1

Qubits, Gates, and Circuits

1.1 Bits and Qubits

Digital systems that are most familiar are based on binary digits, or "bits." Each bit can take on either the value "1" or "0," and any arbitrary data can be represented by such a binary representation. In addition, any arbitrary logical operation can be implemented using bits. We will refer to these familiar systems as "classical" systems, since they are governed by the everyday laws of classical physics.

Quantum computing is different from classical computing in a number of significant ways. The fundamental unit of information in quantum computing is the *qubit* (pronounced "KEW-bit"), short for quantum bit. The capabilities and behavior of qubits are quite different than bits, and we begin by pointing out and discussing the key differences as a launching point for our study of quantum computing.

1.1.1 Circuits in Space vs. Circuits in Time

A simple classical logic circuit is represented by the NOT gate shown in Figure 1.1(a). The NOT gate turns a "0" into "1" and vice versa. In this circuit diagram the horizontal direction represents *space*, i.e., the input and output of the circuit are physically accessible from different points in the circuit, and they can be measured simultaneously.

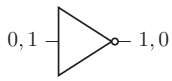
The quantum version of the NOT gate is the X gate shown in Figure 1.1(b). For qubits, the "0" and "1" states are normally written $|0\rangle$, and $|1\rangle$, respectively. We will discuss the meaning of this notation in more detail in a future section, but for now just consider them to be labels for the two states. In this case, the horizontal direction represents *time*, i.e., the input and output of the circuit represent the state of the *same* qubit after performing the X gate operation. In other words, unlike the usual structure of classical logic, a quantum gate represents an operation that you perform on a *single* qubit or set of qubits. The output effectively overwrites the input, and every time a gate is applied it changes the state of the qubit.

1.1.2 Superposition

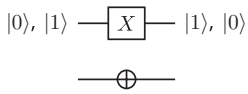
A classical bit must either be a "0" or a "1." In contrast, a qubit can also be in a *superposition state* that is part "0" and part "1." If the qubit is in a "1" or "0" state, we say

Principles of Superconducting Quantum Computers, First Edition. Daniel D. Stancil and Gregory T. Byrd.

Companion website: www.wiley.com/go/stancil/principlesofsuperconductingquantumcomputers



(a) Classical NOT Circuit diagram. The horizontal direction represents *space*, i.e., the input and output of the circuit are physically accessible from different points in the circuit, and they can be measured simultaneously.



(b) Quantum X gate circuit (quantum version of the NOT gate). The horizontal direction represents *time*, i.e., the input and output of the circuit represent the state of the same qubit after performing the X gate operation. The lower part of the Figure shows an alternate symbol for the quantum NOT gate.

Figure 1.1 Interpretation of classical versus quantum NOT gates.

the qubit is in a *basis* state.[1] For basis states, the state of the qubit can be measured any number of times without changing the state, much like measuring the state of a classical gate. A superposition state will also yield either a "0" or a "1" when measured, with a probability determined by the mixture. In this case the action of making the measurement will "collapse" the superposition state onto one of the constituent basis states, and the information stored in the superposition state will be lost. For example, if $|\psi\rangle$ happens to represent a superposition state, then measuring the qubit at any time will collapse the state to either $|0\rangle$ or $|1\rangle$, destroying the information in the superposition state. From this point on, repeated measurement of the qubit will always yield the same result as the first measurement, just like a classical bit.

Mathematically the superposition state can be written

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

where α and β are complex constants.

As mentioned, if such a superposition state is measured, it will always give either $|0\rangle$ or $|1\rangle$ but with probabilities of each determined by α and β. Specifically, the probabilities of measuring the two possible outcomes are given by

$$Pr[0] = |\alpha|^2, Pr[1] = |\beta|^2. \tag{1.2}$$

If these are the only two possible outcomes of the measurement, then the probabilities must sum to 1, or

$$|\alpha|^2 + |\beta|^2 = 1. \tag{1.3}$$

This ability to represent superposition states is one of the secrets to the power of quantum computing: there is a sense in which the qubits are able to explore multiple possibilities in parallel.

1 To be more precise, $|0\rangle$ and $|1\rangle$ are known as the "computational basis" or "standard basis" states. For this chapter, we will restrict our discussion to the standard basis and will simply refer to basis states.

1.1.3 No Cloning

In a classical logic circuit we can measure the state of a bit at any time, and make as many copies of the state as we want. We can also do this for a qubit if it is known to be in one of the basis states; as described above, we can measure the $|0\rangle$ or $|1\rangle$ state without disturbing it, and we can make as many copies of $|0\rangle$ or $|1\rangle$ as needed.

However, it turns out that it is not possible to create a precise, independent copy of an *arbitrary* quantum state. This is known as the *no-cloning theorem*. We'll show a proof in Section 1.7, but for now let us consider the challenges this poses to the quantum programmer.

For example, we can't get estimates of α or β from running a circuit once. Cloning would allow me to run the circuit, make lots of copies of the result, and then measure each copy to estimate $|\alpha|^2$ and $|\beta|^2$ by the probability of measuring $|0\rangle$ or $|1\rangle$. Without cloning, we can only measure the result once. To get many measurements, we must run the same computation many times to produce (hopefully!) the same result over and over.

We cannot make copies of states for breakpoints, or to help with debugging or error recovery. It is also challenging to apply different operators to a single state during the course of a computation. Classical programmers take the ability to copy a state for granted, and this limitation requires some adjustment.

It turns out that it is possible to copy a quantum state using entanglement (Section 1.1.5), but only by destroying the state being copied. This represents a *transfer* of state rather than a copy, and is referred to as *teleportation*.

1.1.4 Reversibility

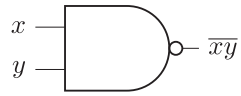
Classical NOT gates are reversible; i.e., two NOT gates in series returns the bit to its original state. However, the situation is different for classical logic gates with multiple inputs. As an example, consider the NAND Gate shown in Figure 1.2. It is not possible to uniquely determine the input bits from the output bit. Because of this, conventional multi-input logic is *irreversible*.[2]

In contrast, quantum gates are *reversible*. As a result, even though the output overwrites the input, the input is not lost since the effect of a series of gate operations can be reversed by applying the appropriate inverse operations.

1.1.5 Entanglement

The fact that we can only calculate the probability of measurement results—not the precise results—on qubits in superposition states allows for a phenomenon that has no classical analog: *entanglement*. If two qubits are entangled, then the states are correlated even though the outcome of a measurement on either of the qubits can only be predicted by its probability. For example, if two qubits are entangled, then a measurement on one of the qubits will give a result with probability determined by its superposition state, just as would happen with an isolated qubit. However, once the

2 Note: There are classical multi-input logic gates that *are* reversible, called "conservative logic" gates. To date these have not been widely used, but they have the advantage of making it possible in principle to compute without dissipating power [1].



(a) NAND circuit diagram

x	y	$\overline{xy}$
0	0	1
0	1	1
1	0	1
1	1	0

(b) NAND truth table

Figure 1.2 NAND circuit diagram.

state of one of the qubits has been collapsed by a measurement, the entanglement means that we know exactly what we will get if we measure the second qubit.

