Chapter 5

Architecture

Noson S. Yanofsky
©March 24, 2007
noson@sci.brooklyn.cuny.edu

From the intrinsic evidence of his creation, the Great Architect of the Universe now begins to appear as a pure mathematician.
Sir James Jeans
Mysterious Universe

Now that we have the mathematical and physical preliminaries under our belt, we can move on to the nuts and bolts of quantum computing. At the heart of a computer is the notion of a bit. Quantum computers use a generalization of the concept of a bit called a qubit which shall be discussed in section 5.1. Bits are manipulated by classical (logical) gates. In section 5.2, classical gates are presented from a new and different perspective. From this angle, it is easy to formulate the notion of quantum gates which manipulate qubits. As mentioned in chapters 3 and 4, the evolution of a quantum system is reversible, i.e., manipulations that can be done must also be able to be undone. This "undoing" translates into reversible gates. In section 5.3, we deal with reversible gates and then move on to conclude with quantum gates in section 5.4.

....
Reader's Tip. Discussion of actual physical implementation of qubits and quantum gates will be dealt with in chapter 11. □
....

5.1 Bits and Qubits

What is a **bit**? A bit is an atom of information that represents one of two disjoint situations. There are many examples of bits:

- A bit is electricity traveling through a circuit or not (or high and low.)
- A bit is a switch turned on or off.
- A bit is a way of denoting "true" or "false."

All these examples are saying the same thing: a bit is a way of describing a system whose set of states is of size two. We usually write these two possible states as 0 and 1, or

or F and T, etc.

Since we have become adept at matrices, let us use them as a way of representing a bit. We shall represent 0–or better the state $|0\rangle$–as a two by one matrix with a 1 in the 0's row and a 0 in the 1's row:

$$|0\rangle \quad = \quad \begin{array}{c|c} 0 & \multirow{2}{*}{} \\ 1 & \end{array} \begin{array}{c} 0 \\ 1 \end{array} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{5.1}$$

We shall represent a 1, or state $|1\rangle$ as:

$$|1\rangle \quad = \quad \begin{array}{c} 0 \\ 1 \end{array} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{5.2}$$

Since these are two different representations (indeed orthogonal), we have an honest-to-goodness bit. We will explore how to manipulate these bits in section 5.2.

A bit is either in state $|0\rangle$ or in state $|1\rangle$, which was sufficient for the classical world. Either electricity is running through a circuit or it is not. Either a switch is on or it is off. Either a proposition is true or it is false. But either/or is not sufficient in the quantum world. In that world, there are situations where we are in one state *and* in the other simultaneously. In the realm of the quantum world, there are systems where a switch is both on *and* off. One quantum system can be in state $|0\rangle$ *and* in state $|1\rangle$. Hence we are led to the definition of a qubit:

Definition 5.1.1 *A* **quantum bit** *or a* **qubit** *is a way of describing a quantum system of dimension two.*

We shall represent a qubit as a two by one matrix with complex numbers

$$\begin{array}{c} 0 \\ 1 \end{array} \begin{bmatrix} c_0 \\ c_1 \end{bmatrix}, \tag{5.3}$$

where $|c_0|^2 + |c_1|^2 = 1$. Notice that a classical bit is a special type of qubit. $|c_0|^2$ is to be interpreted as the probability that after measuring the qubit, it will be found in state $|0\rangle$. $|c_1|^2$ is to be interpreted as the probability that after measuring the qubit it will be found in state $|1\rangle$. Whenever we measure a qubit, it automatically becomes a bit. So we shall never "see" a general qubit. Nevertheless, they do exist and are the main characters in our tale. We might visualize this "collapsing" of a qubit to a bit as

$$\begin{tikzcd} & {[1,0]^T} \\ {[c_0, c_1]^T} \arrow[ur, rightsquigarrow] \arrow[dr, rightsquigarrow] & \\ & {[0,1].^T} \end{tikzcd} \tag{5.4}$$

Following the normalization procedures that we learned in chapter 4 on page ???, any element of $\mathbb{C}^2$ can be converted into a qubit. For example, the vector

$$V = \begin{bmatrix} 5 + 3i \\ 6i \end{bmatrix} \tag{5.5}$$

has norm

$$|V| = \sqrt{\langle V, V\rangle} = \sqrt{[5-3i, -6i]\begin{bmatrix} 5+3i \\ 6i \end{bmatrix}} = \sqrt{34+36} = \sqrt{70}. \tag{5.6}$$

So V describes the same physical state as the qubit

$$\frac{V}{\sqrt{70}} = \begin{bmatrix} \frac{5+3i}{\sqrt{70}} \\ \frac{6i}{\sqrt{70}} \end{bmatrix}. \tag{5.7}$$

After measuring the qubit $\frac{V}{\sqrt{70}}$, the probability of it's being found in state $|0\rangle$ is $\frac{34}{70}$, and the probability of it's being found in state $|1\rangle$ is $\frac{36}{70}$.

Exercise 5.1.1 *Normalize* $V = \begin{bmatrix} 15-3.4i \\ 2.1-16i \end{bmatrix}$.

It is easy to see that the bits $|0\rangle$ and $|1\rangle$ are the canonical basis of $\mathbb{C}^2$. So any qubit can be written as

$$\begin{bmatrix} c_0 \\ c_1 \end{bmatrix} \quad = \quad c_0 \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} + c_1 \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad = \quad c_0|0\rangle + c_1|1\rangle. \tag{5.8}$$

Exercise 5.1.2 *Write* $V = \begin{bmatrix} 5+3i \\ 6i \end{bmatrix}$ *as a sum of* $|0\rangle$ *and* $|1\rangle$.

Let us look at several ways of writing different qubits. $\frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ can be written as

$$\begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}. \tag{5.9}$$

Similarly $\frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ -1 \end{bmatrix}$ can be written as

$$\begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{-1}{\sqrt{2}} \end{bmatrix} = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}. \tag{5.10}$$

It is important to realize that

$$\frac{|0\rangle + |1\rangle}{\sqrt{2}} = \frac{|1\rangle + |0\rangle}{\sqrt{2}}. \tag{5.11}$$

These are both ways of writing $\begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$. In contrast,

$$\frac{|0\rangle - |1\rangle}{\sqrt{2}} \neq \frac{|1\rangle - |0\rangle}{\sqrt{2}} \tag{5.12}$$

The first ket is the vector $\begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix}$ and the second ket is the vector $\begin{bmatrix} -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$. However, the two kets are related:

$$\frac{|0\rangle - |1\rangle}{\sqrt{2}} = (-1)\frac{|1\rangle - |0\rangle}{\sqrt{2}}. \tag{5.13}$$

How are qubits to be implemented? In chapter 11, several different methods will be explored. We might simply state some examples of qubit implementations for the time being:

- An electron might be in one of two different orbits around a nucleus of an atom (ground state and excited state).
- A photon might be in one of two polarized states.
- A subatomic particle might have one of two spin directions.

There will be enough quantum indeterminacy and quantum superposition effects within all these systems to represent a qubit.

Computers with only one bit of storage are not very interesting. Similarly, we will need quantum devices with more than one qubit. Consider a byte, or eight bits. A typical byte can look like

$$01101011. \tag{5.14}$$

If we were to follow the way of describing bits as we did above, we would represent the bits as follows:

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{5.15}$$

We learned previously, that in order to combine systems, one should use the tensor product and so we can describe the above byte as

$$|0\rangle \otimes |1\rangle \otimes |1\rangle \otimes |0\rangle \otimes |1\rangle \otimes |0\rangle \otimes |1\rangle \otimes |1\rangle. \tag{5.16}$$

As a qubit, this is an element of

$$\mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \mathbb{C}^2. \tag{5.17}$$

This can be written as a vector space in shorthand as $(\mathbb{C}^2)^{\otimes 8}$. This is a complex vector space of dimension $2^8 = 256$. Since there is only one complex vector space (up to isomorphism) of this dimension, this vector space is isomorphic to $\mathbb{C}^{256}$.

We can describe our byte in yet another way: As a $2^8 = 256$ row vector

$$\begin{array}{c} 00000000 \\ 00000001 \\ \vdots \\ 01101010 \\ 01101011 \\ 01101100 \\ \vdots \\ 11111110 \\ 11111111 \end{array} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}. \tag{5.18}$$

Exercise 5.1.3 *Express the three bits* 101 *or* $|1\rangle \otimes |0\rangle \otimes |1\rangle \in \mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \mathbb{C}^2$ *as a vector in* $(\mathbb{C}^2)^{\otimes 3} = \mathbb{C}^8$. *Do the same for* 011 *and* 111.

