Introduction to Quantum Computing

Lectures by Leon Riesebos

Steven Oud
soud@pm.me

March 21, 2019

Chapter 1

Fundamentals: A One-Qubit World

Quantum computers are machines that rely on characteristically quantum phenomena, such as quantum interference and quantum entanglement, in order to perform computation.

— Artur Ekert

It is tempting to say that a quantum computer is one whose operation is governed by the laws of quantum mechanics. But since the laws of quantum mechanics govern the behavior of all physical phenomena, this temptation must be resisted. Your laptop operates under the laws of quantum mechanics, but it is not a quantum computer.

— N. David Mermin

1.1 The Qubit

A quantum bit (*qubit*) - like a classical bit - has a *state*. In classical bits this is 0 or 1. Two possible states for a qubit are $|0\rangle$ and $|1\rangle$ (denoted in *Dirac notation*), which correspond to the states 0 and 1 for a classical bit. The difference between bits and qubits is that a qubit can be in other states than $|0\rangle$ or $|1\rangle$, called *superpositions*. The state of a qubit can be denoted as following:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, \tag{1.1}$$

where α and β are complex numbers. The special states $|0\rangle$ and $|1\rangle$ are known as *computational basis states*. We cannot examine a qubit to determine its quantum state. Quantum mechanics tells us we can only acquire much more restricted information about the quantum state. When we measure a qubit we get either 0, with probability $|\alpha|^2$, or 1, with probability $|\beta|^2$. The sum of all probabilities is always equal to 1:

$$|\alpha|^2 + |\beta|^2 = 1. \tag{1.2}$$

For example, a qubit in the state

$$\frac{1}{\sqrt{2}}\,|0\rangle + \frac{1}{\sqrt{2}}\,|1\rangle \tag{1.3}$$

gives 0 fifty percent of the time ($|1/\sqrt{2}|^2$), and 1 fifty percent of the time. This state is often denoted by $|+\rangle$.

1.2 The Bloch Sphere

The Bloch sphere is a geometrical representation of a qubit's state. It's a spherical coordinate system in which a quantum state can be described as

$$|\psi\rangle = e^{i\delta}\left(cos\frac{\theta}{2}\,|0\rangle + e^{i\phi}sin\frac{\theta}{2}\,|1\rangle\right), \tag{1.4}$$

where δ, θ and ϕ are real numbers. Factor $e^{i\delta}$ is the global phase of the state. This factor does not influence measurement probabilities, since $\left|e^{i\delta}\right| = 1$. Therefore we can often omit it, allowing us to write

$$|\psi\rangle = cos\frac{\theta}{2}\,|0\rangle + e^{i\phi}sin\frac{\theta}{2}\,|1\rangle\,. \tag{1.5}$$

The numbers θ and ϕ define a point on the three-dimensional sphere (Figure 1.1). The Bloch sphere visualization can be very useful for describing single qubit operations. It is however limited in that there is no simple generalization of the Bloch sphere known for multiple qubits.

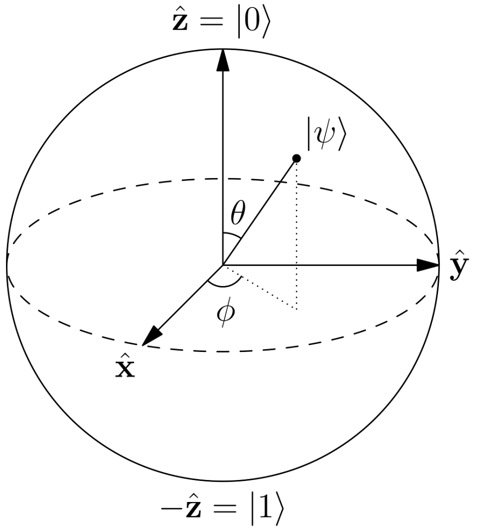


Figure 1.1: *Bloch sphere representation of a qubit.*

1.3 Single-Qubit Quantum Gates

Single-qubit quantum gates can be seen as counter-clockwise rotations on the Bloch sphere. These gates only have one limitation: they have to be *unitary*, that is $U^\dagger U = UU^\dagger = I$, where $U^\dagger$ is the conjugate transpose of U and I the identity operation. Therefore, any $2^n \times 2^n$ unitary operation is a valid gate which acts on n qubits. We will often use circuit notation to describe the transformation of state $|\psi\rangle$ to $|\psi'\rangle$ by a unitary operation U as following:

$$|\psi\rangle \;\; -\boxed{U}- \; |\psi'\rangle$$

Below are some notable single-qubit gates described and visualized.

1.3.1 Pauli Gates

The most simple quantum gates are the *Pauli gates* I, X, Y and Z. I is the identity gate, which does nothing. The other gates rotate π radians (180 degrees) around the X, Y or Z-axis. These gates are self-inverse, meaning $U^2 = I$.

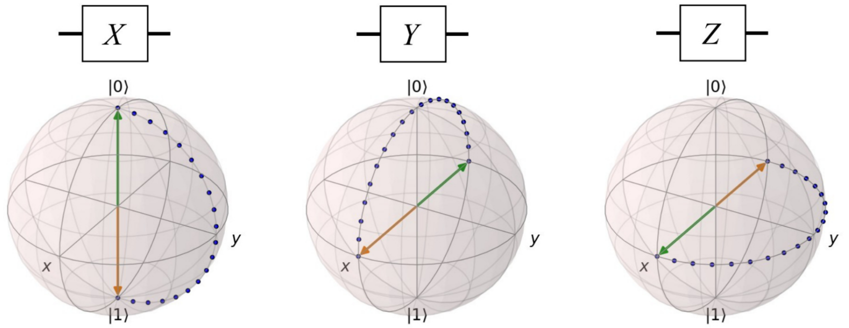


Figure 1.2: *Pauli gates X, Y and Z visualized on the Bloch sphere. The green vector is the starting position and the orange vector the final position.*

1.3.2 Hadamard Gate

The *Hadamard* (H) gate maps the basis states $|0\rangle$ and $|1\rangle$ to superposition states with equal weight. It is the combination of two rotations, π radians about the Z-axis followed by $\pi/2$ radians about the Y-axis.

This gate is sometimes described as a “square root of NOT” gate, because it turns $|0\rangle$ into $(|0\rangle + |1\rangle)/\sqrt{2}$ (also written as $|+\rangle$), halfway between $|0\rangle$ and $|1\rangle$. The Hadamard gate belongs to the Clifford gates.

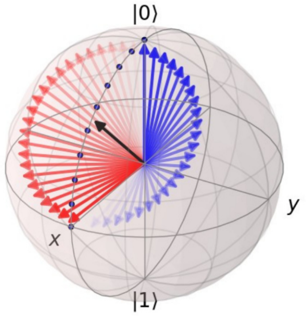


Figure 1.3: *Hadamard gate visualized on the Bloch sphere.*

1.3.3 Phase Gates

Rotation gates that rotate around the Z axis are considered *phase gates*. They rotate the phase of the $|1\rangle$ state by an angle θ and leave the $|0\rangle$ state unchanged:

$$\begin{aligned} R_z(\theta)\,|0\rangle &= |0\rangle \\ R_z(\theta)\,|1\rangle &= e^{i\theta}\,|1\rangle\,. \end{aligned} \tag{1.6}$$

The probability of measuring a $|0\rangle$ or $|1\rangle$ is unchanged after a phase gate, however it modifies the phase of the quantum state. A common phase gate is the *S gate*, where $\theta = \pi/2$ (visualized in Figure 1.4). The Pauli Z gate can be thought of as a phase gate where $\theta = \pi$, because $e^{i\pi} = -1$. Thus you could also think of the S gate as half a Pauli Z gate. Another common phase gate is the *T gate*, where $\theta = \pi/4$ (half an S gate).

