

# A Brief Look At Quantum Computing for Computer Scientists

Bradley S. Rubin, Ph.D.  
Graduate Programs in Software  
University of St. Thomas

In classical computing, the bit is the fundamental unit of computation and it can have two states, 0 and 1. In quantum computing, the qubit is the fundamental unit of computation and takes the strange form (where psi is known as the wave function)

$$|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle$$

subject to a constraint, called the normalization constraint

$$(|\alpha_0|)^2 + (|\alpha_1|)^2 = 1 = \langle\psi|\psi\rangle$$

where each alpha coefficient is a complex number of the form  $a + bi$ . So, there are a total of four continuous dimensions in the qubit information space. The strange notation is called bra-ket notation, created by Paul Dirac, and is commonly used by physicists. Computer scientists are initially more comfortable with an alternative matrix notation, so here is the equivalent notation for representing a conventional 0 and 1 bit.

$$|0\rangle \text{ or } \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad |1\rangle \text{ or } \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

In quantum mechanics, particles can be in two states simultaneously. This is called superposition. A qubit can be in a superposition in any proportion between 0 and 1 as long as the normalization constraint holds. For example, a qubit in superposition with equal parts 0 and 1 would be represented by

$$|\psi\rangle = \frac{1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle$$

An interesting thing happens when a qubit in this state is measured. The output is randomly either a 0 or 1, and the coefficient squared tells us the probability of measuring each state. For the above qubit, the square of each coefficient is  $\frac{1}{2}$ , so the probability of measuring a 0 is  $\frac{1}{2}$  and of measuring a 1 is  $\frac{1}{2}$ . So, the normalization constraint is really just a statement that all probabilities must sum to 1.

Physically, states can be represented in a number of ways. For example, subatomic particles have a property called spin (although nothing is really spinning!), which

can be either up or down. Or, photons can be polarized vertically or horizontally, or left circularly or right circularly.

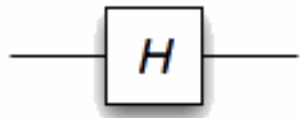
Perhaps even more strangely, two qubits can be entangled, or correlated. This means that making a measurement on one qubit of an entangled pair forces a change in state of the other qubit. And, this happens instantaneously (faster than the speed of light), even if the qubits are located on opposite sides of the universe. As an example, here is a specific entangled qubit pair.

$$|\psi\rangle = \frac{1}{\sqrt{2}} |00\rangle + \frac{1}{\sqrt{2}} |11\rangle$$

Quantum computation leverages both of these two key ideas (superposition and entanglement) to perform some algorithms more efficiently than can be done with conventional computing.

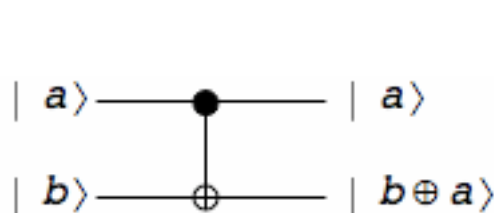
In classical computation, we use logic gates like AND, OR, NOT, and XOR to perform logical operations. The NAND gate is universal, allowing for any logic circuit to be constructed (cumbersomely) from this sole gate. In quantum computation, logic gates are totally unfamiliar, going by names like X, Y, Z, CNOT, Hadamard, etc. We can view a quantum computation as a matrix multiplication between the qubit matrix and the quantum gate matrix, which produces an output matrix (or output qubit(s) state). The Hadamard, CNOT, and a few others form the universal set for quantum computing.