This is fine for the classical world. However, for the quantum world, a generalization is needed: every eight qubits can be written as

$$\begin{array}{c} 00000000 \\ 00000001 \\ \vdots \\ 01101010 \\ 01101011 \\ 01101100 \\ \vdots \\ 11111110 \\ 11111111 \end{array} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{106} \\ c_{107} \\ c_{108} \\ \vdots \\ c_{254} \\ c_{255} \end{bmatrix} \tag{5.19}$$

where $\sum_{i=0}^{255} |c_i|^2 = 1$.

In the classical world, you need to write the state of each bit of a byte which amounts to writing eight bits. In the quantum world, a state of eight qubits is given by writing 256 complex numbers. As we stated in section 3.4, this exponential growth was one of the reasons why researchers started thinking about quantum computing. If one wanted to emulate a quantum computer with a 64 qubit register, one would need to store $2^{64} = 18,446,744,073,709,551,616$ complex numbers. This is beyond our current ability.

Let us practice writing two qubits in ket notation. The qubit pair can be written as

$$|0\rangle \otimes |1\rangle. \tag{5.20}$$

Since the tensor product is understood, we might also write these qubits as $|0,1\rangle$ or $|01\rangle$. We might also look at these qubits as the four by one matrix

$$\begin{array}{c} 00 \\ 01 \\ 10 \\ 11 \end{array} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}. \tag{5.21}$$

Exercise 5.1.4 *What vector corresponds to the state* $3|01\rangle + 2|11\rangle$*?*

The qubit corresponding to

$$\frac{1}{\sqrt{3}} \begin{bmatrix} 1 \\ 0 \\ -1 \\ 1 \end{bmatrix} \tag{5.22}$$

can be written as

$$\frac{1}{\sqrt{3}}|00\rangle - \frac{1}{\sqrt{3}}|10\rangle + \frac{1}{\sqrt{3}}|11\rangle = \frac{|00\rangle - |10\rangle + |11\rangle}{\sqrt{3}}. \tag{5.23}$$

The tensor product of two states is not commutative.

$$|0\rangle \otimes |1\rangle = |0,1\rangle = |01\rangle \neq |10\rangle = |1,0\rangle = |1\rangle \otimes |0\rangle. \tag{5.24}$$

The first ket describes the state where the first qubit is in state 0 and the second qubit is in state 1. The second ket says that first qubit is in state 1 and the second state is in state 0.

5.2 Classical Gates

Classical logical gates are ways of manipulating bits. Bits enter and exit logical gates. We shall need ways of manipulating qubits and will therefore study classical gates from the point of view of matrices. As stated in section 5.1, we represent n input bits as a 2^n by 1 matrix and m output bits as a 2^m by 1 matrix. How should we represent our logical gates? When one multiplies a 2^m by 2^n matrix with a 2^n by 1 matrix, the result is a 2^m by 1 matrix. In symbols:

$$(2^m \text{ by } 2^n) \star (2^n \text{ by } 1) = (2^m \text{ by } 1). \tag{5.25}$$

So bits will be represented by column vectors and logic gates by matrices.

Let us try a simple example. Consider the NOT gate

NOT takes as input one bit, or a 2 by 1 matrix, and outputs one bit, or a 2 by 1 matrix. NOT of $|0\rangle$ equals $|1\rangle$ and NOT of $|1\rangle$ equals $|0\rangle$. Consider the matrix

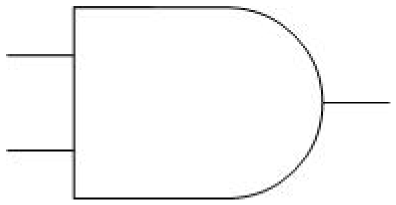
$$\text{NOT} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}. \tag{5.26}$$

This matrix satisfies

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \qquad \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \tag{5.27}$$

which is exactly what we want.

What about the other gates? Consider the AND gate. The AND gate is different from the NOT gate because AND accepts two bits and outputs one bit.



Since there are two inputs and one output, we will need a 2^1 by 2^2 matrix. Consider the matrix

$$\text{AND} = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{5.28}$$

This matrix satisfies

$$\begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{5.29}$$

We can write this as

$$\text{AND}|11\rangle = |1\rangle. \tag{5.30}$$

In contrast, consider another 4 by 1 matrix:

$$\begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}. \tag{5.31}$$

We can write this as

$$\text{AND}|01\rangle = |0\rangle. \tag{5.32}$$

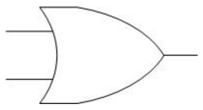
Exercise 5.2.1 *Calculate AND$|10\rangle$.*

What would happen if we put an arbitrary 4 by 1 matrix to the right of AND?

$$\begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 3.5 \\ 2 \\ 0 \\ -4.1 \end{bmatrix} = \begin{bmatrix} 5.5 \\ -4.1 \end{bmatrix} \tag{5.33}$$

This is clearly nonsense. We are only permitted to multiply these classical gates with vectors that represent classical states, i.e., column matrices with a single 1 entry and all the other entries must be 0. In the classical world, the bits are only one state and are described by such vectors. Only later, when we delve into quantum gates will we have more room (and more fun).

The OR gate

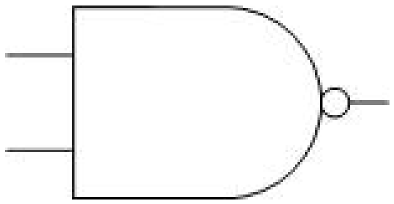


can be represented by the matrix

$$\text{OR} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix}. \tag{5.34}$$

Exercise 5.2.2 *Show that this matrix performs the OR operation.*

The NAND gate

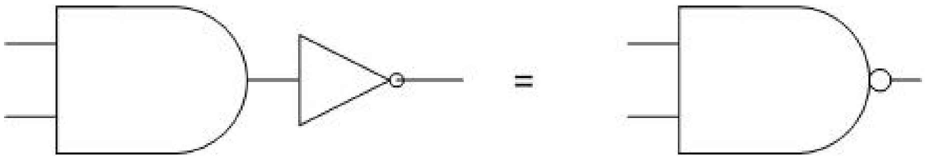


is of special importance because every logical gate can be made out of NAND gates. Let us try to work out which matrix would correspond to NAND. One way is to sit down and consider for which of the four possible input states of two bits (00,01,10,11) does NAND output a 1 (answer: 00,01,10) and in which states does NAND output a 0 (answer: 11). From this we realize that NAND can be written as

$$\text{NAND} = \begin{array}{c} \\ 0 \\ 1 \end{array} \begin{array}{c} \begin{array}{cccc} 00 & 01 & 10 & 11 \end{array} \\ \left[\begin{array}{cccc} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{array}\right] \end{array}. \tag{5.35}$$

Notice that the column names correspond to the inputs and the row names correspond to the outputs. 1 in the jth column and ith row means that on entry j the matrix/gate will output i.

There is, however, another way in which one can determine the NAND gate. The NAND gate is really the AND gate followed by the NOT gate.

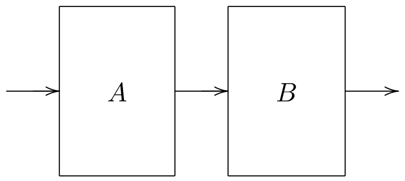


In other words, we can perform the NAND operation by first performing the AND operation and then the NOT operation. In terms of matrices we can write this as

$$NOT \star AND = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \star \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} = \text{NAND}. \tag{5.36}$$

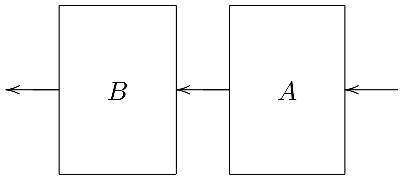
Exercise 5.2.3 *Find a matrix that corresponds to NOR.*

This way of thinking of NAND rings to light a general situation. When we perform a computation, often we have to carry out one operation followed by another.