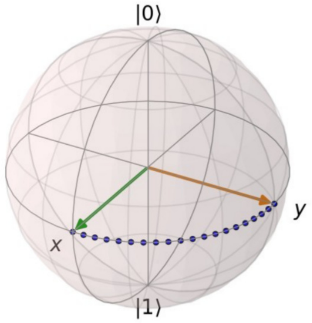


Figure 1.4: *S gate rotation visualized on the Bloch sphere.*

1.4 Measurement

Measuring a quantum state $|\psi\rangle$ *collapses* the quantum superposition to a classical state $|j\rangle$ (which in the computational basis is $|0\rangle$ or $|1\rangle$). The state $|\psi\rangle$ has "disappeared" and all that is left is the classical state $|j\rangle$. The probabilistic result of a measurement of a quantum state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ can be calculated based on the probability amplitudes:

$$\begin{aligned} P(|0\rangle) &= |\alpha|^2 \\ P(|1\rangle) &= |\beta|^2 \end{aligned} \tag{1.7}$$

Consider the following simple single-qubit circuit:

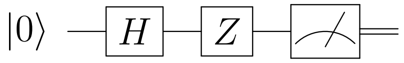


Figure 1.5: *A simple quantum circuit. The output of measuring a qubit is a classical bit, which is distinguished from a qubit by drawing a double-line wire. A visualization of this circuit on the Bloch sphere can be seen in Figure 1.6.*

We start with computing $H|0\rangle = |+\rangle$, followed by $Z|+\rangle = (|0\rangle - |1\rangle)/\sqrt{2}$ (or $|-\rangle$), and finally we measure, giving us a classical state $|j\rangle$ and collapsing the state. Which state we will see is not determined in advance; the only thing we can say is that we will see $|j\rangle$ with probability $|a_j|^2$:

$$\begin{aligned} P(|0\rangle) &= \left|\frac{1}{\sqrt{2}}\right|^2 = \frac{1}{2} \\ P(|1\rangle) &= \left|\frac{-1}{\sqrt{2}}\right|^2 = \frac{1}{2} \end{aligned} \tag{1.8}$$

Giving us equal probabilities of our state being measured as $|0\rangle$ or $|1\rangle$.

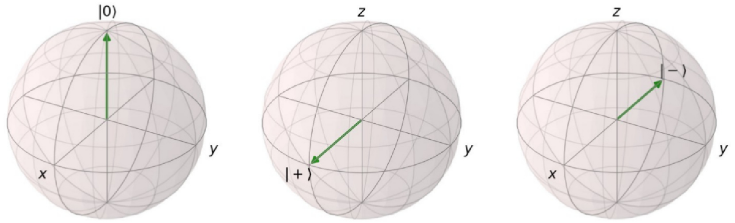


Figure 1.6: *States of our qubit throughout the circuit from left to right: $|0\rangle \rightarrow H|0\rangle \rightarrow ZH|0\rangle$.*

1.5 Vector Notation

Earlier we showed that we can represent a quantum state as

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,. \tag{1.9}$$

The quantum states $|\psi\rangle$, $|0\rangle$ and $|1\rangle$ in the formula above are vectors. We define $|0\rangle$ and $|1\rangle$, the computational basis states, as following:

$$|0\rangle = \begin{pmatrix}1\\0\end{pmatrix}; \quad |1\rangle = \begin{pmatrix}0\\1\end{pmatrix}. \tag{1.10}$$

A quantum state has to be a normalized vector. In general, a qubit's state is a unit vector in a two-dimensional complex vector space. A n qubit state has a 2^n dimensional *Hilbert space*. We can consider a quantum state to be a linear combination of the computational basis states:

$$|\psi\rangle = \alpha\begin{pmatrix}1\\0\end{pmatrix} + \beta\begin{pmatrix}0\\1\end{pmatrix} = \begin{pmatrix}\alpha\\\beta\end{pmatrix}. \tag{1.11}$$

Single-qubit gates can then be represented by 2×2 unitary matrices:

$$I = \begin{pmatrix}1 & 0\\0 & 1\end{pmatrix} \qquad X = \begin{pmatrix}0 & 1\\1 & 0\end{pmatrix} \qquad Y = \begin{pmatrix}0 & -i\\i & 0\end{pmatrix}$$

$$Z = \begin{pmatrix}1 & 0\\0 & -1\end{pmatrix} \qquad S = \begin{pmatrix}1 & 0\\0 & i\end{pmatrix} \qquad H = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\1 & -1\end{pmatrix}$$

Then a computation like $X|0\rangle$ can be calculated by matrix vector multiplication:

$$X\,|0\rangle = \begin{pmatrix}0 & 1\\1 & 0\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix} = \begin{pmatrix}0\\1\end{pmatrix}. \tag{1.12}$$

And we find that $X\,|0\rangle = |1\rangle$. Or, more general:

$$X\begin{pmatrix}\alpha\\\beta\end{pmatrix} = \begin{pmatrix}\beta\\\alpha\end{pmatrix}. \tag{1.13}$$

You can combine quantum gates by multiplying their matrices. For example, let's verify that H is self-inverse by applying it to itself:

$$H^2 = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\1 & -1\end{pmatrix}\frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\1 & -1\end{pmatrix} = \begin{pmatrix}1 & 0\\0 & 1\end{pmatrix} = I. \tag{1.14}$$

The result is the identity gate, showing that the inverse of H is indeed itself. It is also a *Hermitian matrix*, because it is equal to its own conjugate transpose: $H = H^\dagger$.

Chapter 2

Fundamentals: A Multi-Qubit World

We can represent multiple qubit states using the *Kronecker product*. Let A be a $m \times n$ matrix and B a $p \times q$ matrix, then

$$A \otimes B = \begin{pmatrix} a_{11}B & \dots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \dots & a_{mn}B \end{pmatrix}, \tag{2.1}$$

resulting in a $mp \times nq$ matrix. We can then represent the two qubit state $|00\rangle$ as

$$|00\rangle = |0\rangle \otimes |0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ 0 \begin{pmatrix} 1 \\ 0 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}. \tag{2.2}$$

We can write a two qubit state as following in Dirac notation:

$$|ab\rangle = |a\rangle \otimes |b\rangle = \alpha_{00} |00\rangle + \alpha_{01} |01\rangle + \alpha_{10} |10\rangle + \alpha_{11} |11\rangle. \tag{2.3}$$

More generally, we say that a linear combination $\sum_i \alpha_i |\psi_i\rangle$ is a quantum state with states $|\psi_i\rangle$ and amplitude α_i for state $|\psi_i\rangle$. Note that the state vector of two qubits is twice as big as the state vector for one qubit. This is where some of the potential power of quantum computers comes from. With n qubits you can represent 2^n states - the state space grows exponentially with the number of qubits, unlike classical bits. For multi-qubit states the rule remains that the state vector has to be normalized. For a n qubit state:

$$\sum_{i=0}^{2^n-1} |\alpha_i|^2 = 1. \tag{2.4}$$

2.1 Quantum State Evolution

Doing single-qubit operations on a multi-qubit state is possible by combining the identity and our single-qubit gate. Say we want to compute $H_1|0_0 0_1\rangle$. That is, put the second qubit through a Hadamard gate.[1] To do so, the gate matrix's column