It is a bit like flipping two coins at the same time, and having them always come up the same, i.e., both heads or both tails. Or alternatively, having them always come up opposite—one heads and the other tails. Prior to making a measurement, the states of both of the qubits have not yet been determined. However, when the state of one is collapsed by a measurement, the second qubit somehow instantaneously "knows" the result. This is true regardless of how far apart the qubits are. For example, suppose we entangle two qubits, send one to New York and the other to Tokyo, then make prior arrangements to measure both at the same time. We will discover that the results of the measurements will be correlated even though there was not enough time for a signal traveling the speed of light to travel between the qubits. Einstein called this "spooky action at a distance."

If it were possible to control the result of the first measurement, then it would appear that we could communicate faster than the speed of light—violating the principle of special relativity. However, the fact that we cannot control the result of the first measurement means that we cannot actually send any information using this mechanism.[3] So we can rest comfortably knowing that relativity has not been violated.

We will see that the phenomena of *superposition* and *entanglement* give quantum computing its unusual and powerful capabilities.

1.2 Single-Qubit States

Since there are two components in a qubit's state, we can represent the state as a two-dimensional vector, referred to as a *state vector*. The basis states can be written

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{1.4}$$

It follows that we can express a general superposition state as a weighted sum of the basis states:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle = \alpha \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \beta \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}. \tag{1.5}$$

3 Entanglement can be used to increase the *capacity* of a communication channel, however, using a protocol known as *super-dense coding*.

The notation $|\psi\rangle$ is called a "ket." The transpose complex conjugate[4] of a ket is denoted $\langle\psi|$ and is called a "bra":

$$|\psi\rangle^{\dagger} = \langle\psi| = \begin{bmatrix}\alpha^* & \beta^*\end{bmatrix}, \tag{1.6}$$

where the † superscript indicates the transpose complex conjugate, also referred to as the Hermitian conjugate, or adjoint. The "bra" and "ket" terminology may seem strange until you multiply them together to form a "bra-ket." For example,

$$\langle\psi|\psi\rangle = \begin{bmatrix}\alpha^* & \beta^*\end{bmatrix}\begin{bmatrix}\alpha \\ \beta\end{bmatrix} = \alpha^*\alpha + \beta^*\beta \tag{1.7}$$

$$= |\alpha|^2 + |\beta|^2. \tag{1.8}$$

(Note that when written as a bra-ket, only a single vertical bar is used in the center.) The bra-ket operation (i.e., multiplying a bra and ket) is also referred to as an "inner product." Referring to (1.8), we see that the inner product of a state vector with itself gives the sum of the probabilities that each of the basis states will be obtained in a measurement. Since the sum of the probabilities of all possible outcomes must equal 1, we see that the state vectors must be properly normalized to a length of unity.

1.3 Measurement and the Born Rule

We have previously stated that the probability of measuring a given component of a superposition state is given by the magnitude squared of its coefficient. The act of measuring requires an apparatus that interacts with the qubit in order to extract information. The rules of quantum mechanics tell us that the apparatus can only give partial information, related to a set of basis states. For now, we will assume that measurement is always with respect to the standard basis states, $|0\rangle$ and $|1\rangle$, which is the case for most quantum computing systems. However, it is possible for a measurement apparatus to be associated with a different set of basis states. (We will discuss measurement in more detail in later chapters.)

When a qubit is measured: (a) the state is changed to one of the basis states associated with the measurement, and (b) the measurement apparatus tells us the resulting state. In general, the probability that a state $|\psi\rangle$ will be found in the basis state $|a\rangle$ when measured is given by

$$Pr[|a\rangle] = |\langle\psi|a\rangle|^2. \tag{1.9}$$

This is called the *Born Rule.* For example, the probability that the outcome of measuring the state $|\psi\rangle$ above is $|0\rangle, |1\rangle$ is given by

$$Pr[|0\rangle] = |\langle\psi|0\rangle|^2 = |\alpha|^2|\langle 0|0\rangle|^2 = |\alpha|^2, \tag{1.10}$$

$$Pr[|1\rangle] = |\langle\psi|1\rangle|^2 = |\beta|^2|\langle 1|1\rangle|^2 = |\beta|^2, \tag{1.11}$$

as we found before.

---

4 The transpose of a matrix or vector is formed by interchanging the rows and columns. The transpose of a column vector gives a row vector. The transpose complex conjugate is formed by taking the transpose of the vector or matrix and then taking the complex conjugate of each element.

1.4 Unitary Operations and Single-Qubit Gates

We refer to a transformation from one quantum state to another as a *gate*. The effect of a single qubit gate is to change α and β into a new mixture α' and β':

$$\begin{aligned} \alpha' &= U_{11}\alpha + U_{12}\beta \\ \beta' &= U_{21}\alpha + U_{22}\beta. \end{aligned} \tag{1.12}$$

This can be written as a matrix equation

$$\begin{bmatrix} \alpha' \\ \beta' \end{bmatrix} = \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \tag{1.13}$$

$$|\psi'\rangle \equiv U\,|\psi\rangle\,. \tag{1.14}$$

Since the length of the state vector must always be unity, we are only allowed to use matrices U that conserve the length of the vector. In other words, $\langle\psi'|\psi'\rangle = \langle\psi|\psi\rangle = 1$. This puts a very important constraint on the matrix U:

$$\langle\psi'|\psi'\rangle = \langle\psi|U^\dagger U|\psi\rangle = 1, \tag{1.15}$$

using the following observation:

$$\langle\psi'| = (U\,|\psi\rangle)^\dagger = \langle\psi|\,U^\dagger. \tag{1.16}$$

Since $\langle\psi|\psi\rangle = 1$, we conclude that

$$U^\dagger U = I, \tag{1.17}$$

where I is the identity matrix

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{1.18}$$

Matrices that satisfy this requirement are called *unitary* matrices. We can view these matrices as performing an operation on a qubit by changing the mixture of basis states. Consequently, the matrices U are also referred to as *unitary operators*.

The identity matrix I can be considered to be the simplest "gate" and leaves the state vector unchanged. Classically, the NOT gate is the only non-trivial single-bit gate. In contrast, there are many non-trivial single qubit quantum gates (technically, the number of 2×2 unitary matrices is unlimited). The most common non-trivial single qubit gates are the Pauli-X (X), Pauli-Y (Y), Pauli-Z (Z), and Hadamard (H) gates defined as follows:

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \tag{1.19}$$

$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \tag{1.20}$$

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \tag{1.21}$$

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{1.22}$$

To get an understanding of what these gates do, consider applying an X gate to the "ground" state $|0\rangle$:

$$X|0\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} = |1\rangle. \tag{1.23}$$

Similarly,

$$X|1\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} = |0\rangle. \tag{1.24}$$

We see then that the X gate is a "bit flip" gate, and transforms $|0\rangle$ into $|1\rangle$ and vice versa. This, then is the analog of the classical NOT gate. You should verify the following results from applying Y, Z, and H gates:

$$Y|0\rangle = i|1\rangle, \quad Y|1\rangle = -i|0\rangle, \tag{1.25}$$

$$Z|0\rangle = |0\rangle, \quad Z|1\rangle = -|1\rangle, \tag{1.26}$$

$$\begin{aligned} H|0\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \\ H|1\rangle &= \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle). \end{aligned} \tag{1.27}$$

