A Quantum of Quantum Computing

Cora Barrett
Om Joshi
Hyo Sun Park
Ági Villányi
Matt Yeh

IAP 2025: MIT Quantum Winter School

Contents

Interference

In this section, we want to highlight the difference between classical and quantum interference. **Interference is the hallmark of all waves!** When waves meet each other, they interact and interfere, creating interference patterns. Interference itself is not a quantum phenomenon. The key difference between classical and quantum interference is that in quantum, a single object (such as a photon) *is a wave* and can interfere with itself. Classically, a wave must be a collection/ensemble, i.e. a single object cannot interfere with itself.

Defining Interference

Waves

To understand interference, we first introduce how waves are conventionally represented in physics. There are many different types of waves around us, such as sound waves, matter waves, electromagnetic waves (i.e. light), and gravitational waves. Here we focus on an electromagnetic plane wave[1] with a single frequency ω traveling in the y direction:

$$\mathcal{E}(y,t) = \mathcal{E}_0 \cos\left(ky - \omega t + \phi\right), \tag{1}$$

where $\mathcal{E}_0$ is the amplitude of the electric field, and k is the wave number related to the wavelength λ by $k = 2\pi/\lambda$, and ϕ is some arbitrary phase. See Figure **??**. The angular frequency ω of the wave is given by

$$\omega = \frac{2\pi}{T} = 2\pi\nu, \tag{2}$$

where T is the period of the wave in units of seconds, and $\nu = 1/T$ is the frequency of the wave in units of hertz (Hz). Note that ω has units of *radians per second*, whereas ν has units of inverse seconds.

Equivalently, we can express the wave with complex numbers. Using the famous Euler identity:

$$e^{i\theta} = \cos\theta + i\sin\theta, \tag{3}$$

[1] A *plane wave* simply means a type of wave whose physical quantity (such as phase) remains constant on any given plane that is perpendicular to the wave's propagation direction $\vec{k}$.

we can rewrite Eq. (1) as

$$\mathcal{E}_0 e^{i(ky-\omega t+\phi)} = \mathcal{E}_0[\cos(ky-\omega t+\phi) + i\sin(ky-\omega t+\phi)]. \tag{4}$$

Note that Eq. (1) is simply the real part of this complex expression.

Waves change their behavior when they encounter each other or some object, such as a slit, in space. For example, they can reflect, refract, diffract, and interfere. Here we focus on diffraction[2] (bending of waves) and interference (addition of waves).

Particle Nature of Light

Using the expression of a plane wave we have introduced, one can precisely calculate where the intensity maxima and minima of the interference patterns will occur. This confirms the wave nature of light. However, Albert Einstein proposed in his famous 1905 paper that the electromagnetic field is quantized in "packets" of energy, which we now call *photons*. Einstein proposed that each photon has energy

$$E_p = h\nu, \tag{5}$$

where h is a fundamental constant called Planck's constant (6.63×10^{-34} J·s), and ν is the frequency of the light.

Einstein's suggestion was proved with the famous photoelectric effect experiment conducted by Robert Andrews Millikan in 1916, for which they won the Nobel Prize in Physics in 1921 and 1923, respectively. In the experiment, monochromatic light was shone onto a metal surface (cathode) attached to a simple circuit loop, as shown in Figure 1. The light excites the electrons on the metal surface (cathode), ejecting electrons out of the metal. The anode is set to a small negative voltage ($-\phi_0$) relative to the cathode, which *repels* electrons. Thus, only the electrons with sufficient energy to overcome the potential difference can reach the anode, thereby creating the so-called "photocurrent". Electrons that make up the photocurrent are called "photoelectrons". Electrons without sufficient energy can still be ejected but will fall back to the cathode, failing to reach the anode. The intensity of the photocurrent and the maximum kinetic energy of the ejected electrons due to the light were closely monitored during the experiment.

[2] Technically, diffraction is a form of interference: it comes from interference between waves passing through an appropriately small aperture.

