Extracted from:

Quantum Computing

Program Next-Gen Computers for Hard, Real-World Applications

The Pragmatic Bookshelf

Raleigh, North Carolina

The Pragmatic Programmers

Quantum Computing

Program Next-Gen Computers for Hard, Real-World Applications

Nihal Mehta, Ph.D.
edited by Brian MacDonald

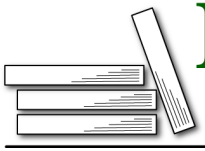
Quantum Computing

Program Next-Gen Computers for Hard, Real-World Applications

Nihal Mehta, Ph.D.

The Pragmatic Bookshelf

Raleigh, North Carolina



Our Pragmatic books, screencasts, and audio books can help you and your team create better software and have more fun. Visit us at *https://pragprog.com*.

The team that produced this book includes:

Publisher: Andy Hunt
VP of Operations: Janet Furlow
Executive Editor: Dave Rankin
Development Editor: Brian MacDonald
Copy Editor: L. Sakhi MacMillan
Indexing: Potomac Indexing, LLC
Layout: Gilson Graphics

For sales, volume licensing, and support, please contact *support@pragprog.com*.

For international rights, please contact *rights@pragprog.com*.

ISBN-13: 978-1-68050-720-1
Encoded using the finest acid-free high-entropy binary digits.
Book version: P1.0—August 2020

Modeling Quantum Bits with the Qubelets Model

Quantum bits or *qubits* are the workhorses of quantum programming. They're governed by quantum mechanical principles that make it seem as if they're oscillating between the two binary states. As a result, we can make the qubits interact with each other in ways that let us, for example, formulate decision-making logic circuits that boost their runtime performance or encrypt data in virtually unbreakable ways. Thus, to become proficient at writing quantum programs, we need to first understand how to control these qubits.

Classical bits are binary; you'll only find them in one of two well-defined states, `1` or `0`. Quantum bits, or qubits, also have two well-defined states corresponding to the classical binary states `0` and `1`. Unlike classical binary bits, though, qubits can be in a *blended* state that is a combination of these two states, such as the coin toss alluded to earlier. This state isn't an average like 0.5. Rather, it's a concept of both states existing at the same time.

To remind us that qubits exist in a blended state, they're decorated with extra symbols to distinguish them from standard binary bits. In particular, we'll label the idealized states as $|0\rangle$ and $|1\rangle$, respectively.

Bra-Ket Notation

The notation to denote quantum states was devised by Paul Dirac, one of the founders of quantum mechanics. In this chapter, we use the *ket*, $|q\rangle$, to represent the quantum bit `q`.

We'll introduce its partner, the *bra*, $\langle 0|$, in Can the Quantum Gate Matrix Be Anything?, on page ?.

You'll hear words such as "blended," "combination," and "superposition" used to describe this state, but they're not to be taken literally. The English language doesn't have good terms for quantum concepts, so we have to resort to approximations.

Readers familiar with quantum mechanics will recognize that the blended states are just another version of Schrödinger's cat, a famous thought experiment illustrating being in two states at the same time. You can learn more about this concept here.[1]

To bring quantum computing into sharper focus and help us recognize its potential to solve some of the most complex computational tasks of today, we first upgrade the coin-toss analogy to a more nuanced model. In this model,

1. https://gizmodo.com/breakthrough-quantum-cat-experiment-captured-on-camera-1786923180

we reinterpret qubits and their quantum states as a collection of imaginary particles that we call *qubelets*. These qubelets are not physical subatomic particles and can't be isolated from a qubit. They're merely a figment of our imagination that gives us something concrete to hold onto when talking about nebulous quantum concepts. Over the subsequent chapters, we'll work with this model to get to the core of quantum computing concepts and intuitively understand them. With this foundation, you'll learn to harness quantum principles to design quantum programs for your own applications instead of blindly running well-known algorithms in the literature.