[1] Note that when I say "second" qubit, I'm talking about the most right, or least significant qubit. I will number them for this example but assume it from here on out.

width has to be equal to the quantum state vector's dimension. We can achieve this by taking the Kronecker product of the identity matrix and our single-qubit matrix (in this case H):

$$H_1 \left|0_0 0_1\right\rangle = (I_0 \otimes H_1) \left|0_0 0_1\right\rangle . \tag{2.5}$$

Writing it out:

$$I_0 \otimes H_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{2.6}$$

$$= \frac{1}{\sqrt{2}} \left[\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \right] \tag{2.7}$$

$$= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{pmatrix} . \tag{2.8}$$

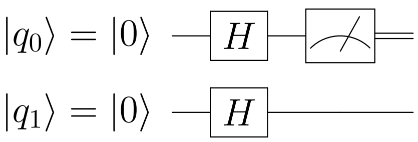
Then we can put our $|00\rangle$ state through it:

$$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} , \tag{2.9}$$

which can also be written in Dirac notation as $\frac{1}{\sqrt{2}}(|00\rangle + |01\rangle)$.

2.2 Partial Measurement

Say we measure the qubit $|q_0\rangle$ in the following circuit:



First, we put both qubits in our state $|00\rangle$ through a Hadamard gate. This puts it in the state

$$(H \otimes H) \left|00\right\rangle = \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle). \tag{2.10}$$

When we measure $|q_0\rangle$ we have a 50/50 probability of getting a 0 or 1. Measuring $|q_0\rangle$ collapses the state to one of the following states:

$$|\psi\rangle = \begin{cases} \frac{1}{\sqrt{2}}(|\underline{0}0\rangle + |\underline{0}1\rangle) \text{ if } M(q_0) = 0 \\ \frac{1}{\sqrt{2}}(|\underline{1}0\rangle + |\underline{1}1\rangle) \text{ if } M(q_0) = 1 \end{cases} \tag{2.11}$$

We have two possible states for q_0 after measurement: 0 or 1. Qubit $|q_1\rangle$ will stay in superposition because we haven't measured it. Notice how the first qubits (underlined) in both states are the same. This makes sense, we've measured that one so we're certain of its state.

2.3 Common Two-Qubit Gates

2.3.1 CNOT Gate

The quantum gate controlled-NOT (CNOT, sometimes called controlled-X) is comparable to a classical computer's XOR, but it's reversible. This gate has two input qubits, the *control* qubit and *target* qubit. If the control qubit is set to 0, then the target qubit is left alone. If the control qubit is set to 1, then the target qubit is flipped. The circuit representation for CNOT can be seen in Figure 2.1. Qubit $|q_0\rangle$ represents the control qubit and $|q_1\rangle$ represents the target qubit. It's essentially a Pauli X gate with a control qubit. CNOT is Hermitian and belongs to the Clifford gates.

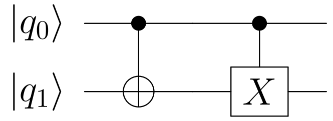
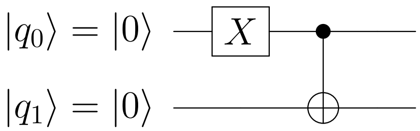


Figure 2.1: *Two different circuit representations of* CNOT*. We will use the left representation.*

$$U_{CN} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Figure 2.2: *Matrix representation of* CNOT.

Let's look at an example of a CNOT gate. Consider the following circuit:



First we put $|q_0\rangle$ (the control qubit) in the $|1\rangle$ state by applying an X gate, giving us the state $|10\rangle$. Then we apply the CNOT gate, giving

$$\text{CNOT}\,|10\rangle = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = |11\rangle\,. \tag{2.12}$$

The target qubit was flipped because the control qubit was set to $|1\rangle$, giving us $|11\rangle$.

2.3.2 CZ Gate

CZ, or the controlled-Z gate, acts in a similar way to other controlled gates. That is, do the operation on the target qubit if the control qubit is $|1\rangle$, otherwise do nothing. In CZ the operation is the Pauli Z gate. CZ is also Hermitian and belongs to the Clifford gates.

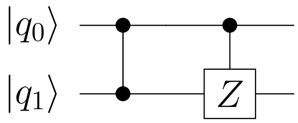


Figure 2.3: *Two different circuit representations of* CZ*.*

2.3.3 Controlled Gates

Controlled gates act on two or more qubits, where one or more qubits act as control for some operation. Generally, if U is a gate that operates on single qubits with the following matrix representation:

$$U = \begin{pmatrix} u_{00} & u_{01} \\ u_{10} & u_{11} \end{pmatrix}, \tag{2.13}$$

then the controlled-U gate is a gate that operates on two qubits where the first qubit serves as control. The general matrix representation of the controlled-U then, if qubit 0 is the control and qubit 1 is the target, looks as following:

$$C_U = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & u_{00} & u_{01} \\ 0 & 0 & u_{10} & u_{11} \end{pmatrix}. \tag{2.14}$$

2.4 Toffoli Gate

The Toffoli gate has three inputs and outputs (Figure 2.4), where two of the input qubits act as control bits. The third qubit is the target bit which is flipped if both control qubits are set to $|1\rangle$, otherwise it's left alone. For example, applying the Toffoli gate to the state $|110\rangle$ flips the third qubit, resulting in the state $|111\rangle$.

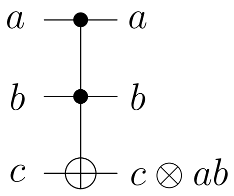


Figure 2.4: *Circuit representation of the Toffoli gate, where $\otimes$ is addition modulo two.*

The Toffoli, or controlled-controlled-X (CCX) gate can be represented by a 8×8 matrix:

$$U_{CCX} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}. \tag{2.15}$$

2.5 Universal Gate Sets

In classical systems the NAND gate is a universal gate, meaning that any other gate can be represented as a combination of NAND gates. In quantum computing there exist universal gate sets. A universal gate set requires the full Clifford and Pauli groups, and one or more non-Clifford gates.

We've seen the Pauli gates in Section 1.3.1. On their own, Pauli gates have no interesting computational capabilities. The Clifford gates we've seen are H, S, CNOT and CZ. Clifford gates introduce the quantum phenomena superposition and entanglement. The Pauli and Clifford gates can be simulated efficiently by classical computers (*Gottesman-Knill theorem*) - showing no increase in efficiency over classical computers.

The non-Clifford gates, which are required for universal quantum computing, cannot be simulated efficiently and are exponentially hard to simulate. Some non-Clifford gates are Toffoli, T and the rotation gates R_x, R_y and R_z (which do arbitrary rotations around the axes). One set of universal gates is $\{H, T, \textsc{cnot}\}$.