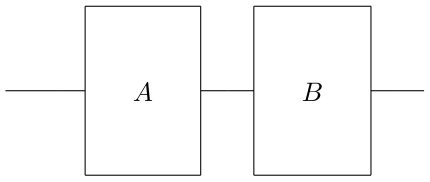
(5.37)

We call this performing **sequential** operations. If matrix A corresponds to performing an operation and matrix B corresponds to performing another operation, then the multiplication matrix $B \star A$ corresponds to performing the operation sequentially. Notice that $B \star A$ looks like the reverse of our picture which has, from left to right, A and then B. Do not be alarmed by this. The reason for this is because we read from left to right and hence we depict processes as going from left to right. We could have easily drawn the above figure as



(5.38)

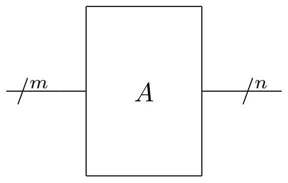
with no confusion. [1] We shall follow the convention that computation flows from left to right and leave out the arrows. And so a computation of A followed by B shall be denoted



(5.39)

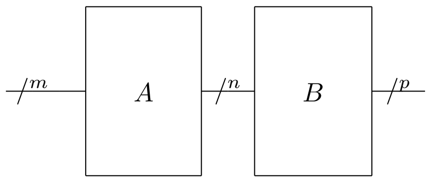
[1]If this text were written in Arabic or Hebrew, this problem would not even arise.

Let us be formal with the number of inputs and the number of outputs. If A is an operation with m input bits and n output bits, then we shall draw this as



(5.40)

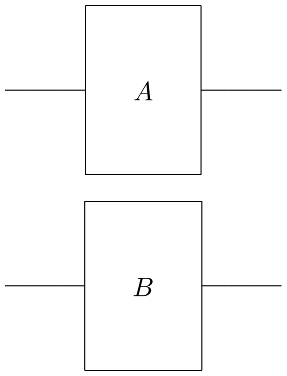
The matrix A will be of size 2^n by 2^m. Say B takes the n outputs of A as input and outputs p bits, i.e.,



(5.41)

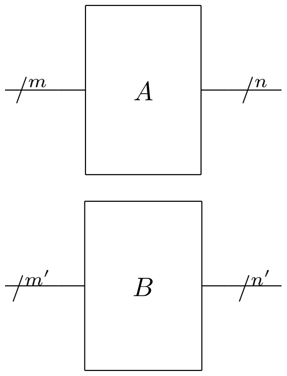
Then B is represented by 2^p by 2^n matrix B, and performing one operation sequentially followed by another operations corresponds to $B \star A$ which is a $(2^p \text{ by } 2^n) \star (2^n \text{ by } 2^m) = (2^p \text{ by } 2^m)$ matrix.

Besides sequential operations, there are **parallel** operations:



(5.42)

Here we are doing A to some bits and B to other bits. This will be represented by $A \otimes B$ (see section 2.7). Let us be exact with the number of inputs and the number of outputs.



(5.43)

A will be of size 2^n by 2^m. B will be of size $2^{n'}$ by $2^{m'}$. Following equation [CITE EQUATION] in section 2.7, $A \otimes B$ is of size $2^n 2^{n'} = 2^{n+n'}$ by $2^m 2^{m'} = 2^{m+m'}$.

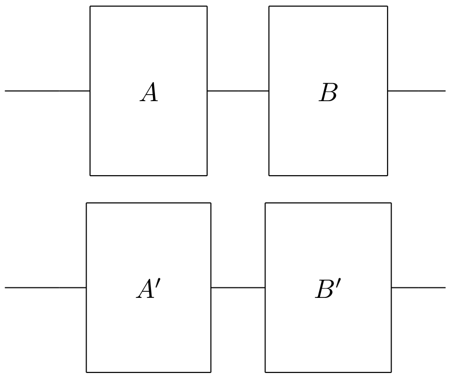
Exercise 5.2.4 *In Exercise CITE EXERCISE we proved that $A \otimes B \cong B \otimes A$. What does this fact correspond to in terms of doing parallel operations to different bits?*

Combination of sequential and parallel operations gates/matrices will be **circuits**. We will, of course, construct some really complicated matrices, but they will all be decomposable into the sequential and parallel composition of simple gates.

Exercise 5.2.5 *In Exercise CITE EXERCISE, we proved that for matrices of the appropriate sizes A, A', B and B' we have the following equation*

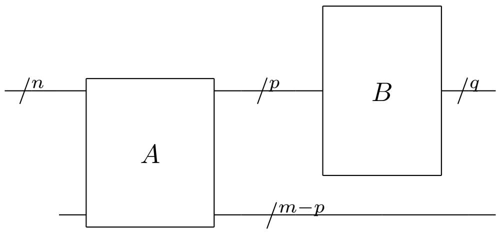
$$(B \otimes B') \star (A \otimes A') \quad = \quad (B \star A) \otimes (B' \star A'). \tag{5.44}$$

What does this correspond to in terms of performing different operations to different (qu)bits? Hint: Consider the following figure



(5.45)

Example 5.2.1 *Let A be an operation that takes n inputs and gives m outputs. Let B take $p < m$ of these outputs and leave the other $m - p$ outputs alone. B outputs q bits*

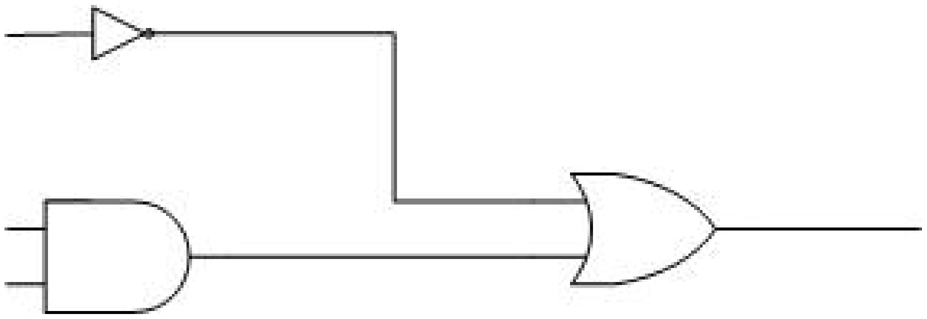


(5.46)

A is a 2^m by 2^n matrix. B is a 2^{m-p} by 2^q matrix. Since nothing should be done to the $m-p$ bits, we might represent this as the 2^{m-p} by 2^{m-p} identity matrix I_{m-p}. We do not draw any gate for the identity matrix. The entire circuit can be represented by the following matrix

$$(B \otimes I_{m-p}) \star A. \tag{5.47}$$

Example 5.2.2 *Consider the circuit*



This is represented by

$$OR \star (NOT \otimes AND). \tag{5.48}$$

Let us see how the operations look like as matrices. Calculating, we get:

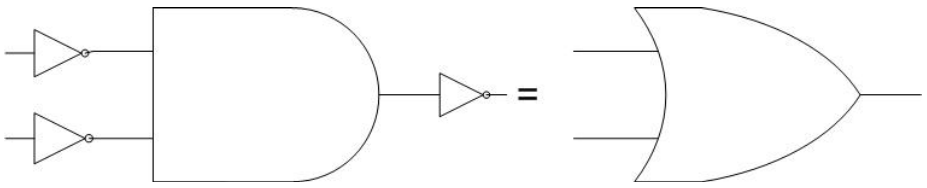
$$NOT \otimes AND = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \otimes \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{5.49}$$

And so we get

$$OR \star (NOT \otimes AND) = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{5.50}$$

Let us see if we can formulate DeMorgan's laws in terms of matrices. One of DeMorgan's law states that $\neg(\neg P \bigwedge \neg Q) = P \bigvee Q$. In pictures this looks like
In terms of matrices this corresponds to

$$\text{NOT} \star \text{AND} \star (\text{NOT} \otimes \text{NOT}) = \text{OR}. \tag{5.51}$$



First, let's calculate the tensor product:

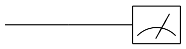
$$\mathrm{NOT} \otimes \mathrm{NOT} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \otimes \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}. \qquad (5.52)$$

This DeMorgan's law corresponds to the following identity of matrices:

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \star \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \star \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix}. \qquad (5.53)$$

Exercise 5.2.6 *Multiply out these matrices and confirm the identity.*

There is one last piece of notation that must be included. Usually at the end of a computation a measurement will be performed. A measurement of qubit(s) shall be denoted with

 (5.54)

Exercise 5.2.7 *Formulate the other DeMorgan's law* $\neg(\neg P \bigvee \neg Q) = P \bigwedge Q$ *in terms of matrices*

Exercise 5.2.8 *Write the matrix that would correspond to a one-bit adder. A one-bit adder adds the bits* x, y *and* c *(a carry-bit from an earlier adder) and outputs the bits* z *and* c' *(a carry-bit for the next adder). There are three inputs and two outputs, so the matrix will be of dimension* 2^2 *by* 2^3. *Hint: Mark the columns as* $000, 001, 010, \ldots, 110, 111$ *where column, say,* 101 *corresponds to* $x = 1$, $y = 0$, $c = 1$. *Mark the rows as* $00, 01, 10, 11$ *where row, say,* 10, *corresponds to* $z = 1$, $c' = 0$. *When* $x = 1$, $y = 0$, $c = 1$, *the output should be* $z = 0$ *and* $c' = 1$. *So place a* 1 *in the row marked* 01 *and a* 0 *in all other rows.*

