QCSG

The Quantum Computing Survival Guide

Written by Zack Schieberl

Chapter 1: Why Quantum Computing?

This entire document is used to describe what quantum computing is, but I think it is important to understand why quantum computers are important. What is their history and what can they do for society?

1.1 History of Quantum Computers

idk what it is lol

1.2 The New Age of Computing

Comparing classical computers to quantum. Moore's Law

1.3 Modern Applications of Quantum Computers

cool stuff

1.4 Future Applications of Quantum Computers

super cool stuff

Chapter 2: Qubits & Superposition

The smallest possible unit of data in a classical computer is a *bit*. It stands for binary digit. A single bit can either be zero or one. The term 'bit' is often used interchangeably with *boolean*, whose value is either true or false. Every piece of data which can be stored in a computer's memory is a sequence of these bits. Qubits are the smallest possible unit of data in a quantum computer and they have more properties than ordinary bits.

2.1 Quantum Particles

Although this document is focused towards computer scientists, it is important to learn about the nature of quantum particles to understand where qubits come from. This explanation will only scape the surface of the wonderful world of quantum particles, but it will suffice for the purposes of this document.

Before discussing quantum particles, let's talk waves. A *wave* is any mathematical function which continues to repeat itself, forever. Think about the waves at the beach, how they go up and down as far as you can see. The distance until the wave starts to repeat itself is called the wave's *period*.

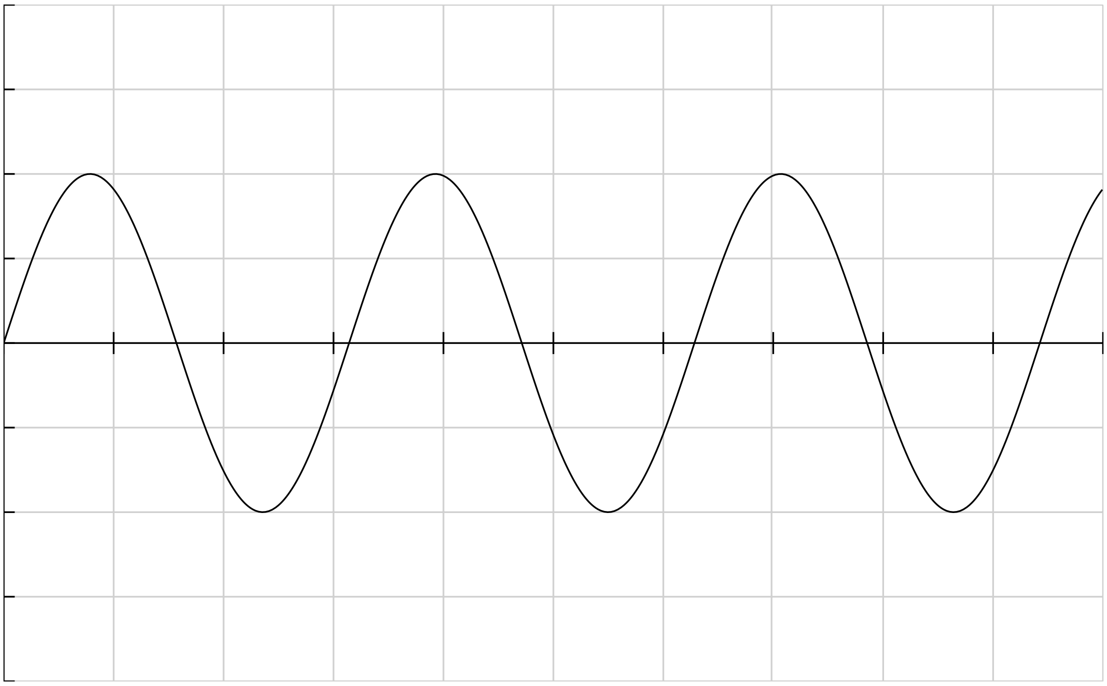

Figure 2.1.1 The Sine Wave

Several famous experiments, such as the double slit experiment, have shown that light and matter can have properties of particles and waves. This is known as *wave/particle duality*. When an electron is released from a source, it acts as a wave which propagates through space until contacting a surface. Once the electron collides with a surface, it is only observable in one location.

