QCOMP103: Quantum Architecture & Algorithms

Lotus Noir Quantum Computing Research Group

Based on:
Yanofsky, N. & Mannucci, M. (2008). *Quantum Computing for Computer Scientists.* Cambridge University Press

---

Lecture Outline: Now that we have laid out the mathematical and physical basic principles, we can move on to the core of theoretical quantum computing. In this lecture we will discover how to encode information in quantum systems with qubits and how to manipulate them with quantum gates. We will then continue our journey toward quantum algorithm design and principles of operation.

Contents

1 Quantum Architecture

1.1 Bits and Qubits

Notion goal 1.1. At the heart of classical computing is the notion of a bit, similarly at the heart of quantum computing is a generalization of the concept of bit called a qubit. In this section we will start by defining the notion of bit and build the notion of quantum bit on top.

Definition 1.1. *A **bit** is a unit of information describing a two-dimensional classical system.*

There are many examples of bits:

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

All those examples are expressing the same concept: **a bit is a way of describing a system whose set of states is of size 2**. We usually write these two possible states as 0 and 1. Let us represent those states as 2-dimensional column matrices such as:

$$|0\rangle = \begin{matrix}\mathbf{0}\\ \mathbf{1}\end{matrix}\begin{pmatrix}1\\0\end{pmatrix} \tag{1}$$

and similarly

$$|1\rangle = \begin{matrix}\mathbf{0}\\ \mathbf{1}\end{matrix}\begin{pmatrix}0\\1\end{pmatrix} \tag{2}$$

Notice how those two vector representations are **orthogonal** and thus mutually exclusive. Indeed a bit can be either in state $|0\rangle$ or in state $|1\rangle$, which was sufficient for classical computing. Either electricity is running through a circuit or it is not. Either a proposition is true or it is false. But **this mutual exclusivity of states is not sufficient in the quantum world**, indeed quantum systems can be in superposition of states, on and off simultaneously, true and false simultaneously. One quantum system can ve in state $|0\rangle$ and $|1\rangle$ simultaneously.

Definition 1.2. *A **quantum bit** or **qubit** is a unit of information describing a two-dimensional quantum system.*

We shall represent a qubit as a 2-by-1 matrix with complex coordinates:

$$|\psi\rangle = \begin{matrix}\mathbf{0}\\ \mathbf{1}\end{matrix}\begin{pmatrix}c_0\\c_1\end{pmatrix} \tag{3}$$

where $|c_0|^2+|c_1|^2 = 1$. Notice that **a classical bit is a special type of qubit**. The amplitude $|c_0|^2$ is to be interpreted as the probability of collapsing into state $|0\rangle$ upon measurement, whereas $|c_1|^2$ represent the probability of collapsing into state $|1\rangle$ upon measurement. It is easy to see that $|0\rangle$ and $|1\rangle$ are the **canonical basis** of $\mathbb{C}^2$, thus we can write any qubit $|\psi\rangle$ as:

$$|\psi\rangle = \begin{pmatrix}c_0\\c_1\end{pmatrix} = c_0\begin{pmatrix}1\\0\end{pmatrix} + c_1\begin{pmatrix}0\\1\end{pmatrix} = c_0\,|0\rangle + c_1\,|1\rangle \tag{4}$$

We can say that **a qubit collapses into a classical bit upon measurement**, which is itself a special case of qubit where $|c_0|^2 = 1$ or $|c_1|^2 = 1$ (note that we are working with normalized qubits where the length $|\psi| = 1$).

Following the normalization procedures that we have learned in the last lecture, any nonzero element of $\mathbb{C}^2$ can be converted into a qubit.

Example 1.1. The vector

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

has norm

$$\begin{aligned} |V| = \sqrt{\langle V, V \rangle} &= \sqrt{V^\dagger V} \\ &= \sqrt{(\overline{V})^T V} \\ &= \sqrt{\begin{pmatrix} 5-3i & -6i \end{pmatrix} \begin{pmatrix} 5+3i \\ 6i \end{pmatrix}} = \sqrt{34+36} = \sqrt{70} \end{aligned} \tag{6}$$

So V describes the same physical state as the qubit

$$\frac{V}{\sqrt{70}} = \begin{pmatrix} \frac{5+3i}{\sqrt{70}} \\ \frac{6i}{\sqrt{70}} \end{pmatrix} = \frac{5+3i}{\sqrt{70}} |0\rangle + \frac{6i}{\sqrt{70}} |1\rangle = \frac{1}{\sqrt{70}}((5+3i)|0\rangle + 6i|1\rangle) \tag{7}$$

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

Exercise 1.1. Normalize $W = \begin{pmatrix} 3+2i \\ 4-2i \end{pmatrix}$, then write it as a sum of $|0\rangle$ and $|1\rangle$ and finally expresses its probabilities of collapsing into either $|0\rangle$ or $|1\rangle$.