The Qubelets Model Is Unique to This Book

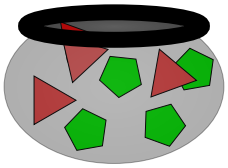
Quantum mechanics is shrouded under dense mathematics that even experts find challenging. As a result, it's hard to develop an intuition for the central concepts related to quantum computing and how they come into play for solving computational problems. To this end, I developed the *Qubelets Model* as a way to bring home the quantum ideas in a form that computer professionals would find familiar.

The Qubelets Model, though, is not just a superficial knock-off of quantum mechanics that loses steam as we get deeper into the subject. We'll ride this model to the crux of quantum computing and explore real-life quantum phenomena with actual programs. This will help build your intuition so that you can design quantum algorithms for your own complex applications. In later chapters, we'll tie this model with the mathematical theory of quantum mechanics and establish that this model is a valid way to think about quantum computing.

In Chapter 3, Elementary, My Dear Watson—Quantum Logic, on page ?, through Chapter 5, Beam Me Up, Scotty—Quantum Tagging and Entangling, on page ?, I'll introduce quantum concepts by extending familiar constructs from classical computing. You'll learn the statements to initiate quantum effects in your programs and do basic but meaningful computational work, with a minimal amount of mathematics, that you can't reproduce on classical computers.

To push quantum computers to take on more complex computational problems, such as those described in Chapter 9, Alice in Quantumland—Quantum Cryptography, on page ?, and Chapter 10, Quantum Search, on page ?, you'll learn new ways to manipulate quantum bits in Chapter 6, Designer Genes—Custom Quantum States, on page ?, and Chapter 7, Small Step for Man—Single Qubit Programs, on page ?. Alhough these chapters rely on the mathematics that underpins quantum mechanics, I'll relate them back to the no-mathematics concepts introduced earlier so that their motivation is clearer.

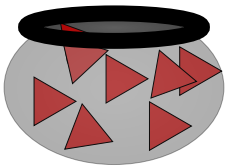
Specifically, imagine a qubit as the brew in a witch's cauldron whose constituents are just two ingredients, pentagons and triangles, that don't dissolve or mix with each other, as shown in the following figure:



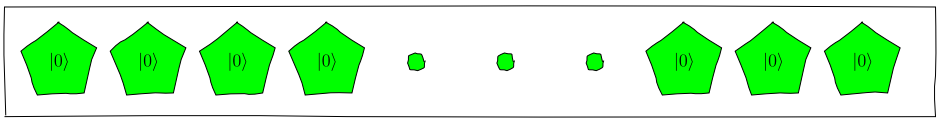
Call these pentagon and triangle shapes *qubelets* or baby qubits:

- The pentagon qubelet is a baby $|0\rangle$ qubit.
- The triangle qubelet is a baby $|1\rangle$ qubit.

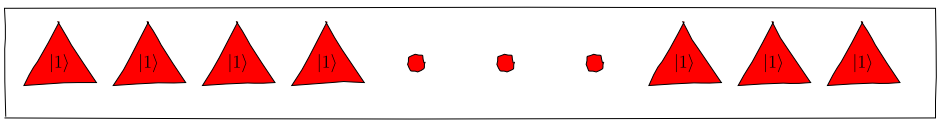
For example, the $|1\rangle$ qubit is a "brew" containing only one ingredient, triangle $|1\rangle$ qubelets:



Going forward, we'll dispense with the cauldron and draw the quantum state as a rectangular box. Thus, the $|0\rangle$ qubit will only consist of pentagon $|0\rangle$ qubelets as shown in the following figure:

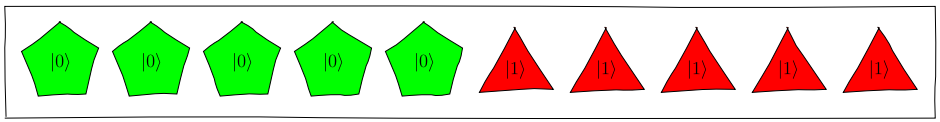


Similarly, the $|1\rangle$ qubit is a bundle of triangle qubelets:



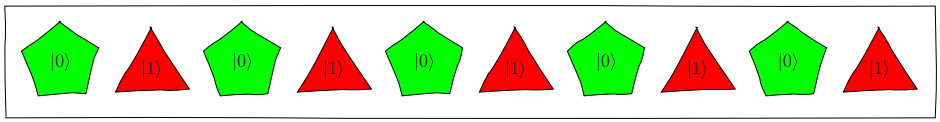
Qubits in Blended States

A qubit in a blended state of $|0\rangle$ and $|1\rangle$ is shown as follows:



Here, there are an equal number of pentagon and triangle qubelets, indicating that the qubit is equally likely to collapse to `1` or `0`.

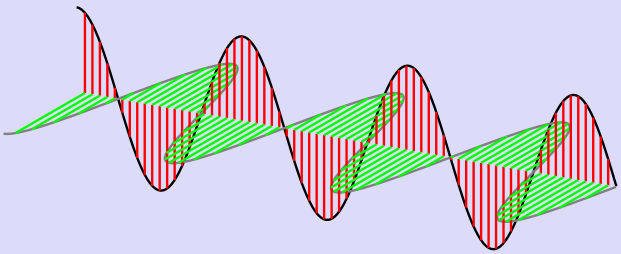
We could also show the pentagon and triangle qubelets alternating:



In this case, you can think of the subatomic particle whirling through the $|1\rangle$ and $|0\rangle$ states like the faces of a spinning coin. Feel free to use whichever format you prefer. The key principle to bear in mind is that it's only the relative number of pentagon and triangle qubelets that's important, not their order or how they're arranged.

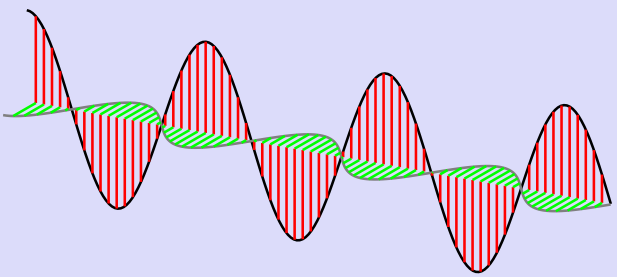
Qubelets and Polarized Light

Another way to think about qubelets and blended states is a beam of light made up of two polarized waves—one traveling in the horizontal plane and the other in the vertical plane:



The vertical wave is made up of triangle $|1\rangle$ qubelets, and the horizontal wave corresponds to pentagon $|0\rangle$ qubelets.

In each polarized wave of light, we may associate the maximum amplitude of a wave with the corresponding number of qubelets: the more qubelets of a particular type, the greater the amplitude. For example, a quantum state that has fewer pentagon $|0\rangle$ qubelets than triangle $|1\rangle$ qubelets would have the following polarized waves in the beam of light:

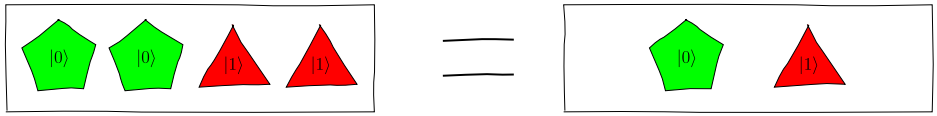


The smaller amplitude of the horizontal polarized wave is a result of there being fewer pentagon $|0\rangle$ qubelets than triangle $|1\rangle$ qubelets in the quantum state.

This view of quantum states, though, fails to explain many of the quantum phenomena that we'll need in quantum computing. But it's a good stand-in till you get comfortable working with qubelets.