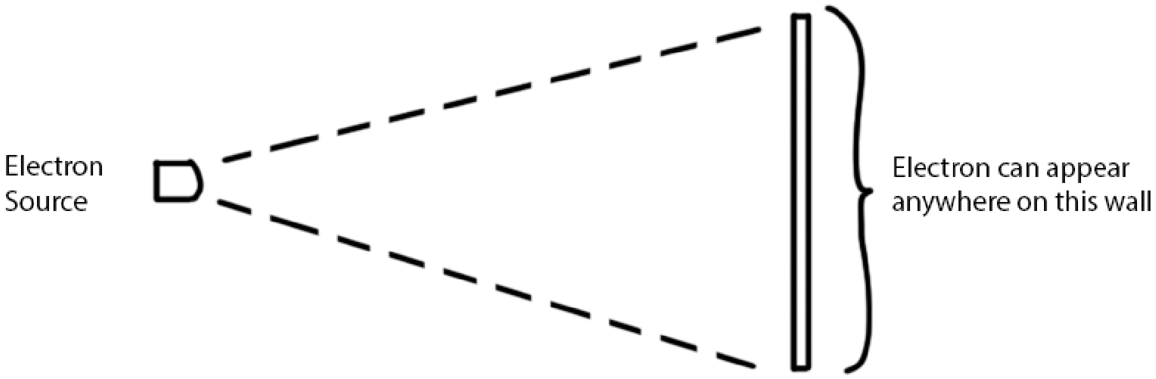

Figure 2.1.2 - Electron Colliding with a Wall in a Random Location

The location where it will be observable is random and cannot be determined ahead of time. However, the likelihood of the electron appearing in a specific location can be calculated using a probability function.

