Johns Hopkins Engineering

Module 13: Quantum Computation

EN.605.204: Computer Organization

JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

What is a qubit?

A **qubit** can be thought of as an abstract mathematical object, just like a bit, but it has one major difference. A qubit can be in a 0-state, a 1-state, or a **superposition** (combination) of the two. The 0-state and 1-state of a qubit are called a **computational basis** and the superposition of a single qubit can be expressed as a linear combination of the basis states.

0 basis state

1 basis state

$$|0\rangle \qquad |1\rangle$$

Computational basis states

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

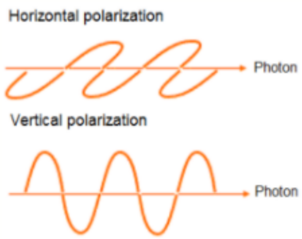
Superposition

Partially in the 0-state, partially in the 1-state

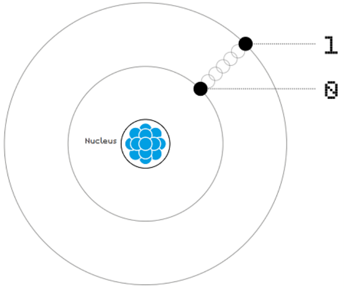
Physical Qubits

Though we'll only talk about qubits abstractly, there are currently 3 primary ways a qubit can be represented physically:

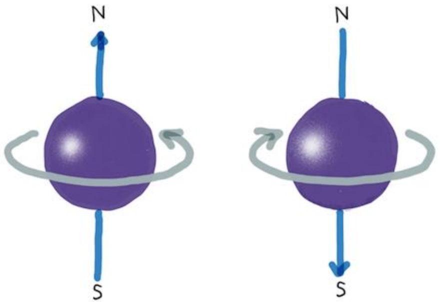
1. Photon polarization
2. Atomic energy levels
3. Electron spin



Photon polarization



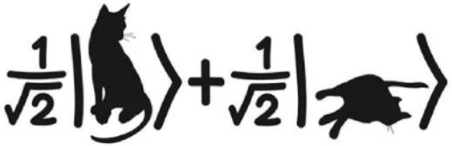
Atomic energy level



Electron spin

State Vector of a Qubit

A single qubit can be in the 0-state, the 1-state, or a superposition. We describe the current state of a qubit using a **state vector** (psi) as shown below. Qubits are probabilistic, that is there is an alpha^2 chance that measuring the qubit yields a 0 and a beta^2 chance of getting a 1. Alpha and beta are called **complex/probability amplitudes** and the sum of their squares must equal 1.



Schrodinger's Cat

$$\boxed{|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, |\alpha^2| + |\beta^2| = 1}$$

Probability amplitudes, sum of squares must be 1

State Vector of a Qubit

Below are the state vectors for the 0-state, the 1-state, and an equally weighted superposition respectively. Note we get the 0-state by setting beta to 0, the 1-state by setting alpha to 1, and our superposition will yield a 0-state or 1-state with probability ½.

$$|\psi\rangle = (1^2)|0\rangle + (0^2)|1\rangle = |0\rangle, \alpha = 1, \beta = 0$$

State vector of qubit in 0 computational basis state

$$|\psi\rangle = (0^2)|0\rangle + (1^2)|1\rangle = |1\rangle, \alpha = 0, \beta = 1$$

State vector of qubit in 1 computational basis state

$$\psi = \frac{|0\rangle + |1\rangle}{\sqrt{2}} = (\frac{1}{\sqrt{2}})|0\rangle + (\frac{1}{\sqrt{2}})|1\rangle, \alpha = \frac{1}{\sqrt{2}}, \beta = \frac{1}{\sqrt{2}}$$

Equally weighted superposition of basis states

Encoding Data with Qubits

With classical bits, linearly scaling the number of bits linearly scales the state space. That is, with 1 bit you can represent a single 1-bit number at-a-time, with 2 bits you can represent a single 2-bit number at-a-time, and so on.