Millikan Setup

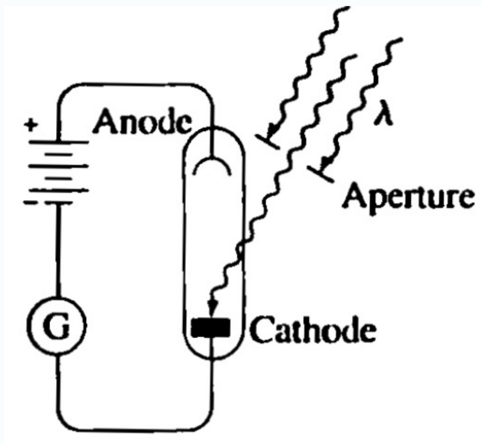


Fig. 1. Experimental setup to test photoelectric effects. Light with a given wavelength λ excites electrons on the cathode. [Townsend, J. S. (2009). *Quantum Physics: A Fundamental Approach to Modern Physics*. University Science Books.]

Classically, light is made up of waves, and its energy, or intensity I, scales with $\mathcal{E}_0{}^2$, where $\mathcal{E}_0$ is the amplitude of the waves. Thus, classically, one would think that the energy of the ejected electrons would depend on the light intensity, regardless of the frequency of light. Moreover, classically, one would expect that even low-intensity light should eject the electrons if shone on the metal for a sufficiently long period of time (energy builds up over time). So, at any frequency, the higher the light intensity, the higher the photocurrent we would expect to observe. However, the experimental results disproved all of these predictions!

Millikan's results showed that the kinetic energy of the ejected electrons actually scales linearly with the *frequency* of light. The photocurrent increases as light intensity goes up, up to some saturation value that depends on the light intensity (the higher the light intensity, the higher the saturation value). What was more surprising was that there seems to be a "cutoff" frequency of light, below which the photocurrent completely ceases no matter how long the light is shone on the metal. Millikan's results are shown in Figure 3.

Polarization to Bloch Sphere

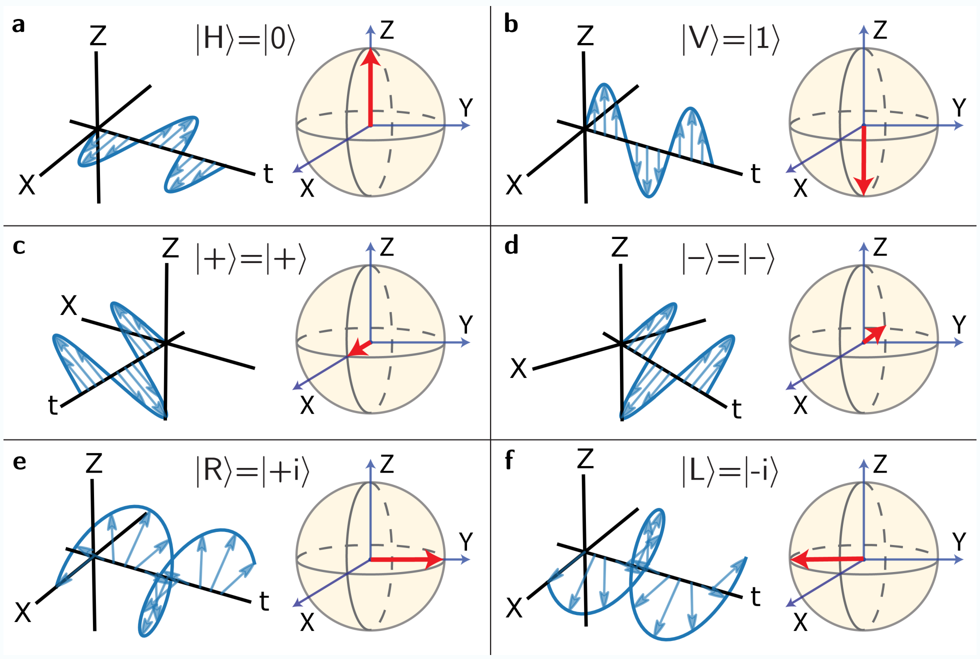


Fig. 2. This figure depicts the mapping between polarization bases to the Bloch sphere.