In addition, it is interesting to note that each one of these matrices is its own Hermitian conjugate. Consequently, these four gates have the property that applying them twice gives the identity matrix:

$$X^2 = Y^2 = Z^2 = H^2 = I. \tag{1.28}$$

Note that while X, Y, and Z gates transform between the $|0\rangle$ and $|1\rangle$ states, the Hadamard gate actually creates a *superposition* state, and therefore will prove to be particularly useful. The states resulting from applying H to the basis states are given their own names:

$$|+\rangle \equiv H|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \tag{1.29}$$

$$|-\rangle \equiv H|1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}. \tag{1.30}$$

Multiple gates can be sequentially applied to a qubit by matrix multiplication:

$$XYH|0\rangle \tag{1.31}$$

This expression says that we start with the ground state, then apply a Hadamard gate, followed by a Pauli-Y and then a Pauli-Z. Note that this process conceptually moves from right to left. This computation can be represented by a quantum circuit diagram as shown in Figure 1.3. Note that the process moves from left to right on the circuit

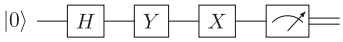


Figure 1.3 Circuit representation of Eq. (1.31). In a quantum circuit diagram, the operation goes from left to right, while the matrix expression is shown going from right to left. The final box is a measurement in the standard basis, resulting in a classical bit.

diagram: we begin with the ground state on the left, then apply a Hadamard gate, a Pauli-Y gate, and a Pauli-X.

Finally, after performing a quantum calculation, we generally measure each qubit. The symbol for a measurement is shown as the last element in Figure 1.3. This action collapses the final state onto one of the basis states. The outcome of a measurement is a classical bit that is stored in a classical register. "Quantum wires" are denoted by solid lines, and "classical wires" are denoted with double lines. A quantum wire simply represents a time-interval over which the state is kept unchanged.

1.5 Two-Qubit Gates

It is also possible to have gates with multiple qubits as inputs. However, unlike classical multi-bit gates, the fact that quantum gates are reversible requires that there must be the same number of output qubits as input qubits. In this section we will explore some common two-qubit gates.

1.5.1 Two-Qubit States

When we consider a system with two qubits, we don't just consider each qubit independently of the other. Instead, this forms a two-qubit system with its own set of basis states. If we know the state of each qubit, then the combined two-qubit state is described using the *tensor product* of the two state vectors, defined as follows:

$$\begin{bmatrix} a \\ b \end{bmatrix} \otimes \begin{bmatrix} c \\ d \end{bmatrix} = \begin{bmatrix} a \begin{bmatrix} c \\ d \end{bmatrix} \\ b \begin{bmatrix} c \\ d \end{bmatrix} \end{bmatrix} = \begin{bmatrix} ac \\ ad \\ bc \\ bd \end{bmatrix}. \tag{1.32}$$

Using the standard basis, the basis states for a two-qubit system are defined by combinations of the $|0\rangle$ and $|1\rangle$ states:

$$|00\rangle = |0\rangle \otimes |0\rangle = \begin{bmatrix} 1 \times \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ 0 \times \begin{bmatrix} 1 \\ 0 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tag{1.33}$$

$$|01\rangle = |0\rangle \otimes |1\rangle = \begin{bmatrix} 1 \times \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ 0 \times \begin{bmatrix} 0 \\ 1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \tag{1.34}$$

$$|10\rangle = |1\rangle \otimes |0\rangle = \begin{bmatrix} 0 \times \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ 1 \times \begin{bmatrix} 1 \\ 0 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \tag{1.35}$$

$$|11\rangle = |1\rangle \otimes |1\rangle = \begin{bmatrix} 0 \times \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ 1 \times \begin{bmatrix} 0 \\ 1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}. \tag{1.36}$$

Any two-qubit state can be written as a linear combination of the basis states:

$$|\psi\rangle = \alpha_{00}\,|00\rangle + \alpha_{01}\,|01\rangle + \alpha_{10}\,|10\rangle + \alpha_{11}\,|11\rangle\,. \tag{1.37}$$

Two-qubit state vectors are also normalized:

$$|\alpha_{00}|^2 + |\alpha_{01}|^2 + |\alpha_{10}|^2 + |\alpha_{11}|^2 = 1. \tag{1.38}$$

As we will see later, while every two-qubit state can be written in the form of Eq. (1.37), not every two-qubit state can be written as the tensor product of single-qubit states.

This can be generalized into a system with n qubits, requiring state vectors with 2^n components with 2^n complex coefficients.

1.5.2 Matrix Representation of Two-Qubit Gates

Just as single qubit gates can be represented by 2×2 matrices, an n-qubit gate can be represented by a $2^n \times 2^n$ matrix. Consequently two-qubit gates require the construction of 4×4 unitary matrices. Given two single-qubit operators A and B, the tensor product is defined as:

$$A \otimes B \equiv \begin{bmatrix} A_{11}B & A_{12}B \\ A_{21}B & A_{22}B \end{bmatrix} = \begin{bmatrix} A_{11}\begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} & A_{12}\begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} \\ A_{21}\begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} & A_{22}\begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} \end{bmatrix}, \tag{1.39}$$

which creates a 4×4 matrix.

Suppose we wanted to construct a two-qubit circuit starting in the state $|10\rangle$ with an X gate applied to the left qubit, and a Y gate applied to the other. Mathematically this

would be written

$$X \otimes Y \, |10\rangle \, . \tag{1.40}$$

Referring to (1.24) we see that the X gate will simply flip the left qubit, and referring to (1.25) we see that the Y gate will flip the right qubit and add the coefficient i. We conclude that

$$X \otimes Y \, |10\rangle = i \, |01\rangle \, . \tag{1.41}$$

To see how this would be implemented using the matrix representation, we first construct the $X \otimes Y$ matrix:

$$X \otimes Y = \begin{bmatrix} 0\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} & 1\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \\ 1\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} & 0\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & -i \\ 0 & 0 & i & 0 \\ 0 & -i & 0 & 0 \\ i & 0 & 0 & 0 \end{bmatrix} . \tag{1.42}$$

Completing the calculation gives the expected result:

$$X \otimes Y \, |10\rangle = \begin{bmatrix} 0 & 0 & 0 & -i \\ 0 & 0 & i & 0 \\ 0 & -i & 0 & 0 \\ i & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = i \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = i \, |01\rangle \, . \tag{1.43}$$

A particularly interesting two-qubit circuit is formed by applying a Hadamard gate to each qubit in the ground state: $H \otimes H \, |00\rangle$. Let us first compute $H \otimes H$:

$$H \otimes H = \frac{1}{2} \begin{bmatrix} 1\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} & 1\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \\ 1\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} & -1\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} . \tag{1.44}$$

Completing the calculation gives:

$$H \otimes H \, |00\rangle = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} . \tag{1.45}$$

Note that the resulting state vector can be decomposed into a sum of all of the two-qubit basis states:

$$\frac{1}{2} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \frac{1}{2} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} + \frac{1}{2} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} + \frac{1}{2} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} , \tag{1.46}$$

or alternatively

$$H \otimes H \, |00\rangle = \frac{1}{2} \left(|00\rangle + |01\rangle + |10\rangle + |11\rangle \right) . \tag{1.47}$$

We see that application of Hadamard gates to each qubit creates an equally weighted superposition of all possible basis states. This is often a very useful starting point for a quantum calculation.