Increasing the number of qubits **linearly exponentially increases the state space** we can represent! That is, with 1 qubit we can represent two values at once, with 2 qubits we can represent 4 values at once, and so on.

$$00, 01, 10, 11$$

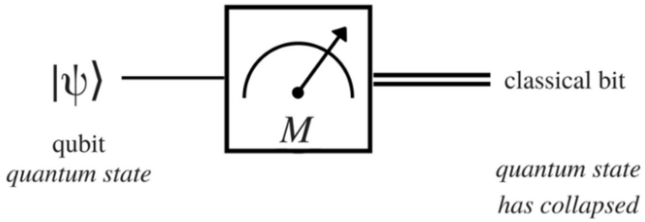
2 classical bits allow us to represent one of 2^2=4 values at once

$$|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle$$

A superposition of 2 qubits allows us to represent 2^2=4, all values at once

Measuring the State of a Qubit

Unfortunately, when we measure a qubit that is in a superposition, we will only receive 1 classical bit of information: a 0 or 1 (with probability $alpha_x^2$). This is because the superposition state is extremely fragile and is “destroyed” when it is observed (measured). So, we can learn nothing about $alpha_x$ directly. Measuring a qubit in the 0 basis state yields a 0 and performing a measurement on a qubit in the 1 basis state yields a 1.



$$|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle$$

Not directly observable

Computation with a Single Qubit

Though qubits are abstract mathematical objects, it will be useful to assign values to them in the form of 2-dimensional unit vectors so we can perform computation on them.

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Vector representation for 0 basis state

$$|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Vector representation for 1 basis state

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \alpha \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \beta \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

Vector representation for superposition

X Gate (Quantum "Not")

The **quantum NOT gate**, called a **Pauli X gate** and denoted 'X', inverts the 0 basis state to the 1 basis state and vice versa. In the case of a superposition, the X gate acts linearly upon the state of the qubit thus interchanging swapping the probability of alpha or beta resulting from a measurement.

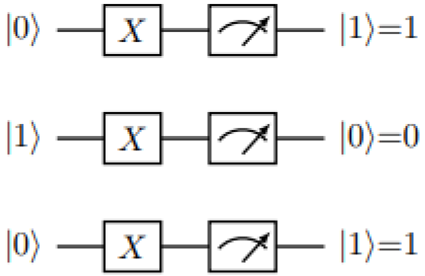
$$X|0\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \qquad X|1\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$X(\alpha|0\rangle + \beta|1\rangle) = \alpha|1\rangle + \beta|0\rangle = \begin{bmatrix} \beta \\ \alpha \end{bmatrix}$$

X Gate (Quantum "Not")

$$|\psi\rangle = |010\rangle, X|\psi\rangle = X|010\rangle = |101\rangle$$

3 qubits in 1 of the 8 basis states, flip their states



3-qubit NOT circuit

$$|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle$$

$$X|\psi\rangle = \alpha_{00}|11\rangle + \alpha_{01}|10\rangle + \alpha_{10}|01\rangle + \alpha_{11}|00\rangle$$

Qubits are in a superposition, flip their states

X Gate (Quantum “Not”)

The **quantum NOT gate**, called a **Pauli X gate** and denoted ‘X’, inverts the 0 basis state to the 1 basis state and vice versa. In the case of a superposition, the X gate acts linearly upon the state of the qubit thus interchanging swapping the probability of alpha or beta resulting from a measurement.

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Matrix representation of X

$$X|0\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} = |1\rangle$$

Applying X gate to qubit in 0 basis state

$$X|1\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} = |0\rangle$$

Applying X gate to qubit in 1 basis state

Hadamard Gate

A Hadamard gate takes a single qubit and does 1 of 2 things:

1. If the qubit is in a base state the result is an equally-weighted superposition
2. If the qubit is in a superposition the result is the corresponding base state

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

Qubit in 0 basis state put into superposition

$$H|1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

Qubit in 1 basis state put into superposition

$$H(\frac{|0\rangle + |1\rangle}{\sqrt{2}}) = |0\rangle$$

Superposition is converted to 0 basis state

$$H(\frac{|0\rangle - |1\rangle}{\sqrt{2}}) = |1\rangle$$

Superposition is converted to 1 basis state

Hadamard Gate

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Matrix representation of H

$$H|0\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 1 \end{bmatrix} = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

H applied to a qubit in the 0 base state

$$H|1\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ -1 \end{bmatrix} = \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

H applied to a qubit in the 1 base state

Tensor Product

A **tensor product** can most simply be thought of as a way to combine multiple qubits whether they are in a base state or a superposition. For each additional qubit beyond the first you grow the state space exponentially. That is, the tensor product of 2 qubits has 4 base states, the tensor product of 4 qubits has 16 base states, and so on.

$$|0\rangle \otimes |1\rangle = |01\rangle$$

$$|0\rangle \otimes |1\rangle \otimes |1\rangle \otimes |0\rangle = |0110\rangle$$

$$\frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes |0\rangle = \frac{|00\rangle + |10\rangle}{\sqrt{2}}$$

$$\frac{|0\rangle - |1\rangle}{\sqrt{2}} \otimes |0\rangle \otimes |1\rangle = \frac{|001\rangle - |101\rangle}{\sqrt{2}}$$