Exercise 5.2.9 *In Exercise CITE EXERCISE, you wrote the matrix that corresponds to a one-bit adder. Check that your results are correct by writing the circuit in terms of classical gates and then converting the circuit to a big matrix.*

5.3 Reversible Gates

Not all of the logical gates that we dealt with in section 5.2 will work in quantum computers. In the quantum world, all operations that are not measurements are reversible and are represented by a unitary matrix. The AND operation is not reversible. Given an output of $|0\rangle$ from AND, one cannot determine if the input was $|00\rangle$, $|01\rangle$ or $|10\rangle$. So from an output of the AND gate, one cannot determine the input and hence AND is not reversible. In contrast, the NOT gate and the Identity gates are reversible. In fact, they are their own inverses.

$$I \star I = I \qquad \mathrm{NOT} \star \mathrm{NOT} = I_2. \tag{5.55}$$

Reversible gates have a history that predates quantum computing. Our everyday computers lose energy and generate a tremendous amount of heat. In the 1960's Rolf Landauer analyzed computational processes and showed that erasing information is what causes energy loss and heat. This notion has come to be known as the **Landauer Principle**.

In order to gain a real-life intuition of why erasing information dissipates energy, consider a tub of water with a wall separating the two sides as in Figure (5.1).

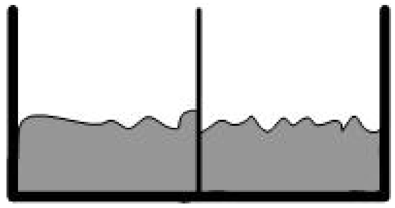


Figure 5.1: Tub with water in no state

This can be used as a way of storing a bit of information. If all the water is pushed to the left, then the system is in state $|0\rangle$ and if all the water is pushed to the right, then the system is in state $|1\rangle$ as in Figure (5.2).

What would correspond to erasing information in such a system system? If there were a hole in the wall separating the 0 and 1 region, then the water could seep out and we would not know what state the system would be in. One can easily place a turbine where the water is seeping out (see Figure (5.3)) and generate energy. Hence loosing information means energy is being dissipating.

Notice also that writing information is a reversible procedure. If the tub is in no state and we push all the water to the right and set the water to state $|0\rangle$, all one needs to do is remove the wall and the water will go into both regions resulting in no state. This is shown in Figure (5.4).

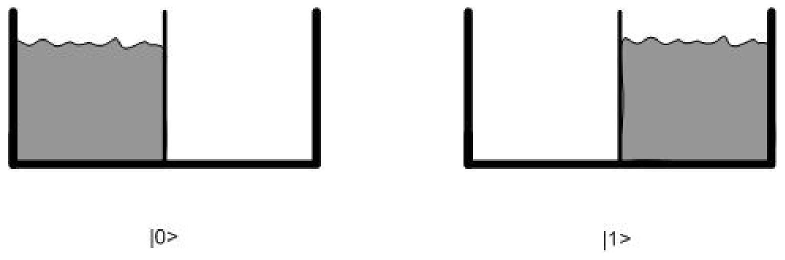


Figure 5.2: Tub with water in state $|0\rangle$ and state $|1\rangle$

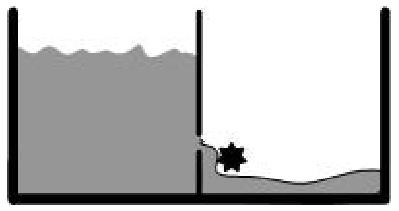


Figure 5.3: State $|0\rangle$ dissipating and creating energy.

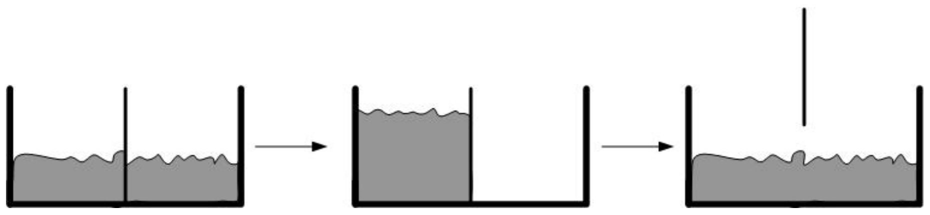


Figure 5.4: Reversibility of writing.

So we have reversed the fact that information was written. In contrast, erasing information is not reversible. Start at state $|0\rangle$, then remove the wall that separates the two parts of the tub. That is erasing the information. How should we go back to the original state? There are two possible states to return to as in Figure(5.5).

The obvious answer is that we should push all the water back to state $|0\rangle$. But the only way we know that $|0\rangle$ is the original state is if that information is copied into your brain. In that case, the system is both the tub and the brain and we did not really erase the fact that state $|0\rangle$ was the original state. Our brain is storing the information.

We can get another intuition of this by considering two people, Alice and Bob. If Alice writes a letter on an empty blackboard and Bob walks into the room, then Bob can erase the letter that Alice wrote on the board and return the blackboard into its original pristine state. Thus, writing is reversible. In contrast, if there is a board with writing on it and Alice erases the board, then when Bob walks into the room he cannot write what Alice had on the board.

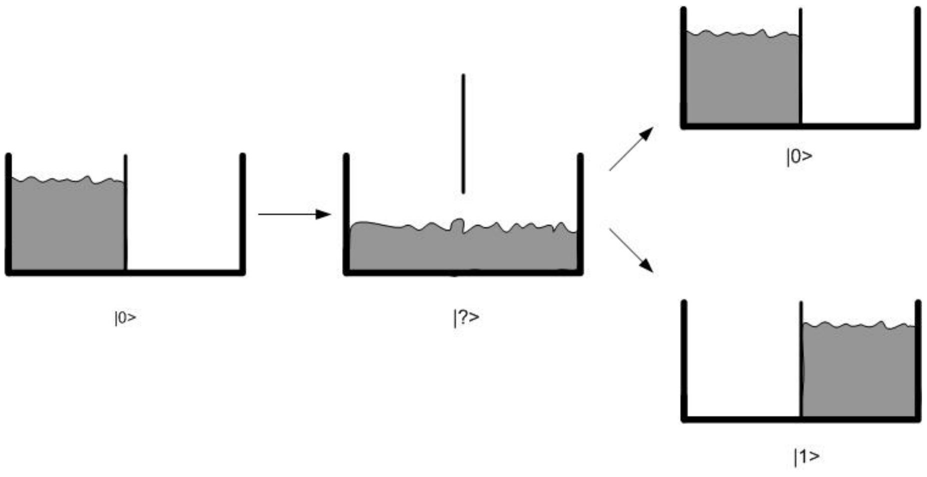
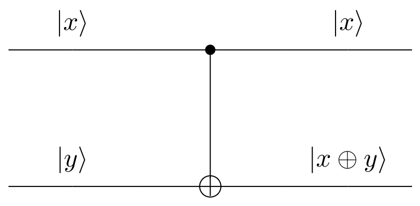


Figure 5.5: Irreversibility of erasing.

Bob does not know what was on the board before Alice erased it (it was not copied to his brain.) So Alice's erasing was not reversible. [2]

We have found that erasing information is an irreversible, energy-dissipating operation. In the 1970's, Charles H. Bennett continued along these lines of thought. If erasing information is the only operation that uses energy, then a computer that does not erase or lose information would be reversible and would not use any energy. Bennett started working on reversible circuits and programs.

What examples of reversible gates are there? We have already seen that the identity gate and NOT gates are reversible. What else is there? Consider the following **controlled-not gate** .



(5.56)

This gate has two inputs and two outputs. The top input is the control bit. It controls what the output will be. If $|x\rangle = |0\rangle$, then the bottom output of $|y\rangle$

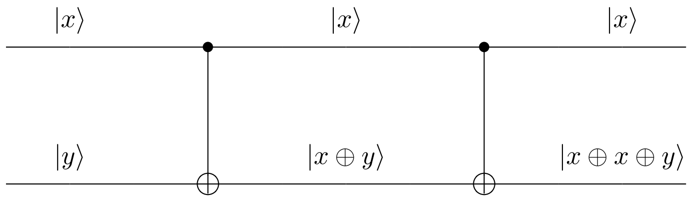
[2]We shall revisit some of these mind-bending ideas in chapter 10.

will be the same as the input. If $|x\rangle = |1\rangle$ then the bottom output will be the opposite. If we write the top qubit first and then the bottom qubit, then the controlled-not gate takes $|x, y\rangle$ to $|x, x \oplus y\rangle$ where $\oplus$ is the binary exclusive or operation.