Using Einstein's theory, Millikan's results can be explained by a simple equation: the maximum kinetic energy K of electrons ejected from the metal by a photon is

$$K = h\nu - W, \tag{6}$$

where W is the work function of the metal, i.e. the minimum energy required to eject an electron out of the metal. The experimental results proved that K is indeed a linear function of ν, as shown in Figure 3. Hence, the experiment clearly demonstrates that light is made of photons.

Millikan's Data

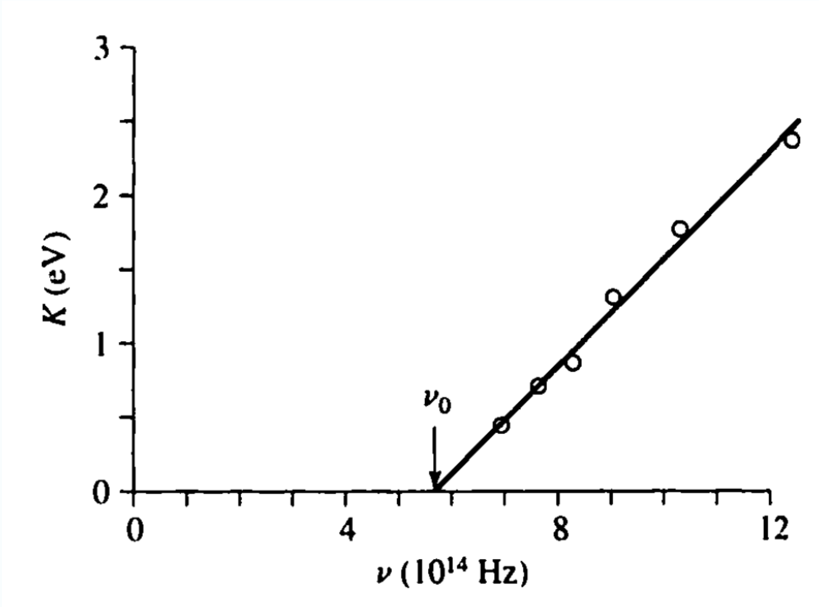


Fig. 3. Millikan's experimental data showing the electron's kinetic energy K as a function of the light frequency ν for sodium. No electrons are ejected below the cutoff frequency $\nu_0 \approx 5.6 \times 10^{14}$ Hz (green light). Millikan extracted the value of Planck's constant from the slope of the curve. [Townsend, J. S. (2009). *Quantum Physics: A Fundamental Approach to Modern Physics*. University Science Books.]

Wave Particle Duality.

We have looked into experiments that proved that light is both a wave *and* a collection of photons, particles of light that are massless. We have also seen that *any* particles with mass, such as atoms, electrons, and protons, are also waves with wavelengths given by their corresponding de Broglie wavelengths.

$$\lambda_{dB} = \frac{h}{p}, \tag{7}$$

In fact, the human body is also a wave! We can calculate the wavelength of the human body using the same formula (7). However, the de Broglie wavelength of a human is extremely small ($\sim 10^{-35}$ m for a person traveling at about 1 m/s). This is why the "particle" nature of a human dominates. Well, you have now understood the so-called "wave-particle duality"!

No Hidden Variables and a Qubit is not a Random Coin

You may have seen videos on various platforms that describe qubits using an analogy to coin tosses. The language goes something like: "a qubit is in an equal superposition of 0 and 1, just like how a coin is both heads and tails when it is spinning in the air". But there is a major issue with this, and the two notions are not at all equal.

The main problem is that the randomness of classical coins is due to our lack of information about the system, rather than inherent randomness of the coin object. Let's consider the following thought experiment. You have a device that allows you to measure the exact initial conditions of the coin and have a model of the dynamics of the coin. Then you could exactly predict the final state of the coin, down to the angle it lands on, by just plugging into the formula and getting out a result. Therefore, a coin toss is actually a deterministic process, physically speaking. Any uncertainty comes about because the experiment wasn't controlled enough, e.g. friction in the air, deviations in the initial force of the coin toss, etc.

