is a process in which a quantum computer simulates another quantum system. Because of the various types of quantum weirdness, classical computers can simulate quantum systems only in a clunky, inefficient way. But because a quantum computer is itself a quantum system, capable of exhibiting the full repertoire of quantum weirdness, it can efficiently simulate other quantum systems. *The resulting simulation can be so accurate that the behavior the computer will be indistinguishable from the behavior of the simulated system itself***.**" (Seth Lloyd, *Programming the Universe*, p. 149.)

In 1951 David Bohm proposed a thought experiment that illuminated the conflict between classical realism and quantum mechanics first articulated by Albert Einstein, Boris Podolsky and Nathan Rosen (EPR) in 1935. John Bell took up the issue in the 1960s demonstrating that the conflict could be adjudicated in the laboratory. The experimental results that followed (and continue today) support the quantum mechanical view of reality. The figure and link below present Bohm’s thought experiment and a Bell-like extension. [SG-ZXD = Stern-Gerlach magnet oriented in the Z, X, or D (diagonal) direction].



<http://www.users.csbsju.edu/~frioux/q-intro/BohmEPR-Extended.pdf>

The following two-qubit circuit provides a simulation of the Bohm-Bell *thought* experiments, which to my knowledge have not been performed in the laboratory for spin ½ particles as diagramed above.



“One of the most promising applications of quantum computing is solving classically intractable chemistry problems. This may enable the design of new materials, medicines, catalysts, or high-temperature superconductors.” (arXiv:1808.10402v2) These calculations are enormously complex so I will just show a simple circuit that calculates the magnetic interaction between an electron and a proton, a small contribution to the total energy of any molecule. The X, Y, Z, H and CNOT matrix operators can be found in the Appendix. The more significant contributions to the energy of a molecule (electron kinetic energy, electron-electron, electron-proton and proton-proton potential energy) can be represented by more complicated sequences of these operators, operating on many more quantum bits.



<http://www.users.csbsju.edu/~frioux/stability/HAtomHyperfineShort.pdf>

**Summary**

The goal of this presentation was to provide an introduction to the field of quantum computing. To that end, except in the case of Shor’s algorithm, only two-qubit circuits have been used to show how a quantum computer calculates, teleports, searches, factors and simulates.

In addition to providing mathematical background, the appendix contains utility circuits which show how entangled states are generated and how quantum errors are corrected.

**Recommended Reading**

* *The Quest for the Quantum Computer*, Julian Brown
* *Quantum Reality*, Nick Herbert
* *Where Does the Weirdness Go?*, David Lindley
* *The Cosmic Code: Quantum Physics as the Language of Nature*, Heinz Pagels
* *The Age of Entanglement*, Louisa Gilder
* *The Quantum Divide*, Christopher Gerry and Kimberley Bruno
* *Quantum Weirdness*, William J. Mullin
* *Through Two Doors at Once*, Anil Ananthaswamy
* *The Meaning of Quantum Theory*, Jim Baggott
* Dream Machine, Rivka Galchen, *The New Yorker*, May 2, 2011

**Appendix**

The qubit states used in quantum computing occupy the poles of a Bloch sphere.





The relationship of this representation of the Bloch sphere and the X, Y and Z Pauli matrices (see below) is provided in the following link.

<http://www.users.csbsju.edu/~frioux/q-intro/BlochSphereAlt.pdf>

Here are the matrix operators (gates) that have been used in the quantum circuits presented previously.



The following equations illustrate the superposition principle.



The following equations illustrate the operation of the Hadamard gate on several state vectors. The Hadamard matrix is a Fourier transform and unitary. Doing it twice is equivalent to doing nothing. The X, Y, Z, CNOT and CZ matrices are also unitary.



**Generating Entangled States**

As was noted in the introduction, quantum computer applications exploit entangled states to outperform classical computers. This circuit generates one of the four two-qubit entangled Bell states. Changing the input qubits from |1>|1> to |0>|0> or |0>|1> or |1>|0> generates the other Bell states.





Obviously a practical quantum computer requires a greater level of entanglement. The following circuit generates one of the eight three-quibit GHZ (Greenberger-Horne-Zeilinger) entangled states.





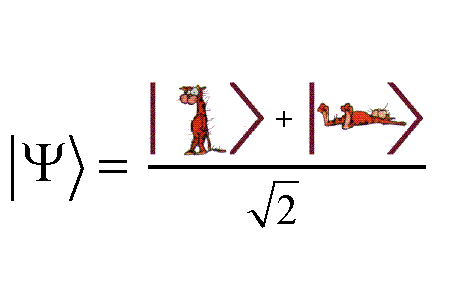
Adding additional CNOT gates to this circuit generates higher entangled states.

**Quantum Error Correction**

Quantum computers are fragile and therefore prone to random internal errors which must be corrected if the quantum circuit is to successfully complete the computational task for which it is designed. The following algorithm (blue) corrects a phase error (red) on the top qubit created by an extraneous Z gate interaction, but does nothing if there is no phase error.

 if there is no error --> 



<http://www.users.csbsju.edu/~frioux/q-intro/CorrectQuantumError.pdf>

Schrödinger’s cat (actually Bill the Cat) is in a superposition of being both alive and dead.