Although the matrix representation can be helpful in understanding the operations, calculations can often be done more compactly once the effect of the gates are understood. For example, we could write $H \otimes H\,|00\rangle = H \otimes H\,|0\rangle\,|0\rangle$, apply the Hadamard gates to each qubit, and simplify:

$$\begin{aligned} H \otimes H\,|0\rangle\,|0\rangle &= H\,|0\rangle \otimes H\,|0\rangle \\ &= \frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + |1\rangle}{\sqrt{2}} \\ &= \frac{1}{2}\left(|00\rangle + |01\rangle + |10\rangle + |11\rangle\right). \end{aligned} \tag{1.48}$$

We conclude this section with a comment on notation. A more compact notation is often used for situations where the same operator is applied across multiple qubits; i.e., $H \otimes H$ is alternatively written $H^{\otimes 2}$, $H \otimes H \otimes H = H^{\otimes 3}$, etc.

1.5.3 Controlled-NOT

The gates that we have considered so far involve operations that are independently applied to separate qubits—there is no qubit–qubit interaction. If we are to entangle two qubits, then we need classes of gates where the operation on one qubit depends on the state of another. One of the most important such gates is the controlled-NOT, or CNOT gate. For this gate, one of the input qubits is the "control," and the other is the "target." If the control qubit is zero, then nothing is done to the target qubit, but if the control qubit is one, then the target qubit is flipped. For example, if the right qubit in our notation is the control and the left qubit is the target, then the CNOT gate transforms the basis states as follows:

$$U_{\mathrm{CN}}\,|00\rangle = |00\rangle\,; U_{\mathrm{CN}}\,|01\rangle = |11\rangle\,; U_{\mathrm{CN}}\,|10\rangle = |10\rangle\,; U_{\mathrm{CN}}\,|11\rangle = |01\rangle\,. \tag{1.49}$$

The effect of a CNOT can be compactly represented by $U_{\mathrm{CN}}\,|t\rangle\,|c\rangle = |c \oplus t\rangle\,|c\rangle$, where $\oplus$ represents exclusive-OR or modulo-2 addition (e.g., 0+1=1, but 1+1=0). The matrix representation of the CNOT gate is

$$U_{\mathrm{CN}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \tag{1.50}$$

and the circuit symbol is shown in Figure 1.4.

It is worth noting at this point that we are putting the least-significant qubit at the top of a circuit diagram, and on the right on the state labels used with kets. This labels states with the natural binary order. However, there are different conventions in use, and this

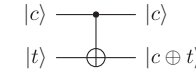


Figure 1.4 Symbol for a CNOT gate, and its effect on basis states.

can be a point of confusion. Some authors put the top-most qubit in a circuit diagram on the *left* when they label kets. In this alternate notation, $U'_{CN}\,|c\rangle\,|t\rangle = |c\rangle\,|c \oplus t\rangle$ so that

$$U'_{\text{CN}}\,|00\rangle = |00\rangle\,; U'_{\text{CN}}\,|01\rangle = |01\rangle\,; U'_{\text{CN}}\,|10\rangle = |11\rangle\,; U'_{\text{CN}}\,|11\rangle = |10\rangle\,. \tag{1.51}$$

The matrix representation of the CNOT gate in this alternate convention is

$$U'_{\text{CN}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{1.52}$$

We will consistently use the first convention, with the least-significant qubit (top-most on a circuit diagram) on the *right* when writing state labels.

1.6 Bell State

Consider the circuit shown in Figure 1.5. The circuit can be described mathematically as $U_{\text{CN}}\,(I \otimes H)\,|00\rangle$. Here the expression $I \otimes H$ simply means a Hadamard gate is applied to the right qubit, and the identity matrix applied to the left qubit (which leaves the left qubit unchanged). Completing the calculation gives

$$U_{\text{CN}}\,(I \otimes H)\,|00\rangle = U_{\text{CN}}\,|0\rangle \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) \tag{1.53}$$

$$= \frac{1}{\sqrt{2}} U_{\text{CN}}\,(|00\rangle + |01\rangle) \tag{1.54}$$

$$= \frac{1}{\sqrt{2}}\,(|00\rangle + |11\rangle)\,. \tag{1.55}$$

Note that there is no way to factor this state into (qubit 1 state) $\otimes$ (qubit 0 state). This is known as a *Bell State*, and it is an example of an *entangled* state, as described in Section 1.1.5.

If we measure the state of one of the qubits in (1.55), then we will obtain either a zero or a one with equal probability, as we would expect. The unusual behavior happens when we measure the second qubit: if we measure $|0\rangle$ on the first qubit, then we know the state has collapsed to $|00\rangle$, so a measurement of the second qubit will also yield $|0\rangle$. Similarly, if the first measurement yields $|1\rangle$, then the state must have collapsed to

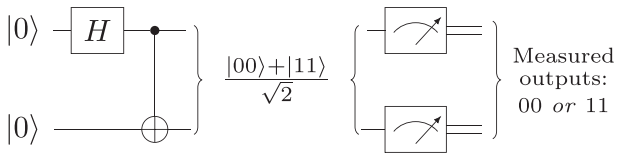


Figure 1.5 Circuit for creating an entangled state known as a Bell State. When the two qubits are measured, they will either both be 0, or they will both be 1.

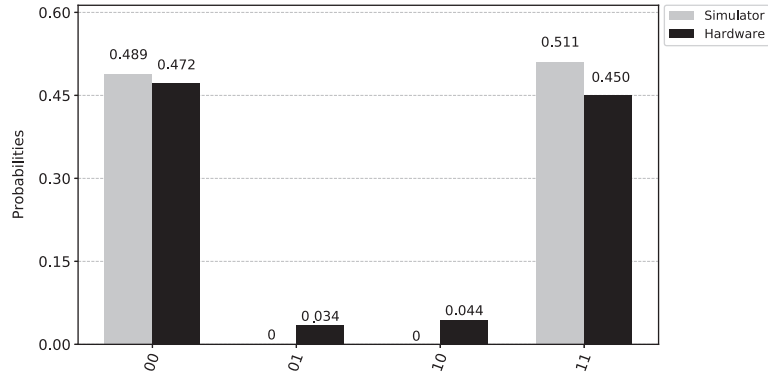


Figure 1.6 Result of executing the circuit 1024 times on a quantum simulator, compared with executing the circuit 1024 times on a real IBM quantum computer.

$|11\rangle$, so a measurement of the second qubit will also yield $|1\rangle$. The measurements are correlated, no matter how far apart the physical qubits might be.