In the case of quantum mechanics, there is inherent randomness in the system. Namely, even if one had access to a measurement device that could precisely measure the initial state and every intermediate state of a qubit, there would still be a probabilistic process that occurs when a qubit is observed. Let's return to the coin example. This would be like saying that you can control the experiment perfectly so that you know exactly the angle (superposition) the quantum coin should land in before measurement – but you still wouldn't be able to predict the outcome, heads or tails, run to run. Annoying, isn't it? You and Einstein would agree – this discomfort with the need for probability to describe quantum phenomena led to his famous statement, "God does not play dice with the universe."

It is then interesting to consider the following thought experiment. So how do we know that there is no internal information influencing the measurement result, as there is in classical mechanics? That is, how do we know that there are no *hidden variables* in the system? This is due to the No Hidden Variable Theorem, thanks to John Stewart Bell. We will revisit this when talking about entanglement on Day 3.

Bloch Sphere

As we saw earlier, any two-level qubit system can be represented by a 2x2 density operator,

$$\rho = \frac{1}{2}\begin{pmatrix} 1+s_z & s_x - is_y \\ s_x + is_y & 1-s_z \end{pmatrix} \tag{8}$$

with s_x, s_y, and s_z all real. This can be seen to fulfill the Hermitian and unit-trace properties of the density operator. Already, we see that the system can be mapped to a three-dimensional Cartesian space – but what volume in that space? For that, we turn to the constraints on the trace.

Specifically, because the purity is upper bounded by 1, we can take the trace of ρ^2 to find

$$\mathrm{Tr}[\rho^2] = \frac{1}{2}(1+|\vec{S}|^2) \leq 1 \tag{9}$$

with

$$\vec{S} = \begin{pmatrix} s_x \\ s_y \\ s_z \end{pmatrix} \tag{10}$$

defined as the Bloch vector, containing the coordinates that define the point in 3D space that the density matrix corresponds to. Simplifying, we find that the constraint on this 3D vector is that its magnitude $|\vec{S}|$ is less than or equal to 1, tracing out a unit ball! This is known as the Bloch ball, and is a powerful visualization of qubit states. For example, full quantum state purity is achieved when $|\vec{S}| = 1$, i.e. the vector traces out the surface of the Bloch ball, aka the *Bloch sphere*. Mixed states therefore lie in the interior of the ball, and decoherence can be visualized as the recession of the Bloch vector into the center.

In fact, the specific decomposition of the density matrix in this way has a convenient geometric connection to common observables. Explicitly pulling out the matrices weighted by each of the Bloch coordinates,

$$\rho = \frac{1}{2}(I + s_x X + s_y Y + s_z Z) \tag{11}$$

where X, Y, and Z are the *Pauli matrices*.

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{12}$$

$$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \tag{13}$$

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{14}$$

For density matrices, it can be shown that the equivalent operation to $\langle\psi| O |\psi\rangle$ to find the expectation of an observable O is to take the trace, i.e.

$$\langle O \rangle = \mathrm{Tr}[\rho O] \tag{15}$$

For the Pauli matrices, it can be further shown that in fact, the elements of the Bloch vector correspond to the expectations of each of the Pauli matrices:

$$\vec{S} = \begin{pmatrix} s_x \\ s_y \\ s_z \end{pmatrix} = \begin{pmatrix} \langle X \rangle \\ \langle Y \rangle \\ \langle Z \rangle \end{pmatrix} \tag{16}$$

Hence the Bloch sphere maps to observations made in common measurement bases. This can be further extended to analyze the evolution of qubit states as rotations about certain axes, in the study of quantum gates.

Bloch Sphere

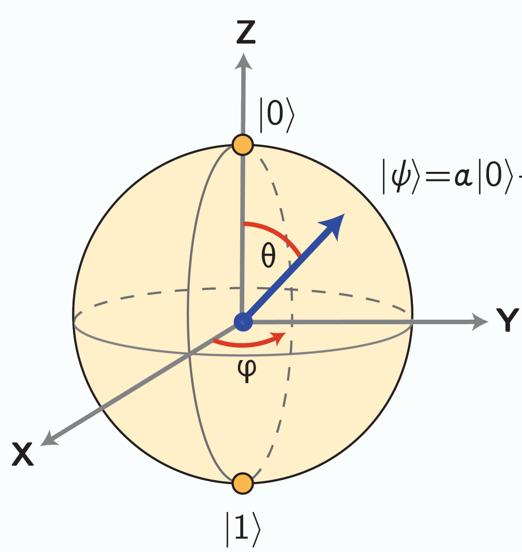


Fig. 4. Here, $\alpha, \beta \in \mathbb{C}$.