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 might be:

$$01101011 \tag{8}$$

Which in our matrix-based notation translates in:

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{9}$$

We learned in the previous lecture that in order to **combine systems** we need to use the **tensor product**, hence, we can describe this byte as:

$$|0\rangle \otimes |1\rangle \otimes |1\rangle \otimes |0\rangle \otimes |1\rangle \otimes |0\rangle \otimes |1\rangle \otimes |1\rangle \tag{10}$$

As a qubit (reminder: qubits are a generalization of classical bits), it is an element of the vector space:

$$(\mathbb{C}^2)^{\otimes 8} = \mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \mathbb{C}^2 \tag{11}$$

This is a complex vector space of dimension $2^8 = 256$. Since there is essentially only one complex vector space of this dimension, this vector space is isomorphic to $\mathbb{C}^{256}$. We can thus

describe our byte as a 256-by-1 column vector:

$$\begin{matrix} \mathbf{00000000} \\ \mathbf{00000001} \\ \vdots \\ \mathbf{01101010} \\ \mathbf{01101011} \\ \mathbf{01101100} \\ \vdots \\ \mathbf{11111110} \\ \mathbf{11111111} \end{matrix} \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix} \tag{12}$$

Exercise 1.2. 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 of $(\mathbb{C}^2)^{\otimes 3} = \mathbb{C}^8$. Do the same for 011 and 111.

This is fine for classical computing. However, for the quantum world, in order to permit superposition, **a generalization is needed**: every state of an eight-qubit can be written as:

$$\begin{matrix} \mathbf{00000000} \\ \mathbf{00000001} \\ \vdots \\ \mathbf{01101010} \\ \mathbf{01101011} \\ \mathbf{01101100} \\ \vdots \\ \mathbf{11111110} \\ \mathbf{11111111} \end{matrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{106} \\ c_{107} \\ c_{108} \\ \vdots \\ c_{254} \\ c_{255} \end{pmatrix} \tag{13}$$

where $\sum_{i=0}^{255} |c_i|^2 = 1$. Eight qubits together are called a **qubyte** and can collapse to $2^8 = 256$ different classical bits combinations upon measurement. Any assemblage of two or more qubit, similarly to its classical counterpart, is called **quantum register**, moreover if we take an example couple of assembled qubits $|0\rangle \otimes |1\rangle$, they can be written equivalently as $|0 \otimes 1\rangle$ and is denoted as a quantum register as $|01\rangle$.

Example 1.2. The qubit corresponding to $\frac{1}{\sqrt{3}} \begin{pmatrix} 1 \\ 0 \\ -1 \\ 1 \end{pmatrix}$ can be written as

$$\frac{1}{\sqrt{3}} |00\rangle - \frac{1}{\sqrt{3}} |10\rangle + \frac{1}{\sqrt{3}} |11\rangle = \frac{|00\rangle - |10\rangle + |11\rangle}{\sqrt{3}} \tag{14}$$

Exercise 1.3. Verify that statement.

In general, a state of two-qubit system can be written as

$$|\psi\rangle = c_{0,0} |00\rangle + c_{0,1} |01\rangle + c_{1,0} |10\rangle + c_{1,1} |11\rangle \tag{15}$$

Note that **the tensor product of two states is not commutative**:

$$|0 \otimes 1\rangle = |01\rangle \neq |10\rangle = |1 \otimes 0\rangle \tag{16}$$

Exercise 1.4. What vector corresponds to the state $4|01\rangle + 2|11\rangle$?

Let us revisit the notion of entanglement applied to qubits with a quick example, consider the state of a two-qubits system defined as:

$$\frac{|11\rangle + |00\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2}}|11\rangle + \frac{1}{\sqrt{2}}|00\rangle \tag{17}$$

Then those two qubits are entangled: if we measure the first qubit and it is found in state $|1\rangle$ then we automatically know that the second qubit is $|1\rangle$, and similarly the other way around.

What we have learned 1.1. In this part we have formalized the notion of classical bit and discovered the notion of quantum bit (or qubit), in particular we saw that:

- A **qubit** is a generalization of the classical bit and represent the amount of information stored in a two-dimensional quantum system (that can collapse on two possible basic states, each corresponding to a classical bit $|0\rangle$ or $|1\rangle$).
- The assemblage of two or more qubits is accomplished with the **tensor product**, such assemblages of multiple qubits are called **quantum registers**. A system of n qubits can store the equivalent of 2^n classical bits.
- Qubits can be **entangled** to one another.

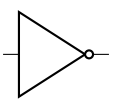
1.2 Classical Gates

Notion goal 1.2. In order to manipulate our qubits we need quantum gates. But before introducing such gates we must find a mathematical way of modelizing gates, we will then start by modelizing classical gates as matrix operators.