Here is a 1-qubit Hadamard gate



$$\text{Hadamard} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

And here is a 2-qubit CNOT gate

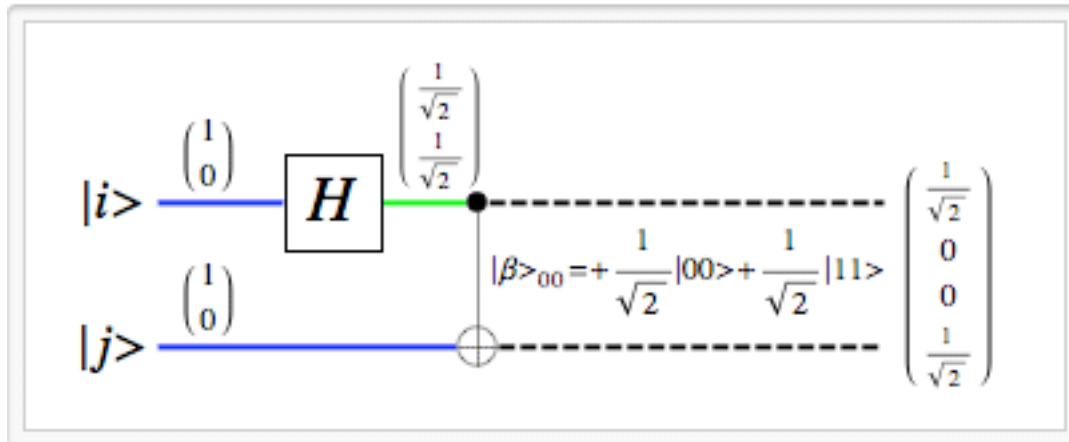


$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

In conventional computing, logic gates can be irreversible. For example, if we are given an AND gate output of 0, we don't uniquely know the two inputs (they could be 00, 01, or 10). The qubit, by contrast, must obey the quantum mechanical Schrödinger equation that has a piece called a Hamiltonian that describes the energy

evolution of a particle, and it is time reversible. This implies that quantum computation logic gates must also be reversible (and the two gates above have this property).

Logic gates can be combined into quantum circuits. Here is an example that uses both of these gates. It takes two input qubits, passes the upper qubit through a Hadamard gate, passes the result along with the lower qubit through a CNOT gate, and produces an entangled qubit pair.



Here is the matrix form of calculating the output from the inputs for the above circuit. The circle symbol with the x inside is a matrix operation called a tensor product.

$$\begin{aligned}
 |\beta\rangle &= \text{CNOT} (\text{H} \otimes \text{I}) ( |0\rangle \otimes |0\rangle ) \\
 &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \left( \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right) \left( \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right) \\
 &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{pmatrix}
 \end{aligned}$$

Note that the “wires” in a quantum circuit represent the progression of time, and have nothing to do with electrical conductivity.

The big deal about quantum computing is that certain specific algorithms, but by no means all algorithms, can be implemented much more efficiently in their computational complexity than can be achieved with conventional computing. For example, factoring is a problem that grows exponentially in computation time as the number to be factored grows in size. The public key cryptographic algorithm RSA leverages this fact to secure much of our information. A quantum algorithm, called the Shor factoring algorithm, runs in polynomial time, specifically  $\log N$ , making factoring feasible. Another algorithm, the Grover algorithm, speeds up brute force search.

Quantum computing is rich in theory, but poor in implementation. Systems of only a handful of qubits have been realized. The journal Science recently reported on the ability to factor the number 15 into 5 and 3 on a chip. One big problem is that as qubits interact with the environment, they undergo a phenomena called decoherence, which destroys the special state and creates errors. Efforts to perform quantum error correction can help fix these errors, but introduce other quantum circuits in doing so, which themselves introduce errors. Some believe that this is a losing game, and a quantum computer of any significance will never be built. Others believe that quantum computers will eventually usher in a new era of special capabilities (and security challenges).

There is a form of quantum computing, called quantum cryptography, which does have real commercial implementations available for purchase today. These systems use the sending and receiving for individual photons, polarized in one of four directions, to distribute a conventional key to another party to subsequently be used with a conventional cryptographic algorithm, such as the Advanced Encryption Standard (AES) or a One-Time Pad (OTP). These systems have the additional unique feature that passive eavesdroppers can be detected because when they measure a photon, they disturb its state, and the two legitimate parties can detect this.

Whatever the future direction these technologies take, the investigation in this field opens up new ideas, computation techniques, and rethinking every fundamental question about what it means to compute, and provides a theoretical and experimental test bed for exploring the strange world of quantum mechanics. It also requires a blend of the skills and knowledge in both the domains of physics and computer science.

## References

- Greenstein, G. and Zajonc, A., “The Quantum Challenge: Modern Research on the Foundations of Quantum Mechanics”, 2<sup>nd</sup> edition, Jones and Bartlett, 2005.
- Kaye, P., Laflamme, R. and Mosca, M., “Introduction to Quantum Computing”, Oxford University Press, 2007.