The matrix that corresponds to this reversible gate is

$$\begin{array}{c c} & \begin{array}{cccc} 00 & 01 & 10 & 11 \end{array} \\ \begin{array}{c} 00 \\ 01 \\ 10 \\ 11 \end{array} & \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \end{array}. \tag{5.57}$$

The controlled-not gate can be reversed by itself. Consider the following figure

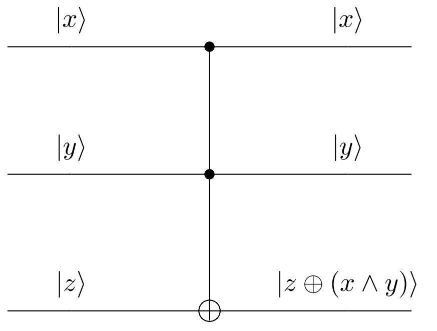


(5.58)

State $|x, y\rangle$ goes to $|x, x \oplus y\rangle$ which further goes to $|x, x \oplus (x \oplus y)\rangle$. This last state is equal to $|x, (x \oplus x) \oplus y\rangle$ because $\oplus$ is associative. Since $x \oplus x$ is always equal to 0, this state reduces to the original $|x, y\rangle$.

Exercise 5.3.1 *Show that the controlled-NOT gate is its own inverse by multiplying the corresponding matrix by itself and getting the identity.*

An interesting reversible gate is the **Toffoli gate**:



(5.59)

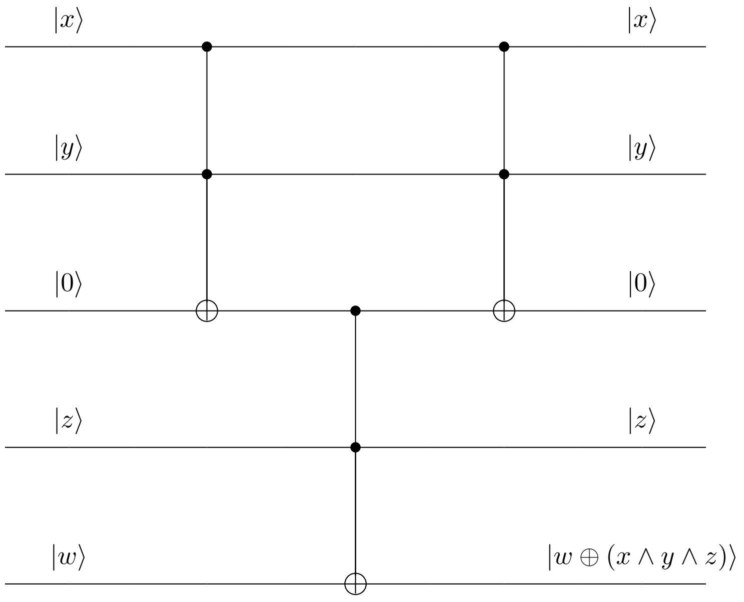
This is similar to the controlled-NOT gate but with two controlling bits. The bottom bit flips only when the top two bits are in state $|1\rangle$. We can write this operation as taking state $|x, y, z\rangle$ to $|x, y, (x \wedge y) \oplus z\rangle$.

Exercise 5.3.2 *Show that the Toffoli gate is its own inverse.*

The matrix that corresponds to this gate is

$$\begin{array}{c} \\ 000 \\ 001 \\ 010 \\ 011 \\ 100 \\ 101 \\ 110 \\ 111 \end{array} \begin{array}{c} \begin{array}{cccccccc} 000 & 001 & 010 & 011 & 100 & 101 & 110 & 111 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array} \tag{5.60}$$

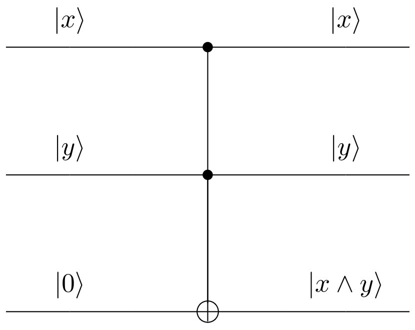
The NOT gate has no controlling bit, the controlled-not gate has one controlling bit, and the Toffoli gate has two controlling bits. Can we go on with this? Yes. A gate with three controlling bits can be constructed from three Toffoli gates as follows.



(5.61)

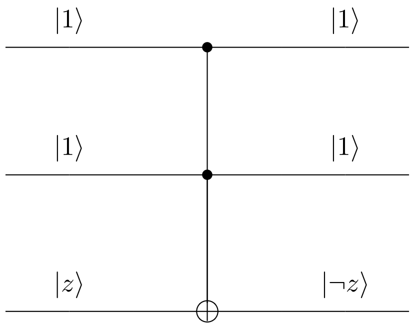
One reason why the Toffoli gate is interesting is that it is universal. In other words, with copies of the Toffoli gate, you can make any logical gate. In particular, you can make a reversible computer using only Toffoli gates. Such a computer would, in theory, neither use any energy nor give off any heat.

In order to see that the Toffoli gate is universal, we shall show that you can make the AND and the NOT gate with it. The AND gate is obtained by setting the bottom z input to $|0\rangle$. The bottom output will then be $|x \wedge y\rangle$.



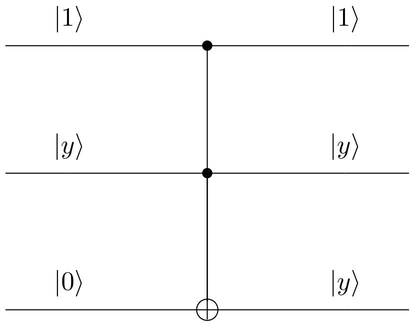
(5.62)

The NOT gate is obtained by setting the top two inputs to $|1\rangle$. The bottom output will be $|(1 \wedge 1) \oplus z\rangle = |1 \oplus z\rangle = |\neg z\rangle$.



(5.63)

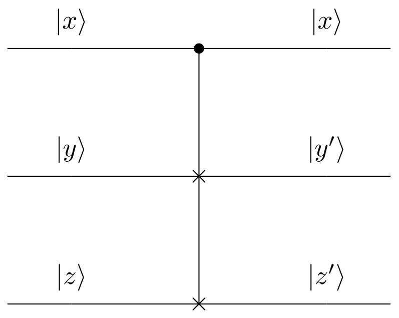
In order to get all gates, we also must have a way of producing a fanout of values. In other words, a gate is needed that inputs a value and outputs two of the same values. This can be obtained by setting x to $|1\rangle$ and z to $|0\rangle$. This makes the output $|1, y, y\rangle$.



(5.64)

Exercise 5.3.3 *Construct the NAND with one Toffoli gate. Construct the NOR and OR gates with two Toffoli gates.*

Another interesting reversible gate is the **Fredkin gate**. The Fredkin gate also has three inputs and three outputs.



(5.65)

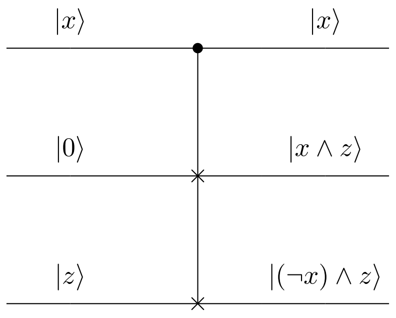
The top $|x\rangle$ input is the control input. The output is always the same $|x\rangle$. If $|x\rangle$ is set to $|0\rangle$, then $|y'\rangle = |y\rangle$ and $|z'\rangle = |z\rangle$, i.e., the values stay the same. If, on the other hand, the control $|x\rangle$ is set to $|1\rangle$, then the outputs are reversed: $|y'\rangle = |z\rangle$ and $|z'\rangle = |y\rangle$. In short $|0, y, z\rangle \mapsto |0, y, z\rangle$ and $|1, y, z\rangle \mapsto |1, z, y\rangle$.