Classical logical gates are ways to **manipulating bits**. Bits enter and exit logical gates. As we saw before, we represent n input bits as a 2^n-by-1 matrix and m output bits as a 2^m-by-1 matrix. To represent our gates we thus need 2^m-by-2^n matrices, let us consider such a matrix G along with an input state $S_{input} \in \mathbb{C}^{2^n}$, we have then:

$$G * S_{input} = S_{output} \in \mathbb{C}^{2^m} \tag{18}$$

So **bits will be represented by column vector and logic gates by matrices**. Let us try with the simple example of the NOT gate.



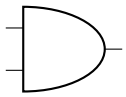
This gate take as input one bit (represented by a 2-by-1 matrix), and outputs one bit. NOT of $|0\rangle$ equals $|1\rangle$ and NOT of $|1\rangle$ equals $|0\rangle$. We can represent such an operation with the matrix:

$$NOT = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{19}$$

This matrix satisfies:

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \text{ and } \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{20}$$

What about the other gates? Consider the AND gate. It accepts two bits input and outputs one bit.



Because there are two inputs and one output, we will need a 2^1-by-2^2 matrix such as:

$$AND = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{21}$$

This matrix satisfies:

$$AND\,|11\rangle = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |1\rangle \tag{22}$$

And naturally:

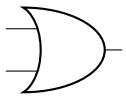
$$AND\,|01\rangle = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |0\rangle \tag{23}$$

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

$$\begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 3.5 \\ 2 \\ 0 \\ -4.1 \end{pmatrix} = \begin{pmatrix} 5.5 \\ -4.1 \end{pmatrix} \tag{24}$$

This is clearly nonsense. We are **allowed only to multiply these classical gates with vectors that represent classical states**, which are column matrices with a single 1 entry and all other entries 0. In classical computing, the bits are only in one state at a time and are described by such vectors. We will have more freedom with quantum gates.

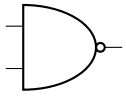
Let us continue to the OR gate.



Which can be represented by:

$$OR = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix} \tag{25}$$

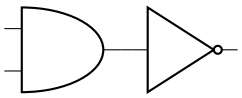
And finally the NAND gate, which is of special importance since **every logical gate can be composed of NAND gates**. It takes 2 bits for input and outputs 1 bit.



Let us try to determine 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 ($|00\rangle$, $|01\rangle$, $|10\rangle$, $|11\rangle$) does NAND output a $|1\rangle$ (answer: $|00\rangle$, $|01\rangle$, $|10\rangle$), and in which states does NAND output a $|0\rangle$ (answer: $|11\rangle$). From this we realize than NAND can be written as:

$$\begin{array}{c} \\ \mathbf{0} \\ \mathbf{1} \end{array} \begin{array}{c} \begin{array}{cccc} \mathbf{00} & \mathbf{01} & \mathbf{10} & \mathbf{11} \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \end{array} \tag{26}$$

There is another way to describe the NAND gate. Indeed a NAND gate is only an AND gate followed by a NOT gate.



In other words, we can perform the NAND operation by first performing the AND operation and then the NOT operation. This can be written as:

$$NOT * AND = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} * \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} = NAND \tag{27}$$

Example 1.3. Let two classical bits $|\alpha\rangle = \begin{pmatrix} w \\ x \end{pmatrix} = w \cdot |0\rangle + x \cdot |1\rangle$ and $|\beta\rangle = \begin{pmatrix} y \\ z \end{pmatrix} = y \cdot |0\rangle + z \cdot |1\rangle$

$$\begin{aligned} NAND * |\alpha\beta\rangle &= NAND * (|\alpha\rangle \otimes |\beta\rangle) \\ &= \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} * \begin{pmatrix} w \cdot \begin{pmatrix} y \\ z \end{pmatrix} \\ x \cdot \begin{pmatrix} y \\ z \end{pmatrix} \end{pmatrix} \\ &= \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} * \begin{pmatrix} wy \\ wz \\ xy \\ xz \end{pmatrix} \end{aligned} \tag{28}$$

So for our NAND gate to output $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, we need to solve the following system of equations:

$$\begin{aligned} &\begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} * \begin{pmatrix} wy \\ wz \\ xy \\ xz \end{pmatrix} = |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ \Leftrightarrow &\begin{cases} xz = 0 \\ wy + wz + xy = 1 \end{cases} \end{aligned} \tag{29}$$

Meaning that for the NAND gate to return $|1\rangle$, x and z cannot be equal to 1 at the same time, in other words, the gate will return true for $|00\rangle$, $|01\rangle$ and $|10\rangle$ but not for $|11\rangle$ since that would implies that $x = z = 1$.