We simulated the Bell circuit using IBM's Qiskit [2], and also executed the circuit on real IBM Quantum hardware. A histogram illustrating the results of simulating the circuit 1024 times (each execution is referred to as a "shot") is shown in Figure 1.6. From (1.55), we see that the probability amplitudes of the states $|00\rangle$ and $|11\rangle$ are both equal to $1/\sqrt{2}$. Since the probability of measuring a state is the absolute magnitude squared of the amplitude, the probability of measuring each state is $1/2$. Consistent with this, the measurements from the 1024 shots are roughly equally divided between the states $|00\rangle$ and $|11\rangle$, with no other states occurring.

The simulator gives the result expected from an error-free quantum processor. In contrast, the quantum processors available today are noisy and contain errors. As an illustration, Figure 1.6 also shows the result of executing 1024 shots on a real IBM Quantum processor. Although the states $|00\rangle$ and $|11\rangle$ do occur most frequently, the error states $|01\rangle$ and $|10\rangle$ occasionally occur as well. Fortunately there are techniques to reduce and partially mitigate such noise (Chapter 9), and these techniques represent an active area of research.

1.7 No Cloning, Revisited

With a better understanding of quantum states and operations, we are now ready to construct a proof of the no-cloning theorem. The proof relies on the fact that unitary operators are *linear*; when applied to a sum of states, the operator operates independently on each component:

$$U(|\psi\rangle + |\phi\rangle) = U|\psi\rangle + U|\phi\rangle\,. \tag{1.56}$$

Suppose we have an operator U_{clone} that takes two qubits, as shown in Figure 1.7. When the second qubit is $|0\rangle$, it is replaced with an exact copy of the first qubit. The

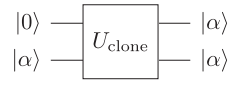


Figure 1.7 Hypothetical cloning operator, that creates an exact and independent copy of unknown quantum state $|\alpha\rangle$. The text will show that such an operator cannot be implemented.

proof will show that such an operator cannot exist, because it is not compatible with the principle of linearity.

Further suppose that we have two states:

$$|\alpha\rangle = a_0 |0\rangle + a_1 |1\rangle \tag{1.57}$$

$$|\beta\rangle = b_0 |0\rangle + b_1 |1\rangle . \tag{1.58}$$

By the definition of cloning:

$$\begin{aligned} U_{\text{clone}} |\alpha\rangle |0\rangle &= |\alpha\rangle |\alpha\rangle = |\alpha\alpha\rangle , \\ U_{\text{clone}} |\beta\rangle |0\rangle &= |\beta\rangle |\beta\rangle = |\beta\beta\rangle . \end{aligned} \tag{1.59}$$

Now consider a new state $|\delta\rangle = (|\alpha\rangle + |\beta\rangle)/\sqrt{2}$. By the definition of cloning:

$$\begin{aligned} U_{\text{clone}} |\delta\rangle |0\rangle &= |\delta\rangle |\delta\rangle \\ &= \frac{|\alpha\rangle + |\beta\rangle}{\sqrt{2}} \otimes \frac{|\alpha\rangle + |\beta\rangle}{\sqrt{2}} \\ &= \frac{1}{2} (|\alpha\alpha\rangle + |\alpha\beta\rangle + |\beta\alpha\rangle + |\beta\beta\rangle) . \end{aligned} \tag{1.60}$$

However, by the linearity of unitary operators:

$$\begin{aligned} U_{\text{clone}} |\delta\rangle |0\rangle &= \frac{1}{\sqrt{2}} U_{\text{clone}} (|\alpha\rangle |0\rangle + |\beta\rangle |0\rangle) \\ &= \frac{|\alpha\alpha\rangle + |\beta\beta\rangle}{\sqrt{2}} . \end{aligned} \tag{1.61}$$

Since Eqs. (1.60) and (1.61) cannot both be true, there is no unitary U_{clone} that can perform the cloning operation for all states.

We stated earlier that we *can* clone a (computational) basis state. This can be done with the CNOT gate, with the first qubit as the control. (With our bottom-to-top ordering, this corresponds to the U'_{CN} operator from (1.52).) Suppose state $|\psi\rangle$ is either $|0\rangle$ or $|1\rangle$, but we don't know which.

$$U'_{\text{CN}} |\psi\rangle |0\rangle = \left\{ \begin{matrix} U'_{\text{CN}} |00\rangle = |00\rangle , \text{ if } |\psi\rangle = |0\rangle \\ U'_{\text{CN}} |10\rangle = |11\rangle , \text{ if } |\psi\rangle = |1\rangle \end{matrix} \right\} = |\psi\psi\rangle . \tag{1.62}$$

If we apply the circuit from Figure 1.5 to an arbitrary state $|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$, we get a result that looks sort of like cloning, but not quite:

$$U_{\text{CN}}(I \otimes H) |0\rangle |\psi\rangle = \frac{\alpha |00\rangle + \beta |11\rangle}{\sqrt{2}} . \tag{1.63}$$

The result is not cloning because the two qubits are entangled. We did not succeed in creating two independent copies of $|\psi\rangle$. This is a useful construct, however, and can be

extended to create n-qubit states that look like this:

$$\frac{\alpha\,|0000\ldots\rangle + \beta\,|1111\ldots\rangle}{\sqrt{2}}.$$

These states will be useful for quantum error correction codes (Chapter 10).

1.8 Example: Deutsch's Problem

We now have enough quantum computing "machinery" to work out a simple example of a quantum algorithm. The example algorithm is admittedly not very useful, but it provides some important insights into how quantum computers can outperform classical computers.

Consider a function that takes a single binary digit as input, and provides a single binary digit as output. There are four possible functions: $f_1(x) = 0$, $f_2(x) = 1$, $f_3(x) = x$, and $f_4(x) = \bar{x}$. Suppose someone gave us an implementation of one of these functions in a black box, and asked us to determine whether $f(0) = f(1)$, or if $f(0) \neq f(1)$. Classically, we could do this in two function calls, one with $x = 0$, and one with $x = 1$. However, using a quantum algorithm we can answer this question with a single function call! Let's see how this could be done.

First, suppose that we have the gate U shown in Figure 1.8(a) for implementing the function $f(x)$. The function is implemented by changing the sign of the state: if $f(x) = 0$ then the sign is unchanged, and if $f(x) = 1$ then the sign of the state is flipped. We will return to the question of how to implement this gate shortly, but for now let us assume that the gate is given to us, and our job is to determine if $f(0) = f(1)$ or $f(0) \neq f(1)$. Now consider the circuit shown in Figure 1.8(b). Applying the Hadamard gate to the input $|0\rangle$ gives

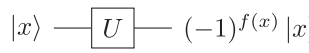
$$\psi_1 = \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right). \tag{1.64}$$

Note that the Hadamard gate enables us to apply U to *both* $|0\rangle$ and $|1\rangle$ at the *same time*. This is referred to as *quantum parallelism* and is one of the secrets behind the power of quantum computing (although there are some key qualifications that make this somewhat less exciting than it would seem at first!).