Exercise 5.3.4 *Show that the Fredkin gate is its own inverse.*

The matrix that corresponds to the Fredkin gate is

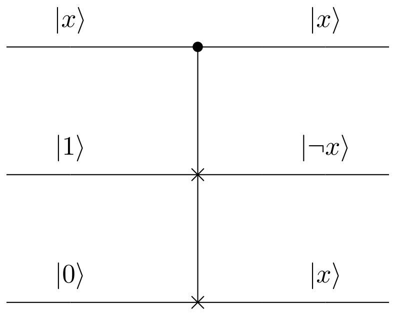
$$\begin{array}{c|cccccccc|}
 & 000 & 001 & 010 & 011 & 100 & 101 & 110 & 111 \\
000 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
001 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
010 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
011 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
100 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
101 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
110 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
111 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{array} . \qquad (5.66)$$

The Fredkin gate is also universal. By setting y to $|0\rangle$ we get the AND gate as follows:



(5.67)

The NOT gate and the fanout gate can be obtained by setting $|y\rangle$ to $|1\rangle$ and $|z\rangle$ to $|0\rangle$. This gives us



(5.68)

So both the Toffoli and the Fredkin gates are universal. Both are not only reversible gates, but a look at their matrices show that they are also unitary. In the next section we shall look at other unitary gates.

5.4 Quantum Gates

A quantum gate is simply any unitary matrix that manipulates qubits. We have already worked with some quantum gates such as the Identity matrix, the Hadamard gate, the NOT gate, the controlled-NOT gate, the Toffoli gate and the Fredkin gate. What else is there?

Let us first concentrate on quantum gates that manipulate a single qubit.

The following three matrices are called Pauli matrices and are very important.

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \qquad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \qquad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \tag{5.69}$$

They occur everywhere in quantum mechanics and quantum computing.[3] Notice that the X matrix is nothing more than our NOT matrix. Other important matrices that will be used are

$$S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}, \qquad T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}. \tag{5.70}$$

There are other quantum gates on single qubits. However these will be enough to tell our tale.

Exercise 5.4.1 *Show that each of these matrices are unitary.*

Exercise 5.4.2 *Show the action of each of these matrices on an arbitrary qubit* $\begin{bmatrix} c_0 \\ c_1 \end{bmatrix}$.

Exercise 5.4.3 *These operations are intimately related to each other. Prove the following relationships between the operations:*

1. $H = \frac{1}{\sqrt{2}}(X + Z)$
2. $X = HZH$
3. $Z = HXH$
4. $-1Y = HYH$
5. $S = T^2$
6. $-1Y = XYX$

There is a beautiful geometric way of representing one-qubit operations. Remember from chapter 1, page CITE PAGE, that for a given complex number $c = x + yi$ whose modulus is 1, there is a nice way of visualizing c as an arrow of length 1 from the origin to the circle of radius 1.

$$|c|^2 = c \times \overline{c} = (x + yi) \times (x - yi) = x^2 + y^2 = 1 = r \tag{5.71}$$

[3] Sometimes the notation σ_x, σ_y and σ_z is used for these matrices.

In other words, every complex number of radius 1 can be identified by the angle ϕ that the vector makes with the x axis.

There is an analogous representation of a qubit as an arrow in three dimensions: let us see how it works. A generic qubit is of the form

$$|\varphi\rangle = c_0|0\rangle + c_1|1\rangle \tag{5.72}$$

where $|c_0|^2+|c_1|^2 = 1$ Although at first sight there are four real numbers involved in the definition above, it turns out that there are only two actual degrees of freedom. Let us see why. To begin with, let us rewrite our qubit in polar form:

$$c_0 = r_0 e^{i\phi_0} \tag{5.73}$$

and

$$c_1 = r_1 e^{i\phi_1} \tag{5.74}$$

and so

$$|\varphi\rangle = r_0 e^{i\phi_0}|0\rangle + r_1 e^{i\phi_1}|1\rangle \tag{5.75}$$

still four real parameters: r_0, r_1, ϕ_0, ϕ_1. However, a quantum physical state does not change if we multiply its corresponding vector by an arbitrary complex number of norm 1 (see chapter 4, page ???), thus we can obtain an equivalent expression for our qubit, where the amplitude for $|0\rangle$ is real, by "killing" its phase:

$$e^{-i\phi_0}|\varphi\rangle = e^{-i\phi_0}(r_0 e^{i\phi_0}|0\rangle + r_1 e^{i\phi_1}|1\rangle = \tag{5.76}$$

$$r_0|0\rangle + r_1 e^{i(\phi_1-\phi_0)}|1\rangle \tag{5.77}$$

now we only have three real parameters, namely r_0, r_1 and $\phi = \phi_1 - \phi_0$. But we can do better: using the constraint above, we have

$$r_0^2 + r_1^2 = 1 \tag{5.78}$$

so we can rename them as

$$r_0 = \cos(\theta) \tag{5.79}$$

and

$$r_1 = \sin(\theta) \tag{5.80}$$

summing up, our qubit is now in the canonical representation

$$(\heartsuit) \qquad |\varphi\rangle = \cos(\theta)|0\rangle + \sin(\theta)e^{i\phi}|1\rangle \tag{5.81}$$

with only two real parameters left!

What is the range of the two angles θ and ϕ? We invite you to show that $0 \le \phi \le 2\pi$ and $0 \le \theta \le \frac{\pi}{2}$ are enough to cover all possible qubits:

Exercise 5.4.4 *Prove that every qubit in the canonical representation* $(\heartsuit)$ *with* $\theta > \frac{\pi}{2}$ *is equivalent to another one where* θ *lies in the first quadrant of the plane. Hint: use a bit of trigonometry and change* ϕ *according to your needs.*

As only two real numbers are necessary to identify a qubit, we can map it to an arrow from the origin to the three-dimensional sphere of R^3 of radius 1, known as the **Bloch Sphere**.

Every qubit can be represented by two angles that describe such an arrow (think of the Bloch sphere as the earth. You just need to specify only two numbers, the latitude and the longitude, to know where you are on the planet). The standard parametrization of the unit sphere is

$$x = \sin\theta\cos\theta \tag{5.82}$$

$$y = \sin\theta\sin\theta \tag{5.83}$$

$$z = \cos\theta \tag{5.84}$$

However, there is yet another final caveat: suppose we use the representation above to map our qubit on the sphere. Then, the points (θ, ϕ) and $(\pi - \theta, \phi + \pi)$ represent the $s$ame qubit, up to the factor -1!

Exercise 5.4.5 *Verify this last statement.*

Conclusion: the parametrization would map the same qubit twice, on the upper hemisphere and on the lower one. To mitigate this problem, we simply double the "latitude" to cover the entire sphere at "half speed":

$$x = \sin 2\theta\cos\theta \tag{5.85}$$

$$y = \sin 2\theta\sin\theta \tag{5.86}$$

$$z = \cos 2\theta \tag{5.87}$$

Now we have it!

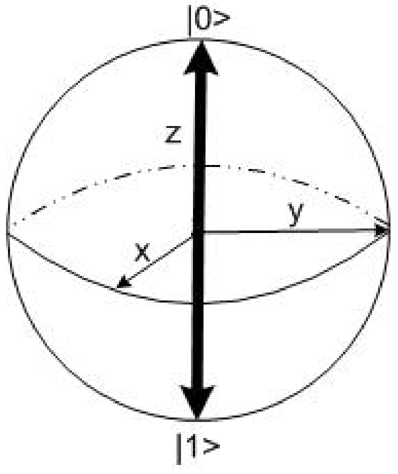


Figure 5.6: Bloch Sphere.

Let us spend a few moments familiarizing ourselves with the Bloch Sphere and its geometry. The north pole corresponds to the state $|0\rangle$ and the south pole corresponds to the state $|1\rangle$. These two points can be taken as the geometrical image of the good ol'-fashioned bit. But there are many more qubits out there, and the Bloch sphere clearly shows it.

Example 5.4.1 *Let us find out which point of the sphere corresponds to the qubit*

$$|q\rangle = \cos(\pi)|0\rangle + \sin(\pi)e^{i\frac{\pi}{2}}|1\rangle \tag{5.88}$$

Using the map we get.....

It is your turn now:

Exercise 5.4.6 *Start from $|q\rangle = -\frac{1}{\sqrt{2}}i|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$. Convert it to the canonical representation and then determine where it goes on the sphere.*

The precise meaning of the two angles in ♡ is the following: φ is half the angle that $|\psi\rangle$ makes from x along the equator and θ is the angle that $|\psi\rangle$ makes with the y axis.

When a qubit is measured in the standard basis it collapses to a bit, or equivalently, to the north or south pole of the Bloch sphere. The probability of which pole the qubit will collapse to depends exclusively on how high or low the qubit is pointing, in other words, to its *latitude.* In particular, if the qubit happens to be on the equator, there is a $50 - 50$ chance of collapsing to $|0\rangle$ and $|1\rangle$. As the angle θ expresses the qubit's latitude, it does control its chance of collapsing north or south.