Tensor Product

$$|0\rangle \otimes |1\rangle = |01\rangle \longrightarrow |0\rangle \otimes |1\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle$$

Tensor product of 2 qubits in base states

$$\frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes |0\rangle = \frac{|00\rangle + |10\rangle}{\sqrt{2}} \longrightarrow \frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes |0\rangle = \frac{\begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix}}{\sqrt{2}} = \frac{1}{\sqrt{2}} \left(\overset{|00\rangle}{\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}} + \overset{|10\rangle}{\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}} \right)$$

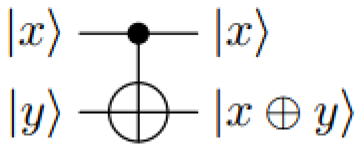
Tensor product of 2 qubits, 1 in a superposition and 1 in base state

Controlled Not (CNOT) Gate – Quantum XOR

A **Controlled Not (CNOT) gate** accepts two qubits, x and y, as input and outputs two qubits: x and x XOR y. This is the quantum analogue to the classical XOR gate. We'll refer to x as our input qubit and y as our target qubit s y is the qubit that will contain the result of the XOR operation.

$$CNOT(|x\rangle, |y\rangle) \rightarrow |x\rangle, |y \oplus x\rangle$$

Formula for a CNOT gate



CNOT circuit

$$\begin{array}{ccc} |0\rangle|0\rangle & \longmapsto & |0\rangle|0\rangle \\ |0\rangle|1\rangle & \longmapsto & |0\rangle|1\rangle \\ |1\rangle|0\rangle & \longmapsto & |1\rangle|1\rangle \\ |1\rangle|1\rangle & \longmapsto & |1\rangle|0\rangle \end{array}$$

Truth table for CNOT

Controlled Not (CNOT) Gate – Quantum XOR

In plain English, the first qubit (x) passes through unchanged while the second qubit (y) gets the value of x XOR y. Or, if you prefer, if alpha is 1 then the XOR flips beta. Otherwise, beta is left unchanged.

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

CNOT matrix

$$|10\rangle = |1\rangle \otimes |0\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

Express tensor product as a column vector

$$CNOT(|10\rangle) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} = |11\rangle$$

Apply CNOT gate to state vector algebraically

Quantum Advantage

We've seen many quantum primitives, but can we demonstrate an example of where a problem can be solved more efficiently with qubits than classical bits? Consider the following (contrived) scenario: We are given a binary function that takes a single bit as an input and produces a single-bit output. We want to compute both f(0) and f(1).

$$f : \{0,1\} \rightarrow \{0,1\}$$

$$f(0) = 0$$ *f* maps 0 → 0

$$f(1) = 1$$ *f* maps 1 → 1

To accomplish this we must execute the function twice.

Quantum Advantage

We've seen many quantum primitives, but can we demonstrate an example of where a problem can be solved more efficiently with qubits than classical bits? Consider the following (contrived) scenario: We are given a binary function that takes a single bit as an input and produces a single-bit output. We want to compute both f(0) and f(1).

$$f : \{0,1\} \to \{0,1\}$$

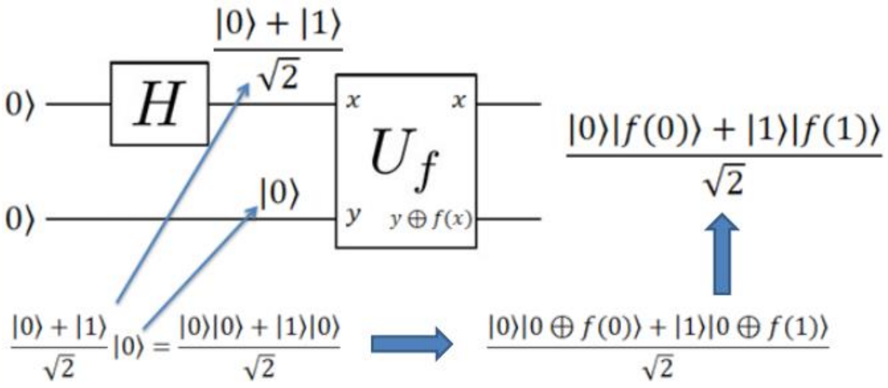
$$f(0) = 0$$ *f* maps 0 → 0

$$f(1) = 1$$ *f* maps 1 → 1

To accomplish this we must execute the function twice.

Quantum Advantage

We can compute f(0) and f(1) simultaneously using only two qubits by taking advantage of the ability to place a qubit into a superposition!



$$U_f : |x, y\rangle \rightarrow |x, y \oplus f(x)\rangle$$