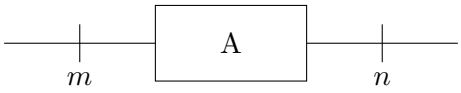
Exercise 1.5. Find a matrix that corresponds to the NOR gate.

Thinking of NAND as the application of the NOT operator followed by the application of the AND operator brings to light a general situation. When we perform a computation, **we often have to carry out one operation followed by another**.

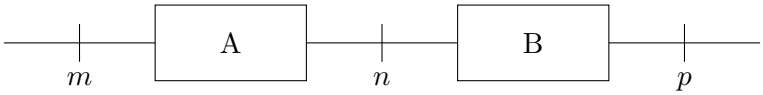
We call this procedure performing **sequential** operations. If matrix A corresponds to performing an operation and matrix B corresponds to another operation, then **the matrix product** $B * A$ (and **not** $A * B$) **corresponds to performing those two operations sequentially in the same order**.

For the rest of this course, we will assume the convention that computation flows **from left to right** and omit the heads of arrows. And so a computation of A followed by B shall be denoted:

Thus if we take in account the number of inputs and outputs, if A is an operation with m inputs and n outputs bits, then we shall draw this as:

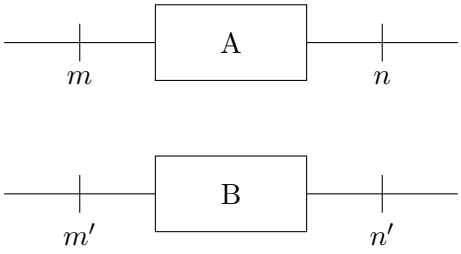


The matrix A would be of size 2^n-by2^m. Say B takes the n outputs of A as input and outputs p bits, thus we obtain:



then B is represented by a 2^p-by2^m matrix. Thus performing the operations A then B sequentially corresponds to the matrix $B * A$, which is a $(2^p\text{-by-}2^n) * (2^n\text{-by-}2^m) = (2^p\text{-by-}2^m)$ matrix.

Beside sequential operations, there are **parallel** operations as well:

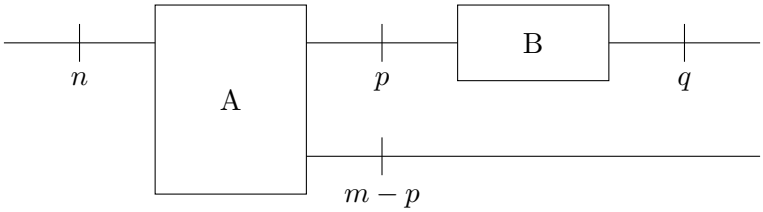


In this case we have A acting on some bits and B acting on others, this would be represented by **the tensor product** $A \otimes B$.

- A will be of size 2^n-by-2^m.
- B will be of size $2^{n'}$-by-$2^{m'}$
- $A \otimes B$ will be of size $2^{n+n'}$-by-$2^{m+m'}$ as we saw previously in the properties of the tensor product.

Combinations of sequential and parallel operations gates/matrices shall be called **circuits**, which can be represented by sometime really complicated matrices but which can all be decoupled into the sequential and parallel compositions of simple gates.

Example 1.4. 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.

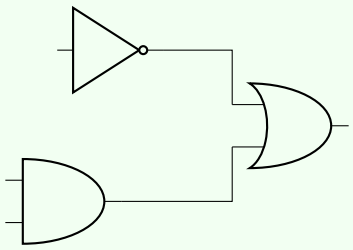


- A is a 2^m-by-2^n matrix.
- B is 2^q-by-2^p matrix.

As nothing should be done to the $m - p$ bits we might represent this as 2^{m-p}-by-2^{m-p} identity matrix I_{m-p}, we de not draw any gate for the identity matrix. Thus the entire circuit can be represented by the following matrix:

$$(B \otimes I_{m-p}) \ * \ A \tag{30}$$

Exercise 1.6. Consider the following circuit:



Find the matrix representing this circuit and calculate its numeric value.

What we have learned 1.2. In this section, we saw that we can manipulate bits of information with **gates**, and in particular:

- Gates **can be represented as matrices**, a gate taking m inputs and having n outputs would be represented by a 2^n-by-2^m matrix.
- **Classical gates** can only process **classical states** (or registers), where only one entry of the register is equal to 1 and all others entries equal 0.
- Gates can be applied in **parallel** or in **sequential** orders, an operation A followed by an operation B would be represented by the matrix $B * A$ and an operation A performing in parallel with an operation B would be represented by the matrix $A \otimes B$. By assembling those sequential and parallel operations we obtain a matrix representing an entire **circuit**.