2.6 Entanglement

Two qubits are *entangled* if and only if the state of those two qubits can *not be expressed as two individual states* (non-separable). Let's first take a look at a separable, or non-entangled state

$$|\psi\rangle = \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle). \tag{2.16}$$

This state can be separated and expressed as the following two individual states:

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle). \tag{2.17}$$

However, consider the state

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle). \tag{2.18}$$

This is an entangled state. It cannot be expressed as two individual states. We say two qubits are entangled if they have *nonzero concurrence*. The concurrence of a state can be calculated using the following formula:

$$C(|\psi\rangle) = 2|\alpha_{00}\alpha_{11} - \alpha_{01}\alpha_{10}|. \tag{2.19}$$

We can check if our entangled state (2.18) is indeed entangled:

$$C\left(\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)\right) = 2\left|\frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{2}}\right)\right| = 1. \tag{2.20}$$

It has a non-zero concurrence, so we can say it's entangled. How about the non-entangled state in 2.16?

$$C\left(\frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)\right) = 2\left|\frac{1}{2}\left(\frac{1}{2}\right) - \frac{1}{2}\left(\frac{1}{2}\right)\right| = 0. \tag{2.21}$$

A concurrence of 0, so it is not entangled.

2.7 The Bell States

The *Bell states* are four maximally entangled quantum states of two qubits:

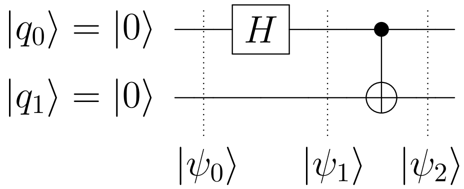
$$\left|\Phi^+\right\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \tag{2.22}$$

$$\left|\Phi^-\right\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) \tag{2.23}$$

$$\left|\Psi^+\right\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \tag{2.24}$$

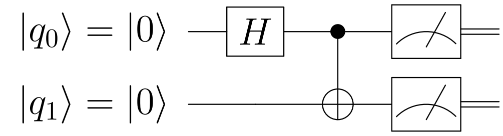
$$\left|\Psi^-\right\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle) \tag{2.25}$$

We can create a Bell state with the following circuit:



We start with our state $|00\rangle$ at $|\psi_0\rangle$. We put $|q_0\rangle$ through a Hadamard gate, giving us the state $(|00\rangle + |10\rangle)/\sqrt{2}$ at $|\psi_1\rangle$. Finally we CNOT that state giving us the final Bell state $|\Phi^+\rangle = (|00\rangle + |11\rangle)/\sqrt{2}$ at $|\psi_2\rangle$.

The significance of Bell states becomes apparent when we start measuring qubits of a Bell state. Take the circuit we used before to create the Bell state $(|00\rangle + |11\rangle)/\sqrt{2}$ and measure both qubits.



You will find that the measurement results are correlated:

$$\begin{aligned} M(q_1) = 0 \text{ if } M(q_0) = 0 \\ M(q_1) = 1 \text{ if } M(q_0) = 1 \end{aligned} \tag{2.26}$$

If you measure q_0 to be 0, q_1 will also be 0 and vice versa. Note that in our example entangled state they correlate as being equal, but for the entangled state $(|01\rangle + |10\rangle)/\sqrt{2}$ they correlate as being opposites (measuring q_0 as 0 means q_1 will be 1).

2.8 Greenberger-Horne-Zeilinger State

A Greenberger-Horne-Zeilinger (GHZ) state is a certain type of entangled state. It is a $M > 2$ system state:

$$|\mathrm{GHZ}\rangle = \frac{|0\rangle^{\otimes M} + |1\rangle^{\otimes M}}{\sqrt{2}}, \tag{2.27}$$

where $|j\rangle^{\otimes M}$ means the Kronecker product with itself M times. For example $M = 3$:

$$|\mathrm{GHZ}\rangle = \frac{|000\rangle + |111\rangle}{\sqrt{2}}. \tag{2.28}$$

GHZ states are used for example in cryptography for secret sharing.

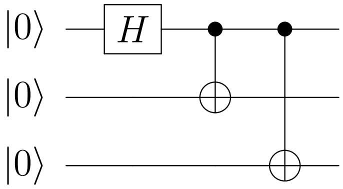


Figure 2.5: *Circuit creating a three-qubit GHZ state.*

2.9 Calculating Parity

Parity checking is one of the simplest forms of error detecting code. It tells us if the number of ones in a set of bits is even or odd. For example, 001 has a parity of 1 (odd) and 110 has a parity of 0 (even). We introduce a quantum algorithm for calculating the parity of an n-qubit state. For this example we'll determine the parity of a two-qubit state. Consider the circuit in Figure 2.6. We will calculate the parity of $|q_0\rangle|q_1\rangle$ with $|q_2\rangle$.

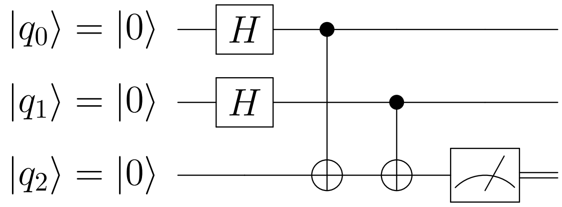


Figure 2.6: *Circuit for calculating parity:* $\mathrm{CNOT}_{1,2}\mathrm{CNOT}_{0,2}H_1H_0\,|000\rangle$.

We start with putting $|q_0\rangle$ and $|q_1\rangle$ in an arbitrary superposition. In this case we apply a Hadamard gate to both qubits, giving us the state

$$|\psi\rangle = \frac{1}{2}(|000\rangle + |010\rangle + |100\rangle + |110\rangle). \tag{2.29}$$

Then we apply a $\textsc{cnot}_{0,2}$ operation, giving

$$|\psi\rangle = \frac{1}{2}(|000\rangle + |010\rangle + |101\rangle + |111\rangle), \tag{2.30}$$

and a $\textsc{cnot}_{1,2}$:

$$|\psi\rangle = \frac{1}{2}(|000\rangle + |011\rangle + |101\rangle + |110\rangle). \tag{2.31}$$

If you look closely at this state, you can see that $|q_2\rangle$ corresponds to the parity of $|q_0\rangle|q_1\rangle$. We have essentially calculated the parities of all possible states of $|q_0\rangle|q_1\rangle$ in parallel. However, we cannot observe a quantum state to extract all the possible states' information. We are limited to measuring one outcome. When we measure $|q_2\rangle$ the state partially collapses leaving only the states with parity q_2. Let's take a look at the possible states after measuring $|q_2\rangle$:

$$|\psi\rangle = \begin{cases} \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \otimes |0\rangle \text{ if } M(q_2) = 0 \\ \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \otimes |1\rangle \text{ if } M(q_2) = 1 \end{cases} \tag{2.32}$$

Measuring a 0 leaves us with even parity states and measuring a 1 leaves us with odd parity states. $|00\rangle$ and $|11\rangle$ have even parity ($q_2 = 0$), $|01\rangle$ and $|10\rangle$ have odd parity ($q_2 = 1$).

2.10 Quantum Teleportation

Quantum teleportation it a technique for moving arbitrary quantum states around. It uses an *EPR pair* (a pair of qubits that is in a Bell state) that is shared between the sender and receiver. Note that it is impossible to clone a qubit state. This is referred to as the *no-cloning theorem*. Quantum teleportation works as following: Alice and Bob generate an EPR pair and both take one qubit before they get separated. Alice wants to deliver a qubit $|\phi\rangle$, whose state is unknown, to Bob. Alice interacts the qubit $|\phi\rangle$ with her half of the EPR pair, and then measures the two qubits in her possession. At this point, Alice's qubits are in one of the four classical states 00, 01, 10 or 11. She sends this information to Bob. Bob then performs one of four operations on his half of the EPR pair, recovering Alice's quantum state $|\phi\rangle$.