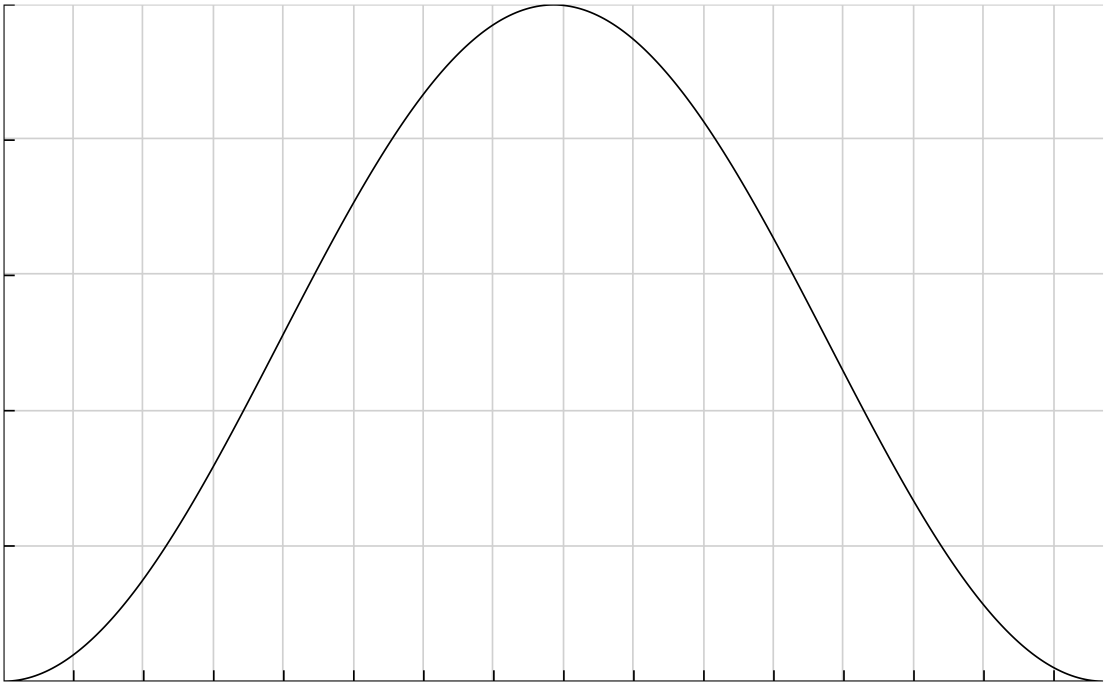

Figure 2.1.3 - A Probability Function

This function shown above contains information on how likely the electron is to be observed at a specified location. In this graph, the X axis is the location and the Y axis is how likely it is to appear at that location. The electron is very likely to appear somewhere in the middle and it is very unlikely to appear near the edges. In the bottom left corner of this graph, when $x = 0$ it shows that $y = 0$ as well. This means the electron can **never** appear at location 0. The same logic applies at the bottom right corner.

The wave equation is a model for how waves move through space. It is a function of two variables, location and time: $F(x, t)$. At the moment the electron is observed, time is no longer a variable and is held constant since it does not change. In quantum computing, how the wave moves is not a concern of ours, so we do not use time as a variable.

This section is self contained, the information presented here isn't used in the following chapters. It is included to provide a foundation for where qubits come from. The rest of this document is dedicated towards explaining the rules governing qubits and their interactions, this is the only section which makes an attempt at showing where the rules are derived from. An inquiring mind may ask "why do qubits follow these rules specifically?" and the only answer is because they follow the laws of quantum mechanics.

2.2 Modeling Waves with Vectors

The fundamental unit of information in classical computers are bits. They can either be on or off. There are only two possible states. Figure 2.1.2 shows a probability function which is continuous, there are an infinite number of locations, each with their own probability of being observed at said location. To make these probability functions palatable to computer scientists, the probability functions need to be transformed into binary.

There is no particular reason why we must make the continuous function binary, we could divide it into 3 sections and transformed into a *trit*, the fundamental unit of information with three states. We could also divide it into 4, 5, 10, 29, 7326, or any number of discrete sections. We could even leave it continuous, without ever dividing it into sections. The only reason for dividing the function into two sections is because there are already many decades worth of development in computer science which is in binary. Having two states allows quantum computer scientists to create algorithms, build architectures, and store data based on classical computers without factoring in a change of radix.

To extend our example from the previous section, we can convert to binary by drawing a line across our wall to see if the electron is observed above or below the line. The electron has a probability α of being above the line where α is $0 \leq \alpha \leq 1$. There is also a probability of $\beta = 1 - \alpha$ that the electron is below the line where β is also $0 \leq \beta \leq 1$.

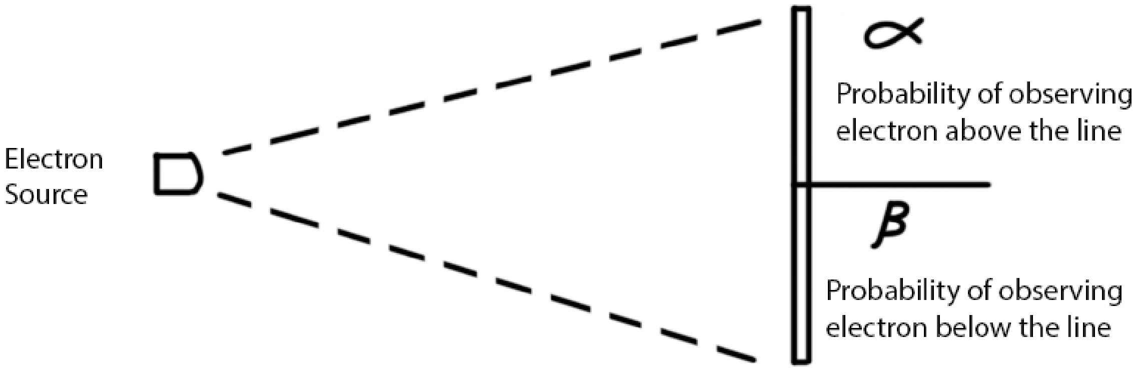
Electron Source
α
Probability of observing electron above the line
β
Probability of observing electron below the line