1.3 Reversible Gates

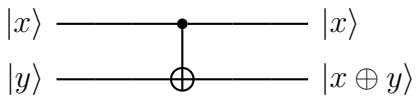
Notion goal 1.3. Not all of the logical gates we saw in the last section would work in quantum computers. As we saw in the last lecture, **all quantum operations that are not measurement are reversible and are represented by unitary matrices**.
For the quantum world we would need reversible gates, which are gate where we **can determine the input state from the output state**, which, for example, is not possible with the AND or the OR gates, in contrast the NOT gate and the identity gate are reversible.

Reversible gates have a history that predates quantum computing. Classical computers lose energy and generate heat. In the 1960s, Rolf Landauer analyzed computational processes and showed that erasing information, as opposed to writing information, is what causes energy loss and heat. This notion is known as the **Landauer's principle** and we can remember two things about it:

- **Writing** information is **reversible**.
- **Erasing** information is **irreversible**.

If erasing information is the only operation that uses energy, then a computer that is reversible and does not erase would not use any energy. What examples of reversible gates are there ? We already seen that the identity gate and the NOT gate are reversible. What else is there?

Consider the following **controlled-NOT gate (CNOT)**:



This gate has two inputs and two outputs. The top input is the **control bit**. It controls what the output will be.

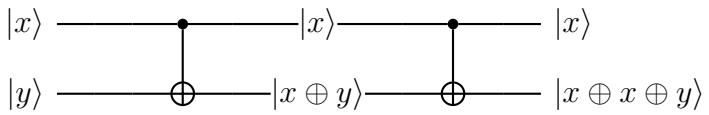
- If $|x\rangle = |0\rangle$ then the bottom output of $|y\rangle$ **will be the same as the input**.
- If $|x\rangle = |1\rangle$ then the bottom output of $|y\rangle$ **will be the opposite**.

If we write the top qubit first and then the bottom qubit, then the CNOT gate takes $|x,\ y\rangle$ to $|x,\ x \oplus y\rangle$, where $\oplus$ is the **bit-wise XOR operation**.

The matrix corresponding to this reversible gate is:

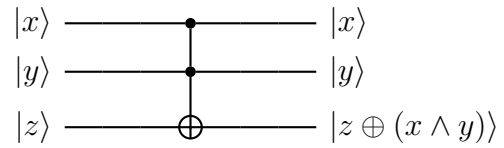
$$\begin{array}{c} \\ \mathbf{00} \\ \mathbf{01} \\ \mathbf{10} \\ \mathbf{11} \end{array} \begin{array}{c} \begin{array}{cccc} \mathbf{00} & \mathbf{01} & \mathbf{10} & \mathbf{11} \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \end{array} \tag{31}$$

The CNOT gate **can be reversed by itself**. Consider the following circuit:



State $|x, y\rangle$ goes to $|x,\ x \oplus y\rangle$ and then $|x,\ x \oplus (x \oplus y)\rangle$. This last state is equal to $|x,\ (x \oplus x) \oplus y\rangle$ because $\oplus$ is associative, and because $x \oplus x$ is always equal to 0 then the final state can be reduced to the original $|x, y\rangle$.

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



This is similar to the CNOT gate, but with two controlling bits. The bottom bit flips only when **both** of the top two bits are in state $|1\rangle$. We can write this operation as taking state $|x,\ y,\ z\rangle$ to $|x,\ y,\ z \oplus (x \wedge y)\rangle$ with $\wedge$ being the **bit-wise AND operator**.

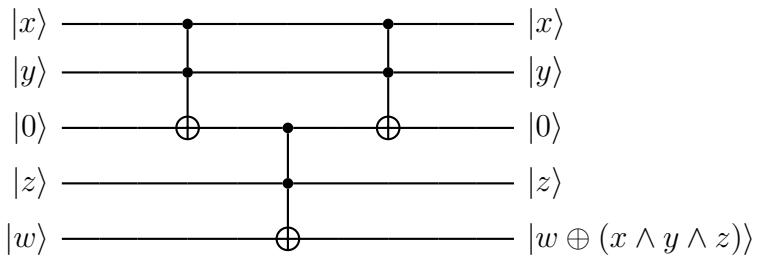
The matrix corresponding this gate is:

$$\begin{array}{c} \\ \mathbf{000} \\ \mathbf{001} \\ \mathbf{010} \\ \mathbf{011} \\ \mathbf{100} \\ \mathbf{101} \\ \mathbf{110} \\ \mathbf{111} \end{array} \begin{array}{c} \begin{array}{cccccccc} \mathbf{000} & \mathbf{001} & \mathbf{010} & \mathbf{011} & \mathbf{100} & \mathbf{101} & \mathbf{110} & \mathbf{111} \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{array} \tag{32}$$

Let us review what we have seen so far:

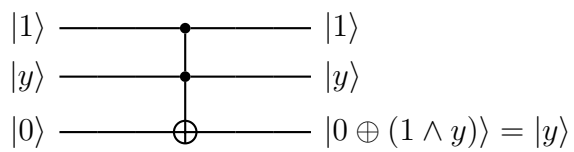
- The **NOT** gate **has no controlling bit**.
- The **CNOT** gate **has one controlling bits**.
- The **Toffoli** gate **has two controlling bits**.