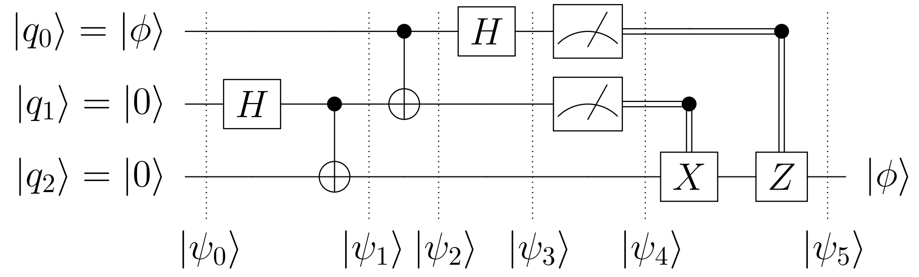


Figure 2.7: *Quantum circuit teleporting a quantum state $|\phi\rangle$.*

The circuit in Figure 2.7 teleports the unknown state $|\phi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$. We start at $|\psi_0\rangle$ with the state $|\phi\rangle|00\rangle$. Then we create an EPR pair with $|q_1\rangle|q_2\rangle$, where $|q_1\rangle$ belongs to Alice and $|q_2\rangle$ to Bob. This gives us the following state at $|\psi_1\rangle$:

$$|\psi_1\rangle = |\phi\rangle\,\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle), \tag{2.33}$$

which we can rewrite as following

$$\begin{aligned} |\psi_1\rangle &= \alpha\,|0\rangle + \beta\,|1\rangle \otimes \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) && (2.34)\\ &= \frac{1}{\sqrt{2}}\Big(\alpha\,|0\rangle\,(|00\rangle + |11\rangle) + \beta\,|1\rangle\,(|00\rangle + |11\rangle)\Big). && (2.35) \end{aligned}$$

Alice then sends her qubits through a $\text{CNOT}_{0,1}$ gate, obtaining

$$|\psi_2\rangle = \frac{1}{\sqrt{2}}\Big(\alpha\,|0\rangle\,(|00\rangle + |11\rangle) + \beta\,|1\rangle\,(|10\rangle + |01\rangle)\Big). \tag{2.36}$$

She then sends $|q_0\rangle$ through a Hadamard gate, giving

$$|\psi_3\rangle = \frac{1}{2}\Big(\alpha(|0\rangle + |1\rangle)(|00\rangle + |11\rangle) + \beta(|0\rangle - |1\rangle)(|10\rangle + |01\rangle)\Big). \tag{2.37}$$

This state can be rewritten in the following way:

$$\begin{aligned} |\psi_3\rangle = \frac{1}{2}\Big(&\;|00\rangle\,(\alpha\,|0\rangle + \beta|1\rangle)\; + |01\rangle\,(\alpha\,|1\rangle + \beta\,|0\rangle)\\ &+ |10\rangle\,(\alpha\,|0\rangle - \beta\,|1\rangle) + |11\rangle\,(\alpha\,|1\rangle - \beta\,|0\rangle)\Big). \end{aligned} \tag{2.38}$$

This expression breaks down in four terms. The first term has Alice's qubits in state $|00\rangle$ and Bob's qubit in state $\alpha\,|0\rangle + \beta\,|1\rangle$, which is our original state $|\phi\rangle$. So in the case that Alice measures $M(q_0q_1) = 00$ at $|\psi_4\rangle$, Bob's qubit will be in state $|\phi\rangle$. Depending on Alice's measurement at $|\psi_4\rangle$, Bob may have to "fix" his state to recover $|\phi\rangle$ by applying the appropriate gate(s). For example, if Alice measures 00, Bob doesn't have to do anything. If Alice measures 01, Bob can fix his state by applying an X gate. If Alice measures 10, Bob can fix his state by applying a Z gate. And if Alice measures 11, Bob can fix his state by first applying an X gate and then a Z gate. After fixing his state, Bob ends up with Alice's state $|\phi\rangle$ at $|\psi_5\rangle$.

Note that quantum teleportation does *not* allow for faster than light communication. This is because Bob needs to be told of the result of Alice's measurements through a classical channel in order to complete the teleportation. Also note that we did not clone the quantum state $|\phi\rangle$, we merely moved it. Teleporting a state depends on the measurement and thus collapsing of the original state $|\phi\rangle$, so it never allows for cloning.

Chapter 3

Quantum Algorithms

Quantum computers promise to solve problems intractable by classical computers. One application is quantum simulation. For example: simulating chemical processes, which are actually quantum processes. As Richard Feynman said: "*Nature isn't classical, dammit, and if you want to make a simulation of nature, you'd better make it quantum mechanical.*" Other possible applications include optimization, cryptography and machine learning. There are a lot of efforts in trying to apply quantum computing, but many remain skeptic. We should assume no guarantees in the applications of quantum computers, but continue exploring the possibilities.

When a quantum computer can calculate something efficiently that a classical computer cannot calculate efficiently, we will have reached *quantum supremacy*. The boundary of reaching this is around 50 qubits, when it becomes impossible to simulate a quantum computer. We have not yet reached quantum supremacy, but we are getting close to it. There have been found quantum algorithms which provide a speedup over classical algorithms, like Shor's algorithm for factoring integers and the Deutch-Jozsa algorithm for determining if a boolean function is balanced or unbalanced.

3.1 Quantum Arithmetic

Any classical circuit can be simulated by a quantum circuit. We can show this by creating quantum versions of classical arithmetic functions. One problem with simulating classical gates as quantum gates is that most of them are non-reversible. In quantum computing, it is required for gates to be reversible. This is often solved by adding an output qubit for every input qubit and an extra result qubit (Figure 3.1). We can create a quantum half adder by creating sum and carry circuits, just like in classical computers. The truth table of the sum function $f_s(x)$ can be found in Figure 3.2. We can implement this as a quantum circuit using CNOT gates as seen in Figure 3.3.

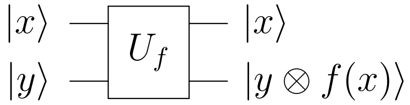


Figure 3.1: *Create output qubits for every input qubit in $|x\rangle$ and store the result of $f(x)$ in $|y\rangle$. This transformation can be described as $U_f\,|x\rangle\,|y\rangle = |x\rangle\,|y \otimes f(x)\rangle$. The $\otimes$ symbol in this context means the binary sum (*XOR*).*

x_0x_1	$f_s(x)$
00	0
01	1
10	1
11	0

Figure 3.2: *Bit sum truth table.*

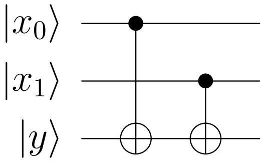


Figure 3.3: *Quantum circuit for summing two bits using two* CNOT *gates.*

To fully add two bits we need to account for the carry. The carry function $f_c(x)$ is 1 if and only if both inputs are 1 (Figure 3.4). This truth table corresponds directly to the Toffoli gate (Figure 3.5).

x_0x_1	$f_c(x)$
00	0
01	0
10	0
11	1

Figure 3.4: *Bit carry truth table.*

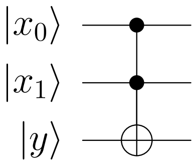


Figure 3.5: *Quantum circuit for taking the carry of two bits implemented using a Toffoli gate.*

We can combine our sum and carry circuit into one, creating a quantum half adder.