As a final note, the Bloch sphere can also be derived from the pure state picture. In this case, any state (up to a global phase) can be written as

$$|\psi\rangle = a\,|0\rangle + be^{i\phi}\,|1\rangle \tag{17}$$
$$= \cos(\theta/2)\,|0\rangle + \sin(\theta/2)e^{i\phi}\,|1\rangle \tag{18}$$

where θ is the polar angle restricted to $0 \leq \theta \leq \pi$ and ϕ is the azimuthal angle restricted to $0 \leq \phi \leq 2\pi$. The factor of $\theta/2$ restricts a to be positive, since relative phase is captured by $e^{i\phi}$. The mapping to a unit sphere is then directly made by corresponding θ and ϕ with spherical coordinates:

$$\vec{S} = \begin{pmatrix} s_x \\ s_y \\ s_z \end{pmatrix} = \begin{pmatrix} \sin(\theta)\cos(\phi) \\ \sin(\theta)\sin(\phi) \\ \cos(\theta) \end{pmatrix} \tag{19}$$

This derivation has a more convenient interpretation in terms of phase. That is, phase lags and advances in ϕ can be directly visualized as precession in the XY plane of the Bloch sphere.

Quantum Circuits

Classical Computers and Universality

A classical computer has gates which act on bits to perform desired logical operations. Examples of these gates, which can act on a single bit or two bits, are outlined in Table 1. Given the discrete nature of the bit, there are only two possible single bit gates: one which does nothing (identity) and one which flips the bit (NOT). For two bit gates, there are a few different possibilities, including the intuitively-named AND and OR gates. We note that a two bit gate takes in two bits and outputs one bit.

When designing a computer, it is important to consider whether the implementation can achieve *all* possible logical operations. One can imagine constructing more and more complex gates to this end. But, in practice, it is also useful to consider the smallest set of logical operations that can be used to generate all computations, since this can simplify physical implementation. This is a property sometimes referred to as *functional completeness* or *universality*.

In classical logic, it turns out the NAND gate alone is a functionally complete gate set. In CMOS logic, this can be implemented using four transistors (two NMOS, two PMOS), helping estimate the number of individual physical devices needed per operation. Note that functionally complete gate sets are not necessarily unique (for example, {NOR} and {NOT, AND} are both functionally complete) nor the most hardware efficient for every operation (for example, NOT can be implemented physically with only two transistors, whereas using NANDs would take more).

Classical Logic Gates

<table>
<tr><th>Gate</th><th>Description</th><th>Truth Table</th></tr>
<tr><td>Identity</td><td>No action</td><td><table><tr><th>Input</th><th>Output</th></tr><tr><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td></tr></table></td></tr>
<tr><td>NOT</td><td>Bit “flip”</td><td><table><tr><th>Input</th><th>Output</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table></td></tr>
<tr><td>AND</td><td>1 iff both bits are 1</td><td><table><tr><th>Input</th><th>Output</th></tr><tr><td>0 0</td><td>0</td></tr><tr><td>0 1</td><td>0</td></tr><tr><td>1 0</td><td>0</td></tr><tr><td>1 1</td><td>1</td></tr></table></td></tr>
<tr><td>NAND</td><td>0 iff both bits are 1</td><td><table><tr><th>Input</th><th>Output</th></tr><tr><td>0 0</td><td>1</td></tr><tr><td>0 1</td><td>1</td></tr><tr><td>1 0</td><td>1</td></tr><tr><td>1 1</td><td>0</td></tr></table></td></tr>
<tr><td>OR</td><td>1 if at least one bit is 1</td><td><table><tr><th>Input</th><th>Output</th></tr><tr><td>0 0</td><td>0</td></tr><tr><td>0 1</td><td>1</td></tr><tr><td>1 0</td><td>1</td></tr><tr><td>1 1</td><td>1</td></tr></table></td></tr>
<tr><td>NOR</td><td>0 if at least one bit is 1</td><td><table><tr><th>Input</th><th>Output</th></tr><tr><td>0 0</td><td>1</td></tr><tr><td>0 1</td><td>0</td></tr><tr><td>1 0</td><td>0</td></tr><tr><td>1 1</td><td>0</td></tr></table></td></tr>
<tr><td>XOR</td><td>1 if exactly one bit is 1</td><td><table><tr><th>Input</th><th>Output</th></tr><tr><td>0 0</td><td>0</td></tr><tr><td>0 1</td><td>1</td></tr><tr><td>1 0</td><td>1</td></tr><tr><td>1 1</td><td>0</td></tr></table></td></tr>
</table>