Equivalent Qubits

Since it's only the ratio of the number of pentagon |0⟩ to triangle |1⟩ qubelets that matters, the qubit on the left is equivalent to the qubit on the right in the following diagram:



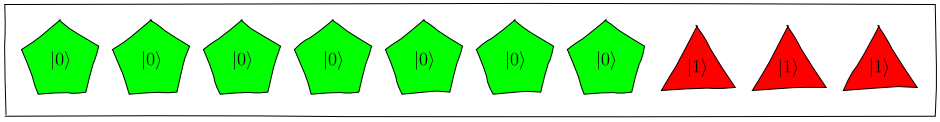
In each of the left and right bundles of qubelets, the number of pentagons and triangles is the same. Thus, the left and right qubit each will collapse to either `1` or `0` with equal probability.

Only the Relative Number of Qubelets Is Important

It's only the proportion of pentagon |0⟩ to triangle |1⟩ qubelets that governs the collapse of the qubit to a classical state—the actual number of pentagon and triangle qubelets doesn't count.

Biased Qubits

The qubits we've seen so far have symmetric quantum states: either all states contain pentagon |0⟩ qubelets, or they have all triangle |1⟩ qubelets, or they have an equal number of pentagon |0⟩ and triangle |1⟩ qubelets. Quantum states, however, can be unbalanced, or *biased*, in which there are more of one type of qubelets than the other. For example, the qubit shown in the following diagram has more pentagon |0⟩ qubelets than triangle |1⟩ qubelets:



So when we talk about qubits, we're actually talking about their quantum states.

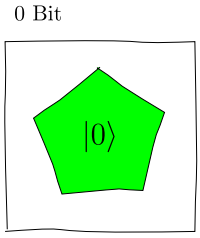
Qubelets Model the Coin in the Air

Going back to the coin-toss analogy, the qubelets model the coin as it's spinning in the air and thus, as you'll see, give a way to influence how it lands.

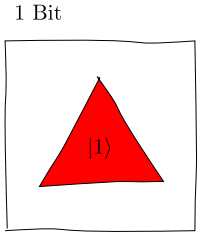
Classical Versus Quantum Bits

Contrasting classical bits with quantum bits is like imagining a world atlas to be like Google Maps—although the paper almanac differs in fundamental

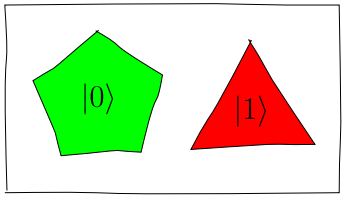
ways from the browser version, you can still make meaningful comparisons that highlight their differences by assuming they work more or less the same. With this in mind, you can think of a 0 classical bit as containing only a single pentagon $|0\rangle$ qubelet, as shown in the following figure:



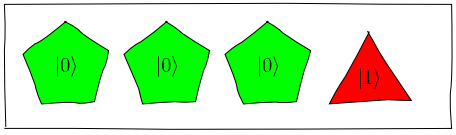
Likewise, a 1 classical bit has just a triangle $|1\rangle$ qubelet:



A quantum bit, as we've just seen, can have both pentagon $|0\rangle$ and triangle $|1\rangle$ qubelets as shown next:



Or it can even be one where the number of pentagon $|0\rangle$ qubelets differs from that of the triangle $|1\rangle$ qubelets:



Moreover, over the course of a program, the number of pentagon $|0\rangle$ and triangle $|1\rangle$ qubelets in a qubit can change—it's not fixed as in classical bits. Simply put, a quantum bit, like Google Maps, is far more versatile than a classical bit, which is more akin to a paper map. We'll see that this added flexibility is largely responsible for giving quantum computers an edge over conventional computers.

Collapsing Qubits

When we want to inspect the state of a classical bit while a program is being executed in a conventional computer program, we simply write out the variable that holds the bit. The act of looking at the bit doesn't affect its state.

In quantum computing, on the other hand, we can never directly observe the quantum state of a qubit: we can't peer into the cauldron to check out a qubit's quantum state—the brew is opaque. The only way to examine the quantum state is to reach into the cauldron and randomly pull out a qubelet. At that point, two things happen:

1. The remaining qubelets in the cauldron fade away, as they are no longer of any use—the laws of quantum mechanics dictate that you can't select another qubelet.