Figure 2.2.1 - Electron Colliding with a Discretely Sectioned Wall

What we have just described is known as a *probabilistic bit*, or *pbit*. A pbit is a bit which has a certain probability or being off, which we have denoted above as α. It also has a certain probability of being on, which we have denoted above as β. As both these values are probabilities, they are constrained to $\alpha + \beta = 1$. The concept of a pbit is not exclusive to quantum, they also are used in binary classifiers from machine learning, random number generators, etc.

Our goal is to convert a wave equation like the sine function into a pbit. One of the first things a precalculus course teaches is the relationship between trigonometric functions and circles. Taking a look at the unit circle, we can see the X and Y coordinates of the circle with radius 1 are the outputs of the cosine and sine functions.

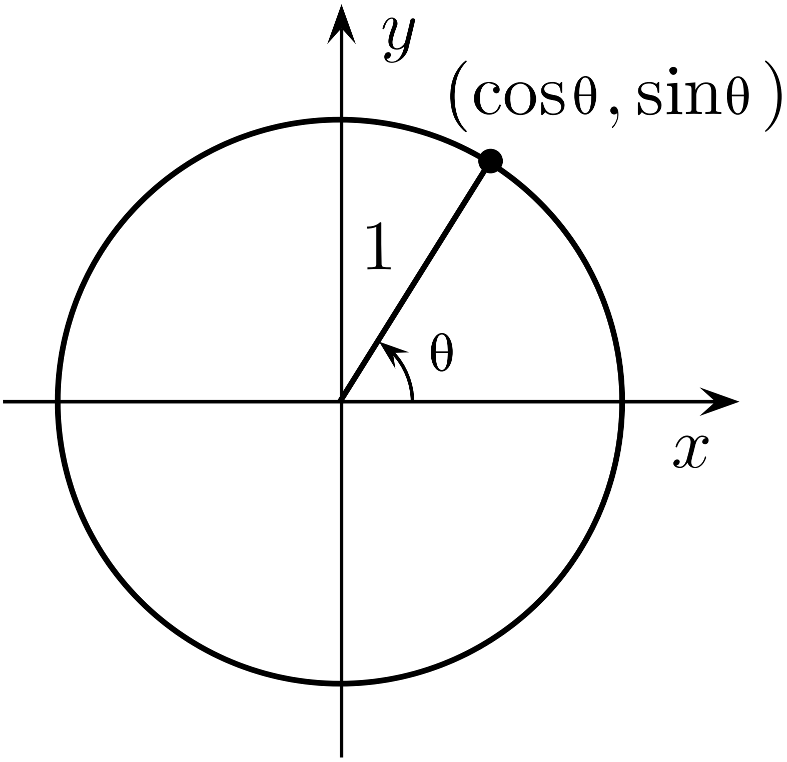

Figure 2.2.2 - The Unit Circle

A *vector* is a mathematical object which has both magnitude and direction. We have a vector here whose magnitude is 1 and direction is determined by the variable θ, the angle from the positive X axis to the vector. θ is called a *polar coordinate* which is the input to the trigonometric functions that produce the cartesian coordinates x and y. The vector $\mathbf{v}$ is described as:

$$\mathbf{v} = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} cos\ \theta \\ sin\ \theta \end{bmatrix}$$

This can be used to fit our description of a pbit by remembering the Pythagorean trigonometric identity:

$$\begin{aligned} x^2 + y^2 &= 1 \\ cos^2\theta + sin^2\theta &= 1 \\ \alpha + \beta &= 1 \end{aligned}$$

From this we can see our probabilistic bit relate to the quantum wave functions as:

$$\begin{aligned} \alpha &= cos^2\theta \\ \beta &= sin^2\theta \end{aligned}$$

Since α and β are always positive, we can apply the constraint $0 \leq \theta \leq \frac{\pi}{2}$.