Table 1. These are classical logic gates. There are only two 1-bit gates (Identity and NOT), and several 2-bit gates

Exercise for the reader: Show that NAND can be used to generate NOT. What is the minimum number of transistors needed to implement NOT using only NANDs?

Quantum Gates

A universal quantum computer must have some sort of analogous gates. It turns out that the quantum superposition of the qubit opens up many more possibilities in terms of what a gate can look like. In Table 2, we outline a few of the most common quantum gates.

Recall that a qubit can be visualized on the Bloch sphere. In this visualization, a quantum gate that acts on a single qubit is a rotation about an axis on the Bloch sphere. For example, the X gate, which is the quantum analog to the classical NOT gate, rotates any state by 180° around the x-axis. So the X gate rotates $|0\rangle$ to $|1\rangle$, and it rotates $|1\rangle$ to $|0\rangle$, i.e. $X|0\rangle = |1\rangle$ and $X|1\rangle = |0\rangle$. But we can also apply the X gate to any arbitrary state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$:

$$\begin{aligned} X|\psi\rangle &= X\big(\alpha|0\rangle + \beta|1\rangle\big) && (20)\\ &= \alpha X|0\rangle + \beta X|1\rangle && (21)\\ &= \alpha|1\rangle + \beta|0\rangle && (22)\\ &= \beta|0\rangle + \alpha|1\rangle && (23) \end{aligned}$$

We can similarly define a Y gate and a Z gate as 180° rotations around their respective axes. We can also have gates which rotate states around other axes. For instance, the Hadamard gate rotates states by 180° around the axis which lies diagonally in the xz-plane. The Hadamard gate turns out to be very useful for entangling qubits, since it maps each of $|0\rangle$ and $|1\rangle$ to equal superpositions of $|0\rangle$ and $|1\rangle$. Quantum gates can also rotate states by an arbitrary amount of degrees; it does not have to be 180°. So we can see that by moving from a discrete bit to a continuous qubit, we expand our single-(qu)bit gate set from size 2 to infinitely large.

An important difference between classical and quantum two (qu)bit gates is that a classical two bit gate takes in two bits of information and outputs one bit of information, whereas a quantum two-qubit gate takes in two qubits and outputs two qubits. We will not use two-qubit gates in this thesis, but it is interesting to point out this difference. On a practical level, this means that running a quantum algorithm, which is a series of quantum gates applied to a quantum processor of qubits, does not change the number of qubits of information.

The most common two-qubit gate is the controlled-NOT gate, defined by:

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{24}$$

The CNOT gate applies an X gate to the second qubit if the first qubit is in the state $|1\rangle$. We can use a CNOT gate with a Hadamard gate to generate entanglement between qubits. Consider two qubits, both prepared in the $|0\rangle$ state. Our overall system state is $|\psi\rangle = |0\rangle_1 |0\rangle_2$. Let's apply a Hadamard gate to the first qubit. We can represent the operation at $H \otimes I$ (applying a Hadamard to the first qubit and the identity to the second qubit):

$$\begin{aligned}
(H \otimes I) |\psi\rangle &= (H |0\rangle_1) |0\rangle_2 && (25)\\
&= \Big(\frac{1}{\sqrt{2}}(|0\rangle_1 + |1\rangle_1)\Big) |0\rangle_2 && (26)\\
&= \frac{1}{\sqrt{2}}(|0\rangle_1 |0\rangle_2 + |1\rangle_1 |0\rangle_2) && (27)\\
& && (28)
\end{aligned}$$