Can we go on with this ? Yes. A gate with three controlling bits can be constructed from three Toffoli gates as follow:



One reason why the Toffoli gate is interesting is that it is **universal**, meaning that 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.

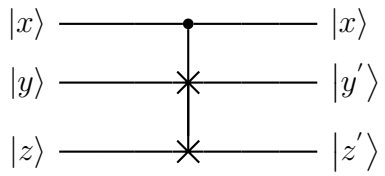
Example 1.5. In order to be able to construct all gates, we must also have a way of producing a fanout of values. In others words, a gate is needed that inputs a value and output two of the same values, this can be obtained by setting the value $|x\rangle$ to $|1\rangle$ and $|z\rangle$ to $|0\rangle$. We thus obtain the output $|1, y, y\rangle$



Exercise 1.7. Show that using one Toffoli gate, we can build an:

- AND gate
- NOT gate
- NAND gate

Another remarkable reversible gate is the **Fredkin gate** also called **Conditional SWAP** or **CSWAP**:



The top $|x\rangle$ is the **control bit**.

- If $|\mathbf{x}\rangle = |\mathbf{0}\rangle$ then $|\mathbf{y}'\rangle = |\mathbf{y}\rangle$ and $|\mathbf{z}'\rangle = |\mathbf{z}\rangle$.
- If $|\mathbf{x}\rangle = |\mathbf{1}\rangle$ then $|\mathbf{y}'\rangle = |\mathbf{z}\rangle$ and $|\mathbf{z}'\rangle = |\mathbf{y}\rangle$.

in other words, $|0, y, z\rangle \longrightarrow |0, y, z\rangle$ and $|1, y, z\rangle \longrightarrow |1, z, y\rangle$.

The matrix corresponding this gate is:

$$\begin{array}{c} \\ \mathbf{000} \\ \mathbf{001} \\ \mathbf{010} \\ \mathbf{011} \\ \mathbf{100} \\ \mathbf{101} \\ \mathbf{110} \\ \mathbf{111} \end{array} \begin{array}{c} \begin{array}{cccccccc} \mathbf{000} & \mathbf{001} & \mathbf{010} & \mathbf{011} & \mathbf{100} & \mathbf{101} & \mathbf{110} & \mathbf{111} \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{array} \tag{33}$$

Exercise 1.8. The Fredkin gate is also universal. Show that we can build an AND and NOT gate using the Fredkin gate.

So **both Toffoli and Fredkin gates are universal**. Not only there are both reversible gates; a glance at their matrices indicates that they are also **unitary**. In the next section, we will look at other unitary gates.

What we have learned 1.3. In this section we have discussed the notion of reversible gates, in particular we saw that:

- **Reversible gates** are logical gates from which **we can determine their inputs by looking at their outputs**. For example, the AND and NOT gates are reversible whereas the OR gate is not.
- **Universal gates** are logical gates **from which we can build any other logical gates**. For example, Toffoli and Fredkin gates are universal.
- **Toffoli**, **Fredkin (CSWAP)** and **CNOT** gates are their own inverses.

1.4 Quantum Gates

Notion goal 1.4. We have introduced the concepts of **reversible** gates and **unitary** gates, as we saw in the previous lecture **quantum states can only be modified by unitary operations**, we shall now introduce **quantum gates** as a way to manipulate qubits.

Definition 1.3. *A **quantum gate** is simply an operator that acts on qubits. Such operators will be represented by unitary matrices.*

We have already worked with some quantum gates such as the **identity operator** denoted **I**, the **NOT**, **CNOT**, **Toffoli** and **Fredkin** gates as well as the **Hadamard** operator denoted **H** which we saw at the very beginning of this course:

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix} = \begin{pmatrix}\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}}\\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}}\end{pmatrix} \tag{34}$$

Which is a **transition matrix** in $\mathbb{R}^2$ allowing to transition between the canonical basis:

$$\left\{\begin{pmatrix}1\\0\end{pmatrix} \quad \begin{pmatrix}0\\1\end{pmatrix}\right\} \tag{35}$$

and the other orthonormal basis:

$$\left\{\begin{pmatrix}\frac{1}{\sqrt{2}}\\ \frac{1}{\sqrt{2}}\end{pmatrix} \quad \begin{pmatrix}\frac{1}{\sqrt{2}}\\ -\frac{1}{\sqrt{2}}\end{pmatrix}\right\} \tag{36}$$

Let us look at some other quantum gates. The following three matrices are called the **Pauli matrices** and are very important:

$$X = \begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix},\ Y = \begin{pmatrix}0 & -i\\ i & 0\end{pmatrix},\ Z = \begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix} \tag{37}$$