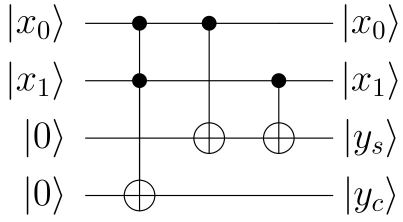


Figure 3.6: *Quantum half adder. The sum ends up in $|y_s\rangle$ and the carry in $|y_c\rangle$.*

Let's check if this circuit works. Say we want to add $|x_0\rangle = |1\rangle$ and $|x_1\rangle = |1\rangle$. We would expect to get 0 as sum and 1 as carry. Our initial state is $|1100\rangle$. We apply a Toffoli gate, setting the carry qubit to 1: $|1101\rangle$. Then we set the sum qubit by doing a CNOT$_{0,2}$ and a CNOT$_{1,2}$ ending up with the state $|1101\rangle$. Measuring the sum and carry qubits gives $M(y_s) = 0$ and $M(y_c) = 1$, giving us what we expected and thus having done a quantum half addition of two qubits.

3.2 Deutch-Jozsa Algorithm

We have shown that we can simulate classical circuits on a quantum computer. This in itself isn't very impressive however, we could already do that on a classical computer. In this section we will describe an algorithm with quantum speedup. Consider the four single-bit operations identity, NOT, reset and set.

Identity
$f_i(x) = x$

x	$f_i(x)$
0	0
1	1

NOT
$f_n(x) = \neg x$

x	$f_n(x)$
0	1
1	0

Reset
$f_r(x) = 0$

x	$f_r(x)$
0	0
1	0

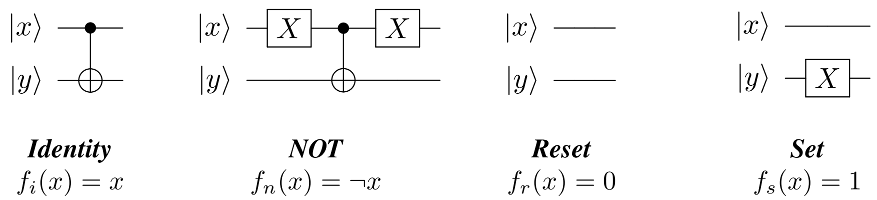
Set
$f_s(x) = 1$

x	$f_s(x)$
0	1
1	1

Figure 3.7: *Truth tables for the four single-bit operations.*

Say we are faced with the following problem: we have a "black box" with one of the four one-bit functions, but we're not told which one. Determine if the function is *balanced* or *unbalanced*. A balanced function is a function which returns the same amount of 0 and 1s it received as input, like identity and NOT. Unbalanced functions are functions which return a constant value, like reset and set. This is known as *Deutch's problem* and is one of the first examples of a problem where there exists a quantum algorithm that is exponentially faster than any deterministic classical algorithm.

On a classical computer, to determine if the black box is balanced or unbalanced, we need to do two function calls. On a quantum computer, using the *Deutch-Jozsa* algorithm, we can figure it out using one call to the black box. To use this algorithm for Deutch's problem we need to translate the four single-bit operations to the quantum world. The quantum circuits corresponding to each single-bit operation can be found in Figure 3.8.



Identity $f_i(x) = x$

NOT $f_n(x) = \neg x$

Reset $f_r(x) = 0$

Set $f_s(x) = 1$

Figure 3.8: *Quantum circuit representation of the four single-bit operations.*

With these definitions we can start using the Deutch-Jozsa algorithm to efficiently determine what kind of function our black box is. The circuit to do so is described in Figure 3.9. In this circuit, U_f refers to our black box function, or as it is sometimes also called, *oracle*.

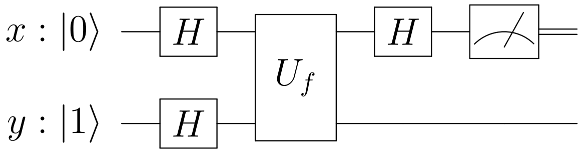


Figure 3.9: *Quantum circuit to determine if f is a balanced or unbalanced function using the Deutch-Jozsa algorithm. U_f is the quantum circuit which transforms $|x\rangle|y\rangle$ to $|x\rangle|y \otimes f(x)\rangle$.*

We can try it out by trying to determine if the identity function $f_i(x) = x$ is balanced or unbalanced. We start with putting $|x\rangle$ and $|y\rangle$ through a Hadamard gate:

$$|\psi\rangle = \frac{1}{2}(|00\rangle - |01\rangle + |10\rangle - |11\rangle). \tag{3.1}$$

We evaluate our identity function using by applying U_f:

$$\begin{aligned} |\psi\rangle = \frac{1}{2}\big(&|0\rangle\,|0 \otimes f_i(0)\rangle - |0\rangle\,|1 \otimes f_i(0)\rangle \\ &+ |1\rangle\,|0 \otimes f_i(1)\rangle - |1\rangle\,|1 \otimes f_i(1)\rangle\big). \end{aligned} \tag{3.2}$$

This state contains information about both $f_i(0)$ and $f_i(1)$! We have essentially evaluated two values of $f_i(x)$ simultaneously, a feature known as *quantum parallelism*. Actually applying the function $f_i(x)$, which as we can see in Figure 3.8 is just a $\textsc{cnot}_{0,1}$, gives us the state

$$|\psi\rangle = \frac{1}{2}(|00\rangle - |01\rangle + |11\rangle - |10\rangle). \tag{3.3}$$

Finally we apply another Hadamard gate on $|x\rangle$:

$$\begin{aligned} |\psi\rangle &= \frac{1}{2\sqrt{2}}\big(|00\rangle + |10\rangle - |01\rangle - |11\rangle \\ &\qquad + |01\rangle - |11\rangle - |00\rangle + |10\rangle) \qquad (3.4) \\ &= \frac{1}{2\sqrt{2}}(|10\rangle + |10\rangle - |11\rangle - |11\rangle) \qquad (3.5) \\ &= \frac{1}{\sqrt{2}}(|10\rangle - |11\rangle). \qquad (3.6) \end{aligned}$$

We end up with the final state $\frac{1}{\sqrt{2}}(|10\rangle - |11\rangle)$ before measurement. We can see from this state that $M(x) = 1$. Curiously, the following measurement outcomes apply:

$$\begin{aligned} &|0_x\rangle \otimes \frac{1}{\sqrt{2}}(|0_y\rangle - |1_y\rangle) \text{ if } f(0) = f(1) \\ &|1_x\rangle \otimes \frac{1}{\sqrt{2}}(|0_y\rangle - |1_y\rangle) \text{ if } f(0) \neq f(1) \end{aligned} \tag{3.7}$$

That is, we get $M(x) = 1$ if $f(0) \neq f(1)$, which means f is a balanced function. Likewise, we get $M(x) = 0$ if $f(0) = f(1)$, which means f is an unbalanced function. Since we got a measurement of 1 for f_i, and given the fact that the identity operation is balanced, we can say that we effectively determined that f_i is a balanced function with one call to the black box.