Now let's apply the CNOT gate:

$$\text{CNOT}\frac{1}{\sqrt{2}}(|0\rangle_1 |0\rangle_2 + |1\rangle_1 |0\rangle_2) = \frac{1}{\sqrt{2}}(\text{CNOT}(|0\rangle_1 |0\rangle_2) + \text{CNOT}(|1\rangle_1 |0\rangle_2)) \tag{29}$$

$$= \frac{1}{\sqrt{2}}(|0\rangle_1 |0\rangle_2 + |1\rangle_1 |1\rangle_2)) \tag{30}$$

The resultant state is an equal superposition of both qubits being $|0\rangle$ and both qubits being $|1\rangle$. This is an entangled state, because the act of measuring one qubit's state will collapse the other qubits state. For instance, if we measure one qubit to be $|0\rangle$, then measuring the other qubit will yield $|0\rangle$ with 100% probability, since the wave function collapsed to $|0\rangle_1 |0\rangle_2$.

Quantum Gates

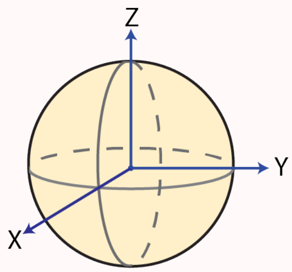
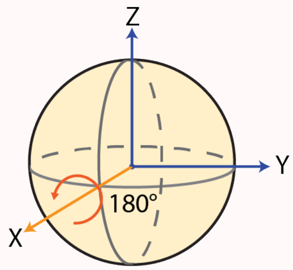
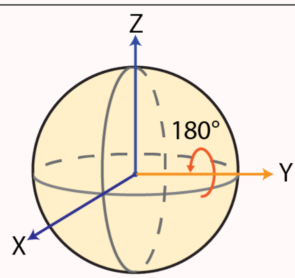
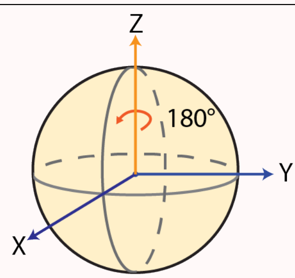
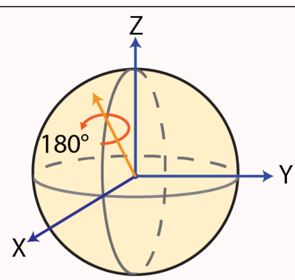
<table>
<tr><th>Gate</th><th>Matrix Representation</th><th>Description</th><th>Bloch Sphere Visualization</th><th>Truth Table</th></tr>
<tr><td>Identity (I)</td><td>$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$</td><td>No action</td><td>
</td><td><table><tr><th>Input</th><th>Output</th></tr><tr><td>$\lvert 0\rangle$</td><td>$\lvert 0\rangle$</td></tr><tr><td>$\lvert 1\rangle$</td><td>$\lvert 1\rangle$</td></tr></table></td></tr>
<tr><td>X Gate</td><td>$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$</td><td>Rotate by 180° around x-axis</td><td>
</td><td><table><tr><th>Input</th><th>Output</th></tr><tr><td>$\lvert 0\rangle$</td><td>$\lvert 1\rangle$</td></tr><tr><td>$\lvert 1\rangle$</td><td>$\lvert 0\rangle$</td></tr></table></td></tr>
<tr><td>Y Gate</td><td>$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$</td><td>Rotate by 180° around y-axis</td><td>
</td><td><table><tr><th>Input</th><th>Output</th></tr><tr><td>$\lvert 0\rangle$</td><td>$i\lvert 1\rangle$</td></tr><tr><td>$\lvert 1\rangle$</td><td>$-i\lvert 0\rangle$</td></tr></table></td></tr>
<tr><td>Z Gate</td><td>$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$</td><td>Rotate by 180° around z-axis</td><td>
</td><td><table><tr><th>Input</th><th>Output</th></tr><tr><td>$\lvert 0\rangle$</td><td>$\lvert 0\rangle$</td></tr><tr><td>$\lvert 1\rangle$</td><td>$-\lvert 1\rangle$</td></tr></table></td></tr>
<tr><td>Hadamard Gate (H)</td><td>$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$</td><td>Rotate by 180° around the axis diagonal in the xz-plane</td><td>
</td><td><table><tr><th>Input</th><th>Output</th></tr><tr><td>$\lvert 0\rangle$</td><td>$\frac{1}{\sqrt{2}}(\lvert 0\rangle + \lvert 1\rangle)$</td></tr><tr><td>$\lvert 1\rangle$</td><td>$\frac{1}{\sqrt{2}}(\lvert 0\rangle - \lvert 1\rangle)$</td></tr></table></td></tr>
</table>