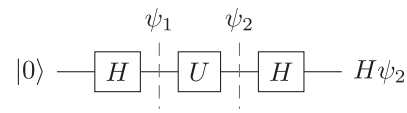
Continuing with the calculation, we next compute $\psi_2 = U\psi_1$ to obtain

$$\psi_2 = \frac{1}{\sqrt{2}}\left[(-1)^{f(0)}\,|0\rangle + (-1)^{f(1)}\,|1\rangle\right]. \tag{1.65}$$

Figure 1.8 Conceptual illustration of the Deutsch Problem.



(a) "Black Box" for evaluating the function $f(x)$.



(b) Circuit for illustrating Deutsch's Problem.

Applying the second Hadamard gate gives the final output:

$$H\psi_2 = \frac{1}{2}(-1)^{f(0)}\left[|0\rangle + |1\rangle\right] + \frac{1}{2}(-1)^{f(1)}\left[|0\rangle - |1\rangle\right] \tag{1.66}$$

$$= \frac{1}{2}\left[(-1)^{f(0)} + (-1)^{f(1)}\right]|0\rangle + \frac{1}{2}\left[(-1)^{f(0)} - (-1)^{f(1)}\right]|1\rangle\,. \tag{1.67}$$

Examining (1.67), we see that if $f(0) = f(1)$ the second term vanishes and the output is $|0\rangle$. In contrast, when $f(0) \neq f(1)$, then the first term vanishes and the output is $\pm|1\rangle$. Therefore, if we measure the output as $|0\rangle$, we know that $f(0) = f(1)$, while if we measure $|1\rangle$, we know that is not the case. (Our measuring device will ignore the minus sign and show us a result of "1" if the output state is $-|1\rangle$.)

Arranging for the amplitudes of the correct states to add and others to cancel is an example of *quantum interference*, another key principle underlying quantum computing. So we see that in a single function call, we are able to determine whether or not $f(x)$ is balanced or constant—a feat that would only be possible classically with two function calls!

Now let us turn our attention to how the gate U can be realized. First, we have to confess that the circuit in Figure 1.8 is hiding something: we need another qubit to implement the U gate.

Consider the function $f(x) = 0$. We want a unitary U_f that converts $|x\rangle$ to $|f(x)\rangle$, but this is clearly not reversible. Given an output of $|0\rangle$, there's no way to know whether the input is $|1\rangle$ or $|0\rangle$. To make the function reversible, we need to include both $f(x)$ and x in the output, so that we can reconstruct the input when reversing the operation.

For two outputs, we need two inputs. We typically use $|0\rangle$ as second input, but to be more general, let's assume that the second input can be any input $|y\rangle$. With $|x\rangle$ and $|y\rangle$ as the inputs, the outputs will be $|x\rangle$ and $|y \oplus f(x)\rangle$, where $\oplus$ is the standard Boolean exclusive-OR (XOR) operation. As shown in Figure 1.9, this function is its own inverse; if we use $|x\rangle$ and $|y \oplus f(x)\rangle$ as the input, we get $|x\rangle$ and $|y \oplus f(x) \oplus f(x)\rangle = |y\rangle$ as the output.

Figure 1.10 shows the four alternative implementations of U_f, depending on which version of $f(x)$ we want. From left to right:

- $f_1(x) = 0$, do nothing to y, equivalent to $y \oplus 0$.
- $f_2(x) = 1$, flip y using a NOT gate, equivalent to $y \oplus 1$.
- $f_3(x) = x$, use a CNOT gate to flip y if $x = 1$, equivalent to $y \oplus x$.
- $f_4(x) = \bar{x}$, flip x and then XOR with y using a CNOT gate, equivalent to $y \oplus \bar{x}$. We then use another NOT gate to restore x to its original value.

Now, using our U_f circuit, we will compute $(-1)^{f(x)}|x\rangle$ using a technique known as *phase kickback*. If we set $|y\rangle = |-\rangle = (|0\rangle - |1\rangle)/\sqrt{2})$, then the top output of U_f will

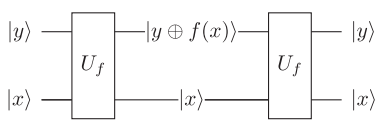


Figure 1.9 Reversible circuit for calculating *f(x)*.

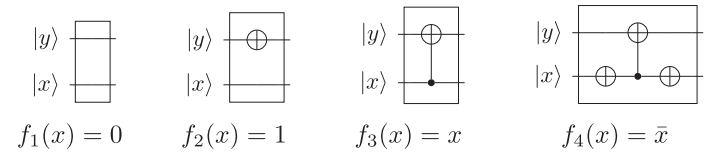


Figure 1.10 Implementations of black-box function U_f for Deutsch's problem. Top output is $|y \oplus f(x)\rangle$, and bottom output is $|x\rangle$.

remain $|-\rangle$ and the bottom output will be $|x\rangle$ if $f(x) = 0$ and will be $-|x\rangle$ if $f(x) = 1$. Let's do the math.

First, consider what happens if we flip $|-\rangle$ using the X gate:

$$\begin{aligned} X|-\rangle &= \frac{1}{\sqrt{2}} X(|0\rangle - |1\rangle) \\ &= \frac{1}{\sqrt{2}} (|1\rangle - |0\rangle) \\ &= -|-\rangle . \end{aligned} \tag{1.68}$$

In the cases where U_f flips $|y\rangle$, we have:

$$U_f |x\rangle |-\rangle = |x\rangle (-|-\rangle) = (-|x\rangle) |-\rangle . \tag{1.69}$$

It might seem that we "magically" moved the minus sign from one qubit to the other, but remember that this is really a two-qubit state, not two individual qubits. And -1 is just a constant, so we can associate it with either part of the two-qubit state.

Figure 1.11 shows the under-the-covers implementation of the circuit from Figure 1.8, where the dashed box replaces U. We add our ancilla qubit as part of this circuit. We typically assume that qubits are initialized in state $|0\rangle$, so we use HX to create the desired $|-\rangle$ state. Then we have our black-box function, U_f, which has the desired sign-flipping effect on $|x\rangle$. Finally, it is good practice to "uncompute" the ancilla bit, restoring it $|0\rangle$, so that the qubit can be used elsewhere in a larger circuit.

If the notions of reversible logic and phase kickback seem strange, don't worry. We'll have a lot more to say about them in Chapters 11 and 12.

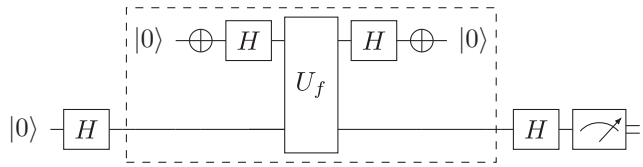


Figure 1.11 Implementation of Deutsch's algorithm. The dashed box is equivalent to U in Figure 1.8.

1.9 Key Characteristics of Quantum Computing

The Bell and Deutsch examples have illustrated several important characteristics of quantum computing:

- *Results are (usually) statistical*: When a classical program is executed, you obtain the same result each time. However, when a quantum circuit is executed, multiple results are possible with probabilities determined by the magnitude squared of the amplitudes of each state. Consequently, in general a circuit must be executed a large number of times, with the results of the computation extracted from a histogram of the measured outcomes.