3.3 Quantum Fourier Transform

An important operation in classical computing and quantum computing as we will see is the *Fourier transform*. The Fourier transform transforms a function $f(t)$ in the time domain to another function $F(x)$ in the frequency domain. The opposite (transforming a function in the frequency domain to a function in the time domain) can be achieved by applying the *inverse* Fourier transform. Consider the sinusoidal function $f(t) = \cos(t)$. Figure 3.10 graphs this function in the time domain, with time on the x axis and amplitude on the y axis. Applying the Fourier transform on this function gives us a function of the frequency domain, as seen in Figure 3.11.

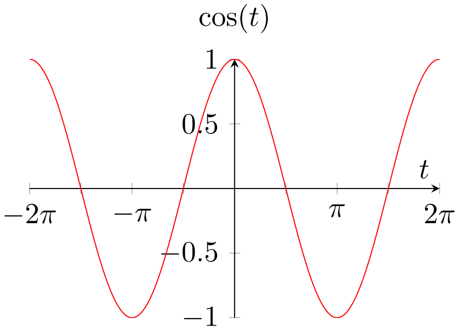


Figure 3.10: *Time domain plot of* $\cos(t)$.

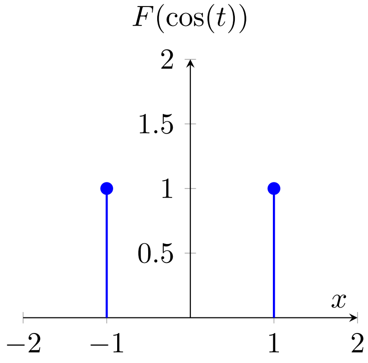


Figure 3.11: *Frequency domain plot of* $F(\cos(t))$.

The Fourier transform is a very useful tool in computer science, used for example in signal processing. The signal shown in Figure 3.10 is a continuous signal, meaning it is assumed to extend to infinity. Computers however can only deal with finite, non-continuous signals. This means we have to discretize the continuous function into discrete counterparts as seen in Figure 3.12. We can then transform the discrete signal using the *discrete Fourier transform*.

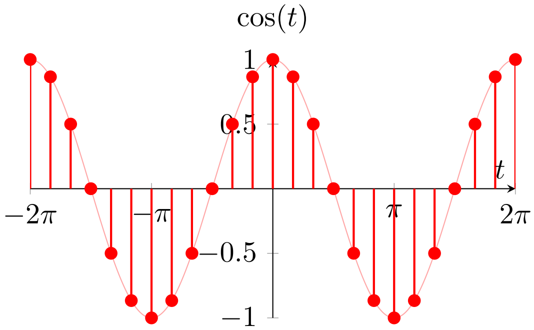


Figure 3.12: *Discretized counterpart of the continuous function cos(t). Each dot represents a sample of the signal. The time interval between the samples is called the sampling interval.*

The discrete Fourier transform can be thought of as a linear operation. This means we can map a N-dimensional state vector to another N-dimensional state vector by a $N \times N$ matrix F_N. For a n qubit state N is equal to 2^n. Such transformation may be written as

$$|x\rangle = \sum_{j=0}^{N-1} x_j \, |j\rangle \longrightarrow \sum_{k=0}^{N-1} y_k \, |k\rangle \,, \tag{3.8}$$

where the amplitudes y_k are the discrete Fourier transform of the amplitudes x_j. In other words, the quantum Fourier transform (QFT) transforms the computational basis states as following

$$F_N \, |j\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i jk/N} \, |k\rangle \,. \tag{3.9}$$

Note that this is the equivalent of a classical *inverse* discrete Fourier transform. The quantum Fourier transform (QFT) has the same effect as the classical inverse Fourier transform and vice versa. The quantum Fourier transform provides an exponential speedup over the classical fast Fourier transform algorithm.

We have already seen and used the QFT on a single qubit. The unitary matrix of the QFT on one qubit is the Hadamard gate:

$$F_2 = H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} . \tag{3.10}$$

For a multi-qubit system it becomes a bit more involved. To describe the circuit for the n-qubit QFT we need the Hadamard and controlled phase gate, which we will write as R_m where

$$R_k = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^k} \end{pmatrix}. \tag{3.11}$$

The circuit for a QFT on 3 qubits can be seen in Figure 3.14. Note that $S = R_2$ and $T = R_3$. The crossed gate at the end is the swap gate, which swaps two qubits. The order of qubits has to be reversed at the end of a QFT to get the correct order as result. Note that inverting the circuit by reversing the order of the gates and taking the inverse $U^\dagger$ of each gate U gives an equally efficient circuit for the inverse quantum Fourier transform $F_N^\dagger$.

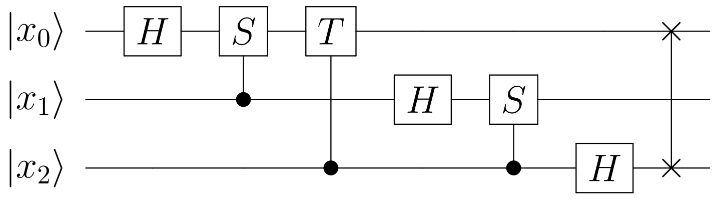


***Figure 3.13:** Quantum Fourier transform on 3 qubits.*

The matrix of a QFT can be written out using $\omega = e^{2\pi i/N}$. For 3 qubits, where $N = 8$:

$$F_N = \frac{1}{\sqrt{N}} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \omega^1 & \omega^2 & \omega^3 & \omega^4 & \omega^5 & \omega^6 & \omega^7 \\ 1 & \omega^2 & \omega^4 & \omega^6 & 1 & \omega^2 & \omega^4 & \omega^6 \\ 1 & \omega^3 & \omega^6 & \omega^1 & \omega^4 & \omega^7 & \omega^2 & \omega^5 \\ 1 & \omega^4 & 1 & \omega^4 & 1 & \omega^4 & 1 & \omega^4 \\ 1 & \omega^5 & \omega^2 & \omega^7 & \omega^4 & \omega^1 & \omega^6 & \omega^3 \\ 1 & \omega^6 & \omega^4 & \omega^2 & 1 & \omega^6 & \omega^4 & \omega^2 \\ 1 & \omega^7 & \omega^6 & \omega^5 & \omega^4 & \omega^3 & \omega^2 & \omega^1 \end{pmatrix}. \tag{3.12}$$

A more general circuit of the QFT for a n qubit state is described below in Figure 3.14.

The result of the Fourier transform of the original state is stored in the amplitudes. Remember however that these amplitudes cannot be extracted. Thus there is no way of determining the Fourier transform of the original state using the QFT. Even though we can't extract the transformed values after a QFT, the QFT has its use in quantum algorithms like Shor's algorithm for factoring integers and the quantum phase estimation algorithm.