2. The selected qubelet, either a pentagon $|0\rangle$ or a triangle $|1\rangle$, is anointed the state of the qubit and we say that the qubit has *collapsed*. The cauldron is effectively "reset" so that it only contains the selected qubelet.

For example, if a pentagon $|0\rangle$ qubelet is selected, the qubit collapses to the *idealized* quantum state $|0\rangle$ and all the other pentagon $|0\rangle$ and triangle $|1\rangle$ qubelets vanish from the state. And if a triangle $|1\rangle$ qubelet is picked from the cauldron, the qubit collapses to the *idealized* quantum state $|1\rangle$ and the pentagon $|0\rangle$ and other triangle $|1\rangle$ qubelets disappear.

In general, though, we can't conclusively predict which of the two idealized states the quantum state collapses to: sometimes it collapses to $|0\rangle$ and at other times to $|1\rangle$. To put it another way, finding a qubit in a specific state after collapsing it doesn't tell us anything about the relative number of pentagon $|0\rangle$ and triangle $|1\rangle$ qubelets in the quantum state before it collapsed. In particular, if we were to start again from a qubit in an identical quantum state and collapse it, there's no guarantee that the qubit will collapse to the same classical state as the previous selection.

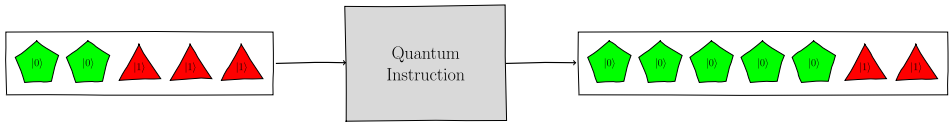
Thus, in quantum computing, as we'll learn in subsequent chapters, all the computational work is done on qubits whose quantum states have not collapsed.

Only after we're convinced that all the qubits hold the optimal or desired states that your program is tasked with finding, do we then collapse them.

What Are Quantum Programs?

A quantum program is a sequence of instructions, or actions, that guides qubits from one quantum state to another such that each qubit arrives at a state that corresponds to the optimal solution.

For example, a quantum instruction changes the quantum state of a qubit, as shown in the following figure:



In this case, the quantum instruction, shown as a box, takes a qubit whose quantum state is shown on the left with two pentagon $|0\rangle$ qubelets and three triangle $|1\rangle$ qubelets to another one shown on the right with five pentagon $|0\rangle$ qubelets and two triangle $|1\rangle$ qubelets. Chaining these quantum instructions together results in a series of steps that varies a qubit's quantum state.

In a quantum program, we apply these instructions to several qubits and modify their quantum states. In subsequent chapters, we'll learn to write the proper sequence of various instructions so as to get the qubits to collapse to the desired binary states.

This graphical depiction of a quantum programming instruction underscores the crucial role that subatomic physics plays in contending with challenging computational problems. In classical computing, each statement acts on binary bits that hold only one value at a time. In contrast, because a qubit in effect has many qubelets, or tiny bits, a quantum instruction juggles with all of them at once. Thus, quantum computing has two features that together offer a powerful break from classical computing:

- A quantum instruction works on all the qubelets in a qubit's quantum state simultaneously.
- As the quantum state of the qubits is altered, the number of pentagon $|0\rangle$ and triangle $|1\rangle$ qubelets in the quantum state can grow or shrink as necessary and isn't limited by any physical constraint.

As a result, because the qubits can hold an exponentially large number of qubelets, or tiny states, even a quantum program with a reasonably small set of qubits can solve extremely large problems. We'll expand on this theme

from Chapter 4, All Together Now—Quantum Superposition, on page ?, onward.

Pentagon and Triangle Shapes Are Visuals for the Idealized Quantum States

The pentagon and triangle shapes have no quantum significance. They are just visuals for the idealized quantum states $|0\rangle$ and $|1\rangle$, respectively, and give a way to pictorially explain the mathematics describing quantum phenomena.