  It is possible, however, for the outcome to be one particular state with probability 1. This is the case for Deutsch's algorithm, for example. We know the answer for certain after one execution (assuming the quantum machine is error-free). So it's not the case that a quantum algorithm *must* be probabilistic, but the most interesting algorithms tend to be that way.
- *Quantum parallelism*: Arranging for the input state to be a superposition allows the calculation to consider multiple cases at once. However, it is not as easy to capitalize on this as it might sound. As indicated in the previous bullet, even though the output state may contain the complete solution, a single measurement will yield only one state with a probability given by the squared magnitude of the amplitude of the state in the solution of the problem.
- *Exponential scaling*: The number of cases that can be considered scales as 2^N, where N is the number of qubits. Beyond about 50 qubits, a general quantum processor cannot be simulated by a supercomputer; said differently, a processor with of order 50 or more qubits is capable of computations not possible on the best classical computers. However, if the quantum program is restricted to certain types of gates, then the operation of the quantum computer can be efficiently simulated by a classical computer.
- *Quantum interference*: When multiple cases are considered using superposition, the goal of the quantum circuit is to arrange for the amplitudes of correct answer(s) to add constructively, while arranging for the incorrect answer(s) to add destructively.
- *Asking the right question*: Although the output state of a quantum calculation will generally contain information about many possibilities, making a measurement collapses the state and therefore only gives a single result. In the Deutsch Problem, two classical function calls would not only tell you whether the function was constant or balanced, but it would tell you precisely what the function was. In contrast, the quantum calculation answers the question about whether the function is constant or balanced in one function call (which requires consideration of both cases), but it does not tell you which of the two possible functions it is.

1.10 Quantum Computing Systems

At this point, you may be asking: what kind of physical system exhibits the behavior that we can exploit for quantum computing? Any two-state quantum mechanical system can represent a qubit, and there are several possibilities, such as the spin of an electron, the polarization of a photon, or the energy level of an electron in a charged ion. Any of these systems can be used to build a quantum computer, but there are tradeoffs

regarding how the qubits can be manufactured and controlled, and how they interact with one another.

In this book, we concentrate on one specific technology for creating qubits and quantum computing systems: superconducting circuits. Unlike many competing technologies, superconducting qubits are macroscopic in size and are based on well-known nanofabrication technologies. They represent the current technology of choice for several companies building quantum computer systems, including IBM, Google, and Rigetti.

A large part of this book, Chapters 2–8, is devoted to a detailed explanation of these devices and how to control them to carry out the fundamental operations of a quantum computer, described above. In this section, before diving into the details, we provide a high-level overview of a superconducting quantum computer.

As we will see in Chapter 2, we will need to couple the qubit to a signal whose frequency depends on the energy difference between the $|0\rangle$ and $|1\rangle$ states, i.e., the ground and excited states. In superconducting quantum computers, this energy difference corresponds to a microwave frequency near 5 GHz. Consequently, we must design a microwave system to control and measure superconducting qubit states.

The general features of the microwave system to control and readout superconducting qubits are shown in Figure 1.12. A key feature is that the qubits must be held at a very low temperature, near absolute zero. Consequently the qubits must be located in a cryogenic refrigerator. To understand why this is necessary, we want to make sure that if we put the qubit in the ground state, it stays in the ground state. In other words, we want to make sure that it is unable to absorb enough energy incidentally from its environment to make a transition. A circuit in equilibrium at temperature T can emit and absorb photons with the energy kT, where k is Boltzmann's constant. The energy of a photon is $\hbar\omega$, where ω is the frequency and $\hbar$ is Planck's constant divided by 2π. We want to make sure that $kT << \hbar\omega$. For $\omega/(2\pi) = 5$ GHz, this means that $T << 0.24$ K. In state-of-the-art dilution refrigerators, the temperature of the qubits can be held at 10–15 mK. In this range of temperatures, thermal excitation of 5 GHz qubits can be neglected.

Referring again to Figure 1.12, the quantum processor (QP) containing the qubits is located at the bottom left of the refrigerator. In addition to being kept very close to absolute zero temperature, the quantum processor is also sensitive to stray magnetic fields, so it is further placed inside a magnetic shield within the coldest stage of the refrigerator.

The round component just above the quantum processor is a *circulator*. This is a non-reciprocal microwave component in which energy can only propagate between ports in the direction of the circulating arrow. By non-reciprocal, we mean that the behavior of the component is different if you interchange the input with the output. For example, the RF signal from the control electronics enters the circulator at the top port; the energy "circulates" around to the port to which the quantum processor is connected. Any reflected energy from the quantum processor, e.g., containing information about the state of a qubit, then re-enters the bottom port of the circulator. However, since the circulator is non-reciprocal, instead of returning to the input port on the top side of the circulator, it flows instead to the port on the right and is conveyed to the circulator located in the center. We will return to this center circulator in a moment, but let us first consider the chain of coaxial cables and attenuators conveying the control signal to the first circulator.

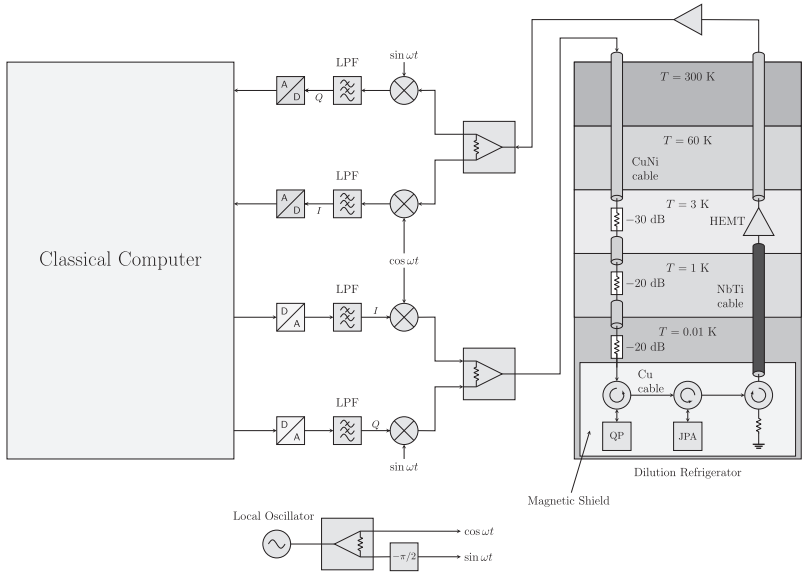


Figure 1.12 System diagram for a superconducting quantum computer.

If we simply used a copper coaxial cable to carry the signal from the room-temperature electronics into the refrigerator, we would have at least two significant problems. First, copper is a good conductor of heat as well as electricity, so the copper cable would convey heat from the upper stages to the lower stages, making it very difficult to reach the temperatures required at the lower stages. To address this, coaxial cables made of an alloy of copper and nickel are used instead. This alloy has very low thermal conductivity to assist with thermally isolating the stages, while having an acceptable electrical conductivity.

The second problem is that a cable coming straight from the outside environment would convey significant noise from the environment into the refrigerator. To combat this, attenuators are placed in the lower stages. These attenuators reduce noise power from the upper stages, but introduce their own noise proportional to their equilibrium temperature. Consequently at the lowest stage, the thermal noise is minimized by the very low temperature of the attenuator. Of course these attenuators also reduce the amplitude of the control signal, so we must make sure that the signal level produced by the signal source is strong enough to produce a satisfactory control signal at the quantum processor.