There are two unit vectors which are special enough to be named: $\hat{\imath}$ and $\hat{\jmath}$. These vectors are both of length 1 and are orthogonal to each other.

$$\begin{aligned} \hat{\imath} &= \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ \hat{\jmath} &= \begin{bmatrix} 0 \\ 1 \end{bmatrix} \end{aligned}$$

These *basis vectors* represent our X and Y axes since $\hat{\imath}$ is directly along the X axis and $\hat{\jmath}$ is directly along the Y axis. The vector $\mathbf{v}$ can be rewritten using vector addition:

$$\mathbf{v} = \begin{bmatrix}\alpha \\ \beta\end{bmatrix} = \alpha\begin{bmatrix}1 \\ 0\end{bmatrix} + \beta\begin{bmatrix}0 \\ 1\end{bmatrix} = \alpha\hat{\imath} + \beta\hat{\jmath}$$

After viewing this relationship, we see:

- $\hat{\imath}$ conceptually represents the off state, or 0, or the electron being observed above the line on the wall
- $\hat{\jmath}$ conceptually represents the on state, or 1, or the electron being observed below the line on the wall
- α and β still represent the probabilities of those states occurring
- $\mathbf{v}$ represents the entire state of the pbit, or the electron & wall system

This model of a pbit is not a completely new innovation, it is simply a transformation from the quantum wave equation which is easier to digest for computer scientists.

However, our transformation is not complete. We have translated the probabilistic nature of quantum particles but there is still another feature of those particles which we have not translated: phase. Without including phase, we cannot say our model acts as a true quantum bit, also known as qubit.

Before extending our vector based model to the true qubit model, a new style of notation needs to be introduced. The notation of quantum physicists: the Dirac notation.

2.3 Dirac Notation

Paul Dirac invented a notation system for discussing quantum mechanics and it has been adopted into quantum computing. The notation is called "bra-ket" since it uses the angle brackets $\langle\rangle$. Truly the peak of twentieth century physics humor.

Any vector can be expressed by placing it inside a vertical line (|) and right angle bracket ($\rangle$). By placing it inside, $|\mathbf{v}\rangle$, we pronounce it "ket-v". This object conceptually represents a quantum state, however it is still mathematically equivalent to a vector. Specifically, a quantum state is a member of a *complex Hilbert space*, but thinking of them as vectors is acceptable in quantum computing.

Since a ket represents a quantum state, we can put our own quantum states inside the ket like: |above-line$\rangle$ and |below-line$\rangle$ to represent the electron being observed above or below the line on the wall. Some more examples of kets are:

- |spin-up$\rangle$ and |spin-down$\rangle$ to represent the spin of a photon
- |ground$\rangle$ and |excited$\rangle$ to represent how close an electron is to its nucleus
- |/$\rangle$ and |\$\rangle$ to represent diagonal and anti-diagonal polarization of light
- |clockwise$\rangle$ and |counterclockwise$\rangle$ to represent direction of current flow in a circuit

However by far the most common kets used in this document and in the quantum computing community are |0$\rangle$ and |1$\rangle$. We can abstract away the physics used for the quantum state and focus on the value. Just like how computer scientists don't care if their bit is representing a magnetic field in their hard drive or amplitude of current in a wire, they just care about the value. |0$\rangle$ and |1$\rangle$ can be any two orthogonal basis states in quantum mechanics.

The quantum state representing an entire system is denoted by the special Greek letter ψ, written "Psi", and pronounced "Sai". Using all of this information, we can rewrite our pbit model using Dirac notation:

$$\begin{aligned} Vector\ Notation \quad & \mathbf{v} = \alpha\hat{\imath} + \beta\hat{\jmath} \\ Dirac\ Notation \quad & |\psi\rangle = \alpha|0\rangle + \beta|1\rangle \end{aligned}$$

Even though we have not extended the pbit to the full qubit model, the above equation still holds for qubits. This equation is called **The General Form of the Qubit** and is one of the most important concepts in all of quantum computing.