Exercise 5.4.7 *Consider the qubit in Example 5.4.1. Its θ is equal to $\frac{\pi}{4}$. Change it to $\frac{\pi}{3}$ and picture the result. Then compute its likelihood of collapsing to the south pole after being observed.*

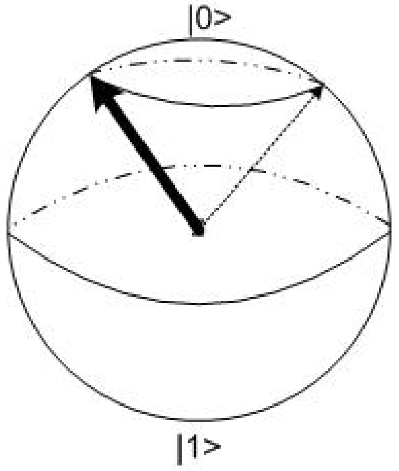
Take an arbitrary arrow and rotate it around the z axis;in the geographical metaphor, you are changing its *l*ongitude:

Notice that the probability of which classical state it will collapse to is not affected. Such a state change is called a **phase change**. In the ♡ representation, this corresponds to altering the phase parameter $e^{i\varphi}$.

Before we move on, one last important thing: just like $|0\rangle$ and $|1\rangle$ sit on opposite sides of the sphere, an arbitrary pair of orthogonal qubits is mapped to antipodal points of the Bloch Sphere.

Exercise 5.4.8 *Show that if a qubit has latitude 2θ and longitude ϕ on the sphere, its orthogonal lives in the antipode $\pi - 2\theta$ and $\pi + \phi$.*

Another reason why the Bloch sphere is interesting is that every unitary 2 by 2 matrix (i.e. a one-qubit operation) can be visualized as a way of manipulating the sphere. We have seen in chapter 2, page ??? that every unitary matrix is an isometry. This means that such a matrix maps qubits to qubits and the



inner product is preserved. Geometrically, this corresponds to a rotation or an inversion of the Bloch sphere.

The X, Y and Z matrices are ways of "flipping" the Bloch sphere 180^0 along the x, y and z axes respectively. Remember that X is nothing other the NOT gate and takes $|0\rangle$ to $|1\rangle$ and vice versa. But it does more, it takes everything above the equator to below the equator. The other Pauli matrices work similarly.

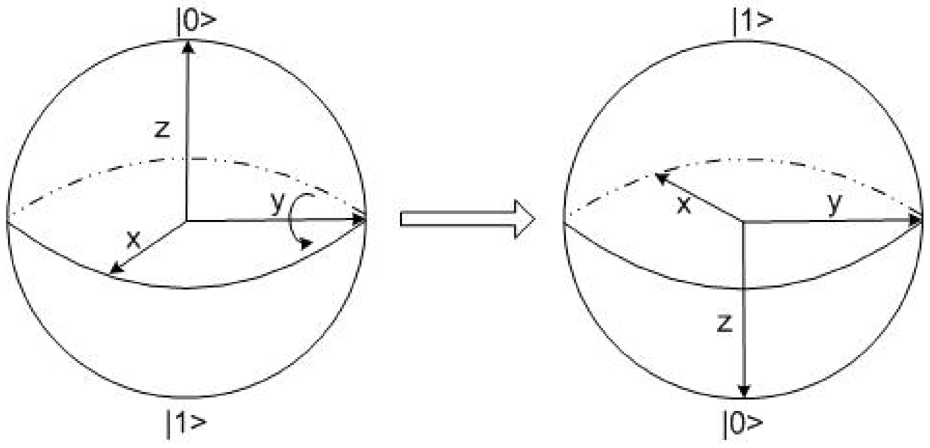


Figure 5.7: A Rotation of the Bloch Sphere at y.

There are times when we are not interested in performing a total flip but just want to turn the Bloch sphere θ degrees along an axis. These three matrices will work.

$$R_x(\theta) = cos\frac{\theta}{2}I - i\ sin\frac{\theta}{2}X = \begin{bmatrix} cos\frac{\theta}{2} & -i\ sin\frac{\theta}{2} \\ -i\ sin\frac{\theta}{2} & cos\frac{\theta}{2} \end{bmatrix} \tag{5.89}$$

$$R_y(\theta) = cos\frac{\theta}{2}I - i\ sin\frac{\theta}{2}Y = \begin{bmatrix} cos\frac{\theta}{2} & -sin\frac{\theta}{2} \\ sin\frac{\theta}{2} & cos\frac{\theta}{2} \end{bmatrix} \tag{5.90}$$

$$R_z(\theta) = cos\frac{\theta}{2}I - i\ sin\frac{\theta}{2}Z = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix} \tag{5.91}$$

There are rotations around axis besides the x, y and z axis. Let $D = (D_x, D_y, D_z)$ be a 3-dimensional vector of size 1 from the origin. That determines an axis of the Bloch sphere that we can spin around. The rotation matrix is given as

$$R_D(\theta) = cos\frac{\theta}{2}I - i\ sin\frac{\theta}{2}(D_xX + D_yY + D_zZ). \tag{5.92}$$

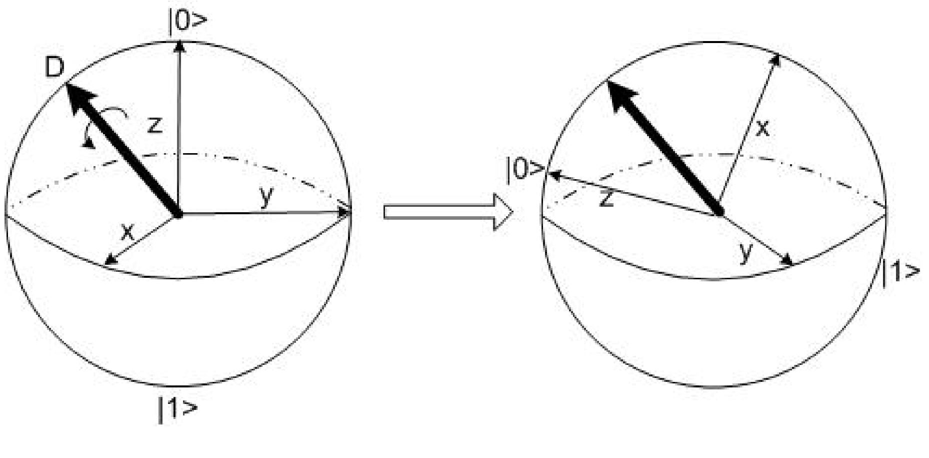
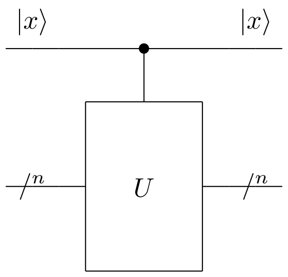


Figure 5.8: A Rotation of the Bloch Sphere at D.

As we have just seen, the Bloch sphere is a very valuable tool when it comes to understanding qubits and one-qubit operations. What about n-qubits? It turns out there is a high-dimension analogue of the sphere, but it is not easy to come to grip with. Indeed, it is a current research challenge to develop new ways of visualizing what happens when we manipulate several qubits at once. Entanglement, for instance, lies beyond the scope of the Bloch sphere (as it involves at least two qubits).

One of the central features of computer science is an operation that is done only under certain conditions and not under others. This is equivalent to an IF-THEN statement. If a certain (qu)bit is true, then a particular operation should be performed, otherwise the operation is not performed. For every n-qubit unitary operation U, we can create a unitary $n+1$-qubit operation **Controlled-**U or ${}^{C}U$.



(5.93)

This operation will perform the U operation if the top $|x\rangle$ input is a $|1\rangle$ and will simply perform the identity operation if $|x\rangle$ is $|0\rangle$.

For the simple case of

$$U = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \tag{5.94}$$

the controlled-U operation can be seen to be

$${}^{C}U = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & a & b \\ 0 & 0 & c & d \end{bmatrix}. \tag{5.95}$$

This same construction works for matrices larger than 2 by 2.

Exercise 5.4.9 *Show that the constructed ${}^{C}U$ works as it should when the top qubit is set to $|0\rangle$ or set to $|1\rangle$.*

Exercise 5.4.10 *Show that if U is unitary, then so is ${}^{C}U$.*

Exercise 5.4.11 *Show that the Toffoli gate is nothing more than ${}^{C}({}^{C}NOT)$*

Throughout the rest of this text we shall demonstrate the many operations that can be performed with quantum gates. However, there are limitations to what can be done with a quantum gate. For one thing, every operation must be reversible. Another limitation is a consequence of the **The No-Cloning Theorem.** This theorem says that it is impossible to clone an exact quantum state. In other words, it is impossible to make a copy of an arbitrary quantum state without first destroying the original. In "computerese," this says that we can "cut" and "paste" a quantum state but we cannot "copy" and "paste" a quantum state. "Move" is possible; "Copy" is imposable.