They occur everywhere in quantum mechanics and quantum computing. Note that the X matrix is nothing more than a NOT gate. Other important matrices that will be used are:

$$S = \begin{pmatrix}1 & 0\\ 0 & i\end{pmatrix} \text{ and } T = \begin{pmatrix}1 & 0\\ 0 & e^{\frac{i\pi}{4}}\end{pmatrix} = \begin{pmatrix}1 & 0\\ 0 & \frac{1+i}{\sqrt{2}}\end{pmatrix} \tag{38}$$

Exercise 1.9. Show that X, Y, Z, S and T are unitary and show the action of each of them on an arbitrary qubit $|\psi\rangle = \begin{pmatrix}c_0\\ c_1\end{pmatrix}$.

All those gates are intimately related to each others:

- $X^2 = Y^2 = Z^2 = I_2$
- $H = \frac{1}{\sqrt{2}}(X + Z)$
- $X = HZH$
- $Z = HXH$
- $-1 \cdot Y = HYH$
- $S = T^2$
- $-1 \cdot Y = XYX$

But wait there is more! Let us consider another one-qubit quantum gate with an interesting name: the **square root of NOT** which is denoted $\sqrt{\textbf{NOT}}$. Its matrix representation is:

$$\sqrt{NOT} = \frac{1}{\sqrt{2}} \cdot \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix} \tag{39}$$

The first thing to notice is that **this gate is not its own inverse**, meaning that:

$$\sqrt{NOT} \neq (\sqrt{NOT})^{\dagger} \tag{40}$$

In order to understand why this gate has such a strange name, let us multiply $\sqrt{NOT}$ by itself:

$$\sqrt{NOT} \ * \ \sqrt{NOT} = (\sqrt{NOT})^2 = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \tag{41}$$

Which is very similar to the NOT gate. Let us put the qubits $|0\rangle$ and $|1\rangle$ through the $\sqrt{NOT}$ gate twice:

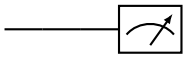
$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \xrightarrow{\sqrt{NOT}} \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} \xrightarrow{\sqrt{NOT}} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |1\rangle \tag{42}$$

and

$$|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \xrightarrow{\sqrt{NOT}} \begin{pmatrix} \frac{-1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} \xrightarrow{\sqrt{NOT}} \begin{pmatrix} -1 \\ 0 \end{pmatrix} = -1 \cdot |0\rangle \tag{43}$$

Remember that since **scalar multiples of a quantum state are equivalent**, $|0\rangle$ and $-1 \cdot |0\rangle$ represent the same same state thus we are confident that **the square of $\sqrt{\textbf{NOT}}$ performs the same operation as the NOT gate**, and hence the name.

There is one other gate we have not yet discussed: the **measurement operation**. This operation is **not unitary nor (in general) even reversible**. This operations is usually performed at the end of a computation when we want to measure qubits (and find bits). We shall denote it as:



There is a beautiful **geometric way of representing one-qubit states and operations**. Remember from the beginning of our course 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**.

$$|c|^2 = c \times \overline{c} = (x + yi) \times (x - yi) = x^2 + y^2 = 1 \tag{44}$$

Which is the formula of a circle of radius 1. In other words, **every complex number of radius/modulus 1 can be identified by the angle ϕ that the vector makes with the positive x axis**. There is an analogous representation of a qubit as an arrow in a three-dimensional sphere. Let us see how it works.

A generic qubit is of the form:

$$|\psi\rangle = c_0 \cdot |0\rangle + c_1 \cdot |1\rangle \ \text{ where } \ |c_0|^2 + |c_1|^2 = 1 \tag{45}$$

Although there is **four real numbers** involved in this qubit, it turns out that there are only **two actual degrees of freedom** to the three-dimensional ball. Let us rewrite this qubit in the polar form:

$$c_0 = r_0 e^{i\phi_0} \text{ and } c_1 = r_1 e^{i\phi_1} \tag{46}$$

giving us the qubit with r_i representing the **modulus** and ϕ_i representing the **phase** of c_i:

$$|\psi\rangle = r_0 e^{i\phi_0} \cdot |0\rangle + r_1 e^{i\phi_1} |1\rangle \tag{47}$$

There are still **four real parameters**: $r_0,\ r_1,\ \phi_0$ and ϕ_1.