Table 2. There are several single-qubit gates. Five of the most common ones are shown here

Example Gates

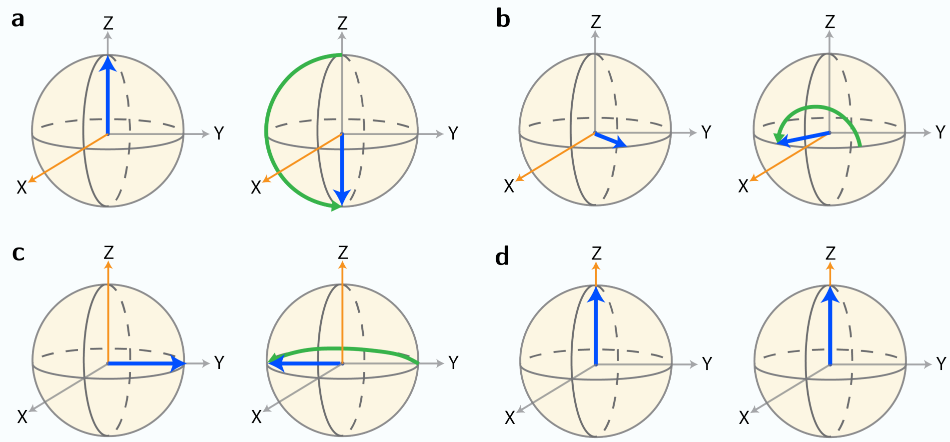


Fig. 5. Examples of rotations on the Bloch sphere.

In general, the statement that a quantum computer is universal is akin to the statement that it can perform any unitary operation (see Appendix, Postulate 5 of Quantum Mechanics). In terms of gates, this can be decomposed in several ways mathematically, for example as $\{R_x(\theta), R_y(\theta), R_z(\theta), P(\phi),$ CNOT$\}$ which encompasses single-qubit rotations about any of the Bloch sphere axes, single-qubit phase shifts, and the two-qubit CNOT. In practical implementations, this is usually summarized as arbitrary single-qubit unitaries coupled with an entangling gate.

Representation

In the same way that digital circuits are represented diagrammatically as bits feeding into a sequence of gates, quantum computing has developed a similar diagrammatic shorthand for representing operations on qubits. Each qubit is associated with a rail, connected and operated on by a series of qubit gates. The order of operations goes from left to right.

For example, mapping these symbols and rules to the gate sequence described in the previous section, we obtain the following circuit correspond to Bell state preparation.

Bell state preparation circuit

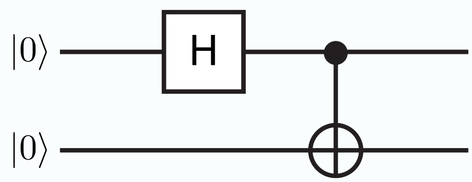


Fig. 6. Hadamard followed by a CNOT gate. This is the Bell state preparation circuit used to generate entanglement.

Bell state measurement circuit

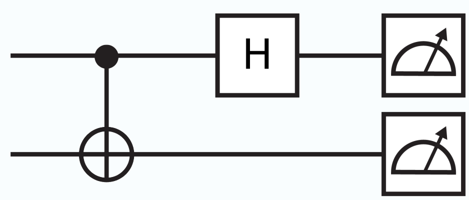


Fig. 7. CNOT gate followed by a Hadamard and measurement symbols. This is a Bell state circuit used to make a measurement.

Bibliography