Using our new form of notation, let's reason about the process of projecting an electron at a wall. If the electron is detected above the line, then we know $|\psi\rangle = |0\rangle$. This is the same as saying "the state of the electron is the state of being above the line". On the other hand, if the electron is detected below the line, we know $|\psi\rangle = |1\rangle$. This is the same as saying "the state of the electron is the state of being below the line".

However, what do we call the state before the electron contacts the wall? The electron is still a wave moving through space. We do not know if the electron will be observed above or below the line since they are random, but we know the likelihood of those events occurring $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$. Until the electron makes contact with the wall, it is reasonable to say "the state of the electron is both the state of being above and below the line". In other words, the electron's state is a combination of the two outcome states. When a quantum state can only be described as a combination of two or more basis states, we say the quantum state is in *superposition*.

Superposition is one of those words which people think is more magical than it really is. Flipping a coin can be described with superposition too. When the coin is in the air, before it lands, the state can be modeled as: $|coin\rangle = P|heads\rangle + Q|tails\rangle$ where P and Q are the probabilities for the events. When this coin is twirling around in the air, you have no idea which side it will land on. At this moment in time, it is reasonable to say the coin is a combination of heads and tails. The coin is in superposition. To a human, the outcome of a coin flip is a random event. The only difference between the coin and the electron is if you have a good enough computer, it could simulate the physics of the coin toss and determine the outcome before the coin landed whereas the outcome of the electron cannot be determined before it lands.

When the electron collides with the wall, it is observed. The moment the electron is observed, the state goes from being in superposition to a basis state. This process is called *collapsing* and is instantaneous. The probabilities α and β and destroyed and only the state $|0\rangle$ or $|1\rangle$ is left. Colliding with a wall is not the only way to collapse a quantum state. Making any sort of measurement or observation on the system will cause the superposition to be destroyed and the system will choose one of the basis states to collapse to using the assigned probabilities.

2.4 Bloch Sphere Model

pure states

mixed states

2.5 Transforming a Qubit

Rotation around axis -> new angles

new angles -> rotations around axis(es)

2.6 Sequential Transformations

Any transformation from one coord to another can be accomplished in 2 rotations

You only need 2 axes to rotate to any position: XZ, YZ, or XY

2.7 Beam Splitter

2.8 Qubit Implementations

superconductors

ion trap

photon

more about this in quantum architecture

Chapter 3: Quantum Circuits

yum

3.1 Definition of a Gate

boolean function

classical gates

matrix

reversible

universal set of gates

3.2 Pauli Gates

single qubit rotations

their matrices

half a bit flip (both x and y)

3.3 Named Gates

Hadamard, S, T

3.4 Quantum Registers

3.5 CNOT Gate

Interact with multiple qubits

Swap = 3 cnots

3.6 Controlled-U Gate

Any controlled gate!

Chapter 4: Entanglement

swoosh

4.1 The Tensor Product

distributing vectors and distributing kets

4.2 Superposition of Quantum Registers

4.3 Generating Entanglement

undistributable

EPR pair

information instantly

4.4 Bell States

hadamard -> CNOT

4 bell states

read 1 then know the other instantly

Chapter 5: Quantum Networks

it cool

5.1 No-Cloning Theorem

Heisenberg uncertainty principal

5.2 Teleportation

not like the movies

moving a qubit state over a long distance

distributing quantum pairs ahead of time

decoherence

5.3 Superdense Coding

teleportation in reverse

2 bits for the price of 1

Chapter 6: Quantum Cryptography

Chapter 7: Quantum Error Correction

Chapter 8: Quantum Algorithms

QFT, Shors, Grovers, Dueshs Jogns Problem, Simons? Problem,

Chapter 9: Quantum Computer Architecture

Chapter 10: Quantum Compilers

phase normal form

Chapter 11: Simulating Quantum Computers

classically!

matrix form

algorithm form: bitwise CNOT & bitwise X&Y&Z

qiskit universal gate form

Chapter 12: Quantum Artificial Intelligence

Chapter 13: Chemical Simulation