However, **a quantum physical state does not change if we multiply its corresponding vector by an arbitrary complex number of norm 1**. We can therefore obtain an equivalent expression of our qubit $|\psi\rangle$ where the amplitude for $|0\rangle$ is **real**, by "killing" its phase:

$$\begin{aligned} e^{-i\phi_0}|\psi\rangle &= e^{-i\phi_0}(r_0 e^{i\phi_0}\cdot|0\rangle + r_1 e^{i\phi_1}\cdot|1\rangle) \\ &= r_0 e^{i(\phi_0-\phi_0)}\cdot|0\rangle + r_1 e^{i(\phi_1-\phi_0)}\cdot|1\rangle \\ &= r_0\cdot 1\cdot|0\rangle + r_1 e^{i(\phi_1-\phi_0)}\cdot|1\rangle \\ \\ &= r_0\cdot|0\rangle + r_1 e^{i(\phi_1-\phi_0)}\cdot|1\rangle \end{aligned} \tag{48}$$

We now have **only three real parameters**: $r_0,\ r_1$ and $\phi = \phi_1 - \phi_0$.
But we can do better using the fact that:

$$\begin{aligned} 1 &= |c_0|^2 + |c_1|^2 \\ &= \left|r_0 e^{i\phi_0}\right|^2 + \left|r_1 e^{i\phi_1}\right|^2 \\ &= |r_0|^2\cdot\left|e^{i\phi_0}\right|^2 + |r_1|^2\cdot\left|e^{i\phi_1}\right|^2 \end{aligned} \tag{49}$$

we get that

$$r_0^2 + r_1^2 = 1 \tag{50}$$

Remembering the trigonometric identity stating that $cos(\theta)^2 + sin(\theta)^2 = 1$, we can rename them as:

$$r_0 = cos(\theta) \text{ and } r_1 = sin(\theta) \tag{51}$$

Summing up, we obtain the following representation of $|\psi\rangle$:

$$|\psi\rangle = cos(\theta)\cdot|0\rangle + sin(\theta)e^{i\phi}\cdot|1\rangle \tag{52}$$

with **only two real parameters remaining**. Where $0 \leq \phi < 2\pi$ and $0 \leq \theta \leq \frac{\pi}{2}$ are enough to cover all possible qubits.

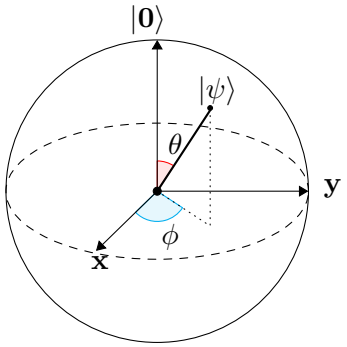


Figure 1: The Bloch sphere

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

Every qubit **can be represented by two angles** that describe such an arrow, by analogy with the case of geography, θ would correspond to the **latitude** and ϕ would correspond to **longitude**. The standard parametrization of the unit sphere would be:

$$\begin{aligned} x &= cos(\phi)sin(\theta) \\ y &= sin(\phi)sin(\theta) \\ z &= cos(\theta) \\ &\text{with } 0 \leq \phi < 2\pi \text{ and } 0 \leq \theta \leq \pi \end{aligned} \tag{53}$$

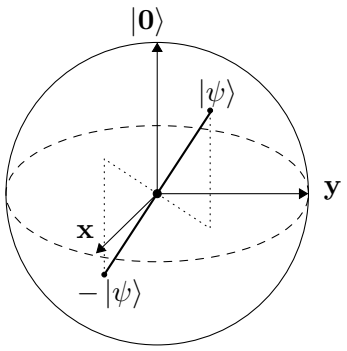


Figure 2: $|\psi\rangle \longrightarrow (\theta,\ \phi)$ and $-|\psi\rangle \longrightarrow (\pi - \theta,\ \phi + \pi)$ represent the same state with the sphere's standard parametrization

However **there is a problem with the standard parametrization** as we can see in Figure 2, the points $(\theta,\ \phi)$ and $(\pi - \theta,\ \phi + \pi)$ represent the same qubit up to factor -1, this **would map the same qubit twice, on the upper hemisphere and on the lower one, thus wasting half of our qubit "address range"**.

To mitigate this problem, we simply **double our "latitude"** (θ) to cover the entire sphere at "half speed", thus **doubling our effective "address range" to compensate**, we thus obtain the following parametrization of the sphere:

$$\begin{aligned} x &= cos(\phi)sin(2\theta) \\ y &= sin(\phi)sin(2\theta) \\ z &= cos(2\theta) \\ &\text{where } 0 \leq \phi < 2\pi \text{ and } \theta \leq \frac{\pi}{2} \end{aligned} \tag{54}$$

It's important to notice that with the **standard parametrization** described in Equation 53, $|1\rangle$ would be represented by an arrow on the plane formed by the axis x and y:

$$|1\rangle = 0 \cdot |0\rangle + 1 \cdot |1\rangle = cos(\frac{\pi}{2} + 2k\pi) \cdot |0\rangle + 1 \cdot |1\rangle \text{ with } k \in \mathbb{Z} \tag{55}$$

Giving us in the standard parametrization:

$$\begin{