Returning to the signal reflected from the quantum processor, upon entering the center circulator, the energy is transferred to the bottom port and delivered to a Josephson Junction Parametric Amplifier (JPA). This is a quantum-limited amplifier, meaning that the noise it introduces to the circuit is close to the fundamental minimum allowed by quantum theory. The JPA works in reflection mode, so the amplified reflected signal is returned to the circulator and transferred to the final circulator at the bottom right.

The circulator at the bottom right is operated as an *isolator*. Power applied to the input (left port) is delivered to the output (top port), but any power reflected from

impedance mismatches further down line, e.g., at the input of the HEMT amplifier, will be delivered to the matched load attached to the bottom port. In this way, the very sensitive quantum processor and quantum limited amplifier are isolated from reflected power or noise from the upper stages.

The measured signal at this point is very weak, and we definitely would not want to use a stack of attenuators on the output line! Instead, a coaxial cable made of an alloy of Niobium and Titanium (NbTi) is used to convey the signal to the 3 K stage. NbTi is a Type II superconductor with a transition temperature of about 10 K, so it provides an extremely low loss path to the 3 K stage. At the 3 K stage, there is a more conventional low-noise amplifier (LNA), but made using a high electron mobility transistor (HEMT)—a type of transistor that is known for producing very low noise. The signal from the output of the HEMT LNA is further amplified at room temperature before delivery to the signal processing electronics.

Returning to the control-signal electronics, any band-limited signal centered on the frequency ω can be represented by sine and cosine components:

$$f(t) = I(t)\cos\omega t + Q(t)\sin\omega t, \tag{1.70}$$

where $I(t)$ and $Q(t)$ are "in-phase" and "quadrature" functions whose time variations are slow compared with the period $T = 2\pi/\omega$. Digital samples of I and Q are converted to band-limited analog signals by digital-to-analog converters followed in general by low-pass filters to eliminate high-frequency components resulting from aliasing. The circular components in Figure 1.12 containing "×" are called *mixers*, and produce an output that is simply the product of the two input signals. For example, the bottom-most mixer produces $Q(t)\sin\omega t$ on its output. Similarly, the output of the next-to-bottom mixer is $I(t)\cos\omega t$. The outputs of the two mixers are applied to a power combiner, shown as the square component with two terminals on the left, and one on the right. The output of the combiner is the sum of the two input signals, creating the desired general signal of the form given by Eq. (1.70). This circuit is referred to as an *IQ modulator*.

There are different ways of combining RF signals, but the particular one shown is known as a Wilkinson combiner, or Wilkinson divider. Note that this component *is* reciprocal, so it can be used either as a combiner or a divider. An attractive feature of the Wilkinson circuit when used as a divider is that if all of the ports are matched, it does not introduce any loss.

The output signal from the quantum processor is applied to a signal processing circuit at the top center of Figure 1.12 that is very similar to the IQ modulator used to generate the control signal. The signal is first split into equal parts using a Wilkinson divider, and the divided signals are multiplied by either $\cos\omega t$ or $\sin\omega t$ before being applied to low-pass filters. The effect of the low-pass filters is to integrate the applied signal, so since the cross term proportional to $\cos\omega t\sin\omega t$ averages to zero, the filter outputs are proportional to either $I(t)$ or $Q(t)$. These signals are then digitized and analyzed by the classical computer.

With this high-level description as motivation, we are now ready to discuss in detail the principles that underlie the hardware and software of superconducting quantum computing.

- Chapters 2 and 3 explore the quantum physics that determine the behavior of qubits and gates. This gives us the tools to understand the fundamentals of quantum states and how they can be manipulated.
- Chapters 4–8 show how qubits are constructed from superconducting devices, how they are coupled to each other with microwave transmission lines, and how they are controlled and measured by external systems.
- Chapters 9 and 10 discuss how imperfections in the systems we build affect the quantum information we are trying to process. In the near term, we need mechanisms to characterize and compensate for errors, while the long-term hope is that we will have a sufficient number of high-quality qubits to correct errors dynamically and sustain long-running, fault-tolerant quantum computations.
- Finally, Chapters 11 and 12 describe the computations that can be accomplished using qubits and gates, and the potential for applications beyond the capabilities of classical computers.

Our hope is to lay a firm foundation for those new to the quantum computing field, whether students or practicing engineers, as a first step toward tackling the many research and engineering challenges that are needed to make large-scale quantum computers a reality.

1.11 Exercises

1.1 How many basis states are there for a three-qubit system? Show the column vectors for the (computational) basis states.

1.2 There are four Bell states that can be created by entangling two qubits. In addition to the state shown in Figure 1.5, the three additional states are listed below. Construct a circuit for generating each state.

(a) $\frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)$

(b) $\frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$

(c) $\frac{1}{\sqrt{2}}(|10\rangle - |01\rangle)$

1.3 Prove the following equivalencies.

(a) $HZH = X$

(b) $HXH = Z$

(c) $HYH = -Y$

(d) $\text{CNOT}_{1,0} = H^{\otimes 2}\,\text{CNOT}_{0,1}\,H^{\otimes 2}$
In (d), $\text{CNOT}_{i,j}$ means a CNOT with qubit i as the control and qubit j as the target. $H^{\otimes 2}$ means a Hadamard gate applied to both qubits.

1.4 Create a quantum circuit that swaps two qubit states. In other words: $|ab\rangle \mapsto |ba\rangle$. *Hint:* Consider this classical algorithm that swaps two numbers x and y using an exclusive-OR (XOR) instruction.

$$x \leftarrow x \text{ XOR } y$$

$$y \leftarrow y \text{ XOR } x$$

$$x \leftarrow x \text{ XOR } y$$

1.5 Suppose we design a superconducting qubit where the energy difference between $|0\rangle$ and $|1\rangle$ is around 10 GHz. What is the temperature needed to minimize the effect of thermal energy on the qubit, assuming that the qubit is in thermal equilibrium with its environment?

1.6 Consider the following three-qubit quantum state:

$$|q_2 q_1 q_0\rangle = \frac{|010\rangle + |101\rangle + |001\rangle + |110\rangle}{2}$$

Is qubit q_2 entangled with the other two qubits? Explain why or why not.

1.7 Suppose we have a way of measuring a qubit in the $|+\rangle$ and $|-\rangle$ basis. As a reminder:

$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \quad |-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

(a) Given a qubit in the $|0\rangle$ state, what is the probability of measuring $|+\rangle$? (*Hint:* Use the Born Rule.)

(b) Given a qubit in the $|1\rangle$ state, what is the probability of measuring $|+\rangle$?

(c) Given a qubit $|R\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{i}{\sqrt{2}}|1\rangle$, what is the probability of measuring $|-\rangle$?

1.8 Given a qubit

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

(a) What is the probability of measuring $|0\rangle$?

(b) What is the probability of measuring $|+\rangle$?

(c) What is the probability of measuring $|-\rangle$?