Why can't we? What would such a cloning operation look like? There would be two different places having the same vector space that describes a quantum system, say $\mathbb{V}$. Although these two systems would be apart, we are interested in looking at the two systems as one, i.e., we are interested in $\mathbb{V} \otimes \mathbb{V}$. A potential cloning operation would be a linear map

$$C : \mathbb{V} \otimes \mathbb{V} \longrightarrow \mathbb{V} \otimes \mathbb{V} \tag{5.96}$$

that should take an arbitrary state $|x\rangle$ in the first system and, perhaps, nothing in the second system and clone $|x\rangle$, i.e.,

$$C(|x\rangle \otimes 0) = (|x\rangle \otimes |x\rangle). \tag{5.97}$$

This seems like a harmless enough operation, and there is no problem cloning for an arbitrary *classical* state $|x\rangle$. In fact, every Xerox machine clones and every time we copy a file, classical states of information are cloned. However, this information is in a collapsed classical state. It is not in a superposition of states.

The problem is cloning a *superposition* of states, that is, an arbitrary quantum state. Suppose we have a superposition of states $\frac{|x\rangle + |y\rangle}{\sqrt{2}}$. Cloning such a state would mean that

$$C\left(\frac{|x\rangle + |y\rangle}{\sqrt{2}} \otimes 0\right) = \left(\frac{|x\rangle + |y\rangle}{\sqrt{2}} \otimes \frac{|x\rangle + |y\rangle}{\sqrt{2}}\right). \tag{5.98}$$

However if we insist that C is a quantum operation, then C must be linear and hence must respect the scalar multiplication and the addition in $\mathbb{V} \otimes \mathbb{V}$. If C was linear, then

$$C\left(\frac{|x\rangle + |y\rangle}{\sqrt{2}} \otimes 0\right) = C\left(\frac{1}{\sqrt{2}}(|x\rangle + |y\rangle) \otimes 0\right) = \frac{1}{\sqrt{2}}C((|x\rangle + |y\rangle) \otimes 0) \tag{5.99}$$

$$= \frac{1}{\sqrt{2}}(C(|x\rangle \otimes 0 + |y\rangle \otimes 0)) = \frac{1}{\sqrt{2}}(C(|x\rangle \otimes 0) + C(|y\rangle \otimes 0)) = \frac{1}{\sqrt{2}}((|x\rangle \otimes |x\rangle) + (|y\rangle \otimes |y\rangle)) \tag{5.100}$$

$$= \frac{(|x\rangle \otimes |x\rangle) + (|y\rangle \otimes |y\rangle)}{\sqrt{2}}. \tag{5.101}$$

But

$$\left(\frac{|x\rangle+|y\rangle}{\sqrt{2}}\otimes\frac{|x\rangle+|y\rangle}{\sqrt{2}}\right)\neq\frac{(|x\rangle\otimes|x\rangle)+(|y\rangle\otimes|y\rangle)}{\sqrt{2}}. \tag{5.102}$$

So C is not linear and hence no such map can exist.

In contrast to cloning, there is no problem **transporting** arbitrary quantum states from one system to another. Such a transporting operation would be a linear map

$$T:\mathbb{V}\otimes\mathbb{V}\longrightarrow\mathbb{V}\otimes\mathbb{V} \tag{5.103}$$

that should take an arbitrary state $|x\rangle$ in the first system and, say, nothing in the second system and transport $|x\rangle$ to the second system leaving nothing in the first system, i.e.,

$$T(|x\rangle\otimes 0)=(0\otimes|x\rangle). \tag{5.104}$$

We do not run into the same problem if we transport a superposition of states. In detail,

$$T\left(\frac{|x\rangle+|y\rangle}{\sqrt{2}}\otimes 0\right)=T\left(\frac{1}{\sqrt{2}}(|x\rangle+|y\rangle)\otimes 0\right)=\frac{1}{\sqrt{2}}T((|x\rangle+|y\rangle)\otimes 0) \tag{5.105}$$

$$=\frac{1}{\sqrt{2}}T((|x\rangle\otimes 0)+(|y\rangle\otimes 0))=\frac{1}{\sqrt{2}}(T(|x\rangle\otimes 0)+T(|y\rangle\otimes 0))=\frac{1}{\sqrt{2}}((0\otimes|x\rangle)+(0\otimes|y\rangle)) \tag{5.106}$$

$$=\frac{(|0\otimes(|x\rangle+|y\rangle)}{\sqrt{2}}=0\otimes\frac{(|x\rangle+|y\rangle)}{\sqrt{2}}. \tag{5.107}$$

This is exactly what we would expect from a transporting operation.[4]

Fans of Star Trek have long known that when Scotty "beams" Captain Kirk down from the Starship Enterprise to the planet Zygon, he is transporting Captain Kirk to Zygon. The Kirk of the Enterprise gets destroyed and only the Zygon Kirk survives. Captain Kirk is not being cloned. (Would we really want many copies of Captain Kirk all over the Universe?)

The reader might see an apparent contradiction in what we have stated. On the one hand, we have stated that the Toffoli and Fredkin gates can mimic the fanout gate. The matrices for the Toffoli and Fredkin gates are unitary, hence they are quantum gates. On the other hand, the no-cloning theorem says that no quantum gates can mimic the fanout operation. What is wrong here? Let us carefully examine the Fredkin gate. We have seen how this gate performs the cloning operation

$$(x,1,0)\qquad\mapsto\qquad(x,\neg x,x). \tag{5.108}$$

However what would happen if the x input was in a superposition of states, say, $\frac{|0\rangle+|1\rangle}{\sqrt{2}}$ while leaving $y=1$ and $z=0$. This would correspond to the state

$$\begin{array}{cccccccc} 000 & 001 & 010 & 011 & 100 & 101 & 110 & 111 \\ 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 \end{array}^{T}. \tag{5.109}$$

[4] In fact, we shall show how to transport arbitrary quantum states at the end of chapter 9.

Multiplying this state with the Fredkin gate gives us

$$\begin{array}{c} \\ 000 \\ 001 \\ 010 \\ 011 \\ 100 \\ 101 \\ 110 \\ 111 \end{array}\begin{array}{c} \begin{array}{cccccccc} 000 & 001 & 010 & 011 & 100 & 101 & 110 & 111 \end{array} \\ \left[\begin{array}{cccccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right] \end{array} \begin{bmatrix} 0 \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \\ 0 \end{bmatrix}. \quad (5.110)$$

The resulting state is

$$\frac{|0,1,0\rangle + |1,0,1\rangle}{\sqrt{2}}. \quad (5.111)$$

So, whereas on a classical bit x, the Fredkin gate performs the fanout operation as

$$(x,1,0) \qquad \mapsto \qquad (x,\neg x,x) \quad (5.112)$$

on a superposition of states, the Fredkin gate performs the following very strange operation

$$\left(\frac{|0\rangle + |1\rangle}{\sqrt{2}},1,0\right) \qquad \mapsto \qquad \frac{|0,1,0\rangle + |1,0,1\rangle}{\sqrt{2}}. \quad (5.113)$$

This strange operation is not a fanout operation. And so the no-cloning theorem safely stands.

Exercise 5.4.12 *Do a similar analysis for the Toffoli gate. Show that the way we set the Toffoli gate to perform the fanout operation does not clone a superposition of states.*

. . . .

Reader's Tip. The no-cloning theorem will be of major importance in chapters 9 and 10. □

. . . .

. . . .

References:

The basics of the qubits and quantum gates can be found in any textbook on quantum computing.

The history of reverable computation can be found in [1]. This [3] readable article by one of the forefathers of reversible computation is strongly recommended.

The no-cloning theorem was first proved in [2] and [4].

Bibliography

[1] Charles H. Bennett. Notes on the history of reversible computation. *IBM J. Res. Dev.*, 32(1):16–23, 1988.

[2] D. Dieks. Comunicating by epr devices. *Phys. Letters A*, 92(6):271–272, 1982.

[3] R. Landauer. Information is physical. *Physics Today*, 44:23–29, 1991.

[4] W. K. Wootters and W. H. Zurek. A single quantum cannot be cloned. *Nature*, 299(5886):802–803, October 1982.