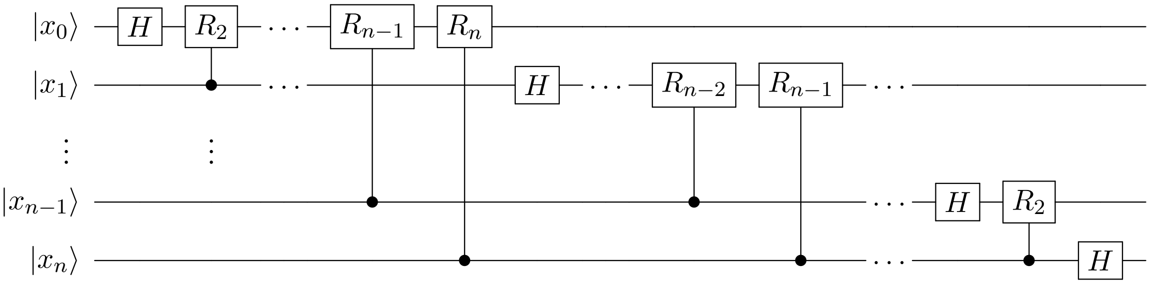


Figure 3.14: *Quantum Fourier transform on n qubits. Swap gates required to get the correct order at the end are omitted.*

3.4 Quantum Phase Estimation Algorithm

The quantum phase estimation algorithm (QPE) is a quantum algorithm to estimate the phase of an eigenvector of a unitary operation. Say we prepare an eigenstate $|u\rangle$ of a unitary operator U, where $U|u\rangle = e^{2\pi i\varphi}|u\rangle$ and the value of φ is unknown $(0 \leq \varphi \leq 1)$. The goal is to estimate the value of φ when we don't necessarily know U or $|u\rangle$, but have available black boxes capable of preparing $|u\rangle$ and applying controlled-U operations.

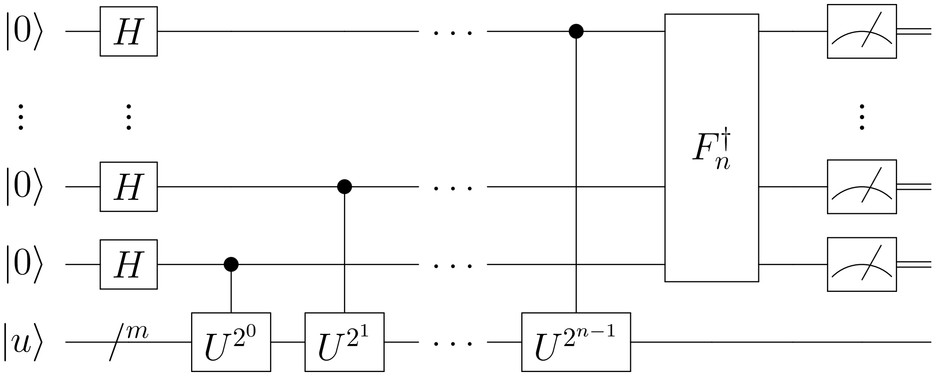


Figure 3.15: *Quantum phase estimation circuit, where $F_n^\dagger$ represent the inverse quantum Fourier transform on n qubits. The '/' denotes a register of m qubits.*

The circuit for the QPE algorithm found in Figure 3.15 consists of two registers. The upper n qubits comprise the first register which will be used to calculate the estimate of φ. The size of the first register can be chosen based on two factors: accuracy and success probability. More qubits will give you better accuracy and a higher success probability. The lower m qubits are the second register which will be prepared with the eigenstate $|u\rangle$ whose phase we want to estimate.

We start with the system in the state $|0\rangle^{\otimes n}|u\rangle$. After applying n Hadamard operations $H^{\otimes n}$ on the first register we end up with the state

$$\frac{1}{\sqrt{2^n}}(|0\rangle + |1\rangle)^{\otimes n}|u\rangle . \tag{3.13}$$

We continue by applying the controlled unitary operators U. Remembering that $U|u\rangle = e^{2\pi i\varphi}|u\rangle$, then $U^{2^j}|u\rangle = e^{2\pi i 2^j\varphi}|u\rangle$. A controlled-$U^{2^j}$ operator then transforms a state as following:

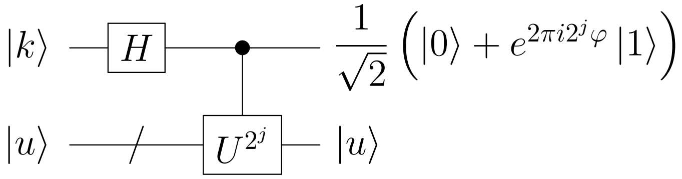

$|k\rangle$
H
$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 2^j\varphi}|1\rangle\right)$
$|u\rangle$
U^{2^j}
$|u\rangle$

Figure 3.16: *Controlled-U^{2^j} operation on a qubit $|k\rangle$ and eigenstate $|u\rangle$.*

Note that the phase ends up in $|k\rangle$ while $|u\rangle$ stays the same. This is a phenomenon called *quantum phase kickback* and can be explained by examining the state throughout the transformation. The system in Figure 3.16 after the Hadamard gate is in the state

$$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|u\rangle = \frac{1}{\sqrt{2}}(|0\rangle|u\rangle + |1\rangle|u\rangle). \tag{3.14}$$

Then a controlled-U^{2^j} is applied, meaning that U^{2^j} is only applied when the control qubit ($|k\rangle$) is $|1\rangle$. We then end up with the state

$$\begin{aligned} &\frac{1}{\sqrt{2}}\left(|0\rangle|u\rangle + |1\rangle U^{2^j}|u\rangle\right) && (3.15)\\ =&\frac{1}{\sqrt{2}}\left(|0\rangle|u\rangle + |1\rangle e^{2\pi i 2^j\varphi}|u\rangle\right) && (3.16)\\ =&\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 2^j\varphi}|1\rangle\right)|u\rangle . && (3.17)\end{aligned}$$

So, after applying the controlled-U operators in the QPE circuit as seen in Figure 3.15, the state of the first register becomes

$$\begin{aligned} &\frac{1}{\sqrt{2^n}}\left(|0\rangle + e^{2\pi i 2^{n-1}\varphi}|1\rangle\right)\cdots\left(|0\rangle + e^{2\pi i 2^1\varphi}|1\rangle\right)\left(|0\rangle + e^{2\pi i 2^0\varphi}|1\rangle\right) && (3.18)\\ &= \frac{1}{\sqrt{2^n}}\sum_{k=0}^{2^n-1} e^{2\pi i\varphi k}|k\rangle . && (3.19)\end{aligned}$$

The second register stays in the state $|u\rangle$ throughout this computation. The total circuit is in the following state after the controlled-U operations:

$$\frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i \varphi k} |k\rangle |u\rangle . \tag{3.20}$$

Note that this state is similar to the result of a QFT (3.9). By performing the *inverse* QFT on the first register we get

$$\frac{1}{2^n} \sum_{j=0}^{2^n-1} \sum_{k=0}^{2^n-1} e^{-2\pi i kj/2^n} e^{2\pi i \varphi k} |k\rangle |u\rangle \tag{3.21}$$

$$= \frac{1}{2^n} \sum_{j=0}^{2^n-1} \sum_{k=0}^{2^n-1} e^{-2\pi i kj/2^n + 2\pi i \varphi k} |k\rangle |u\rangle . \tag{3.22}$$

If φ has a n bit binary fraction representation $\varphi = j/2^n$, measuring the first register in the computational basis will give us exactly φ. Note that φ doesn't always have a n bit binary fraction. In this case, we will get probabilistic measurement results. The probability $P(j)$ is large for values of j where $\varphi \approx j/2^n$, so in this case we can only get an approximation of φ. An important idea in this algorithm is the ability of the inverse quantum Fourier transform to perform the transformation

$$\frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i \varphi k} |k\rangle |u\rangle \longrightarrow |\tilde{\varphi}\rangle |u\rangle , \tag{3.23}$$

essentially "extracting" an estimation $|\tilde{\varphi}\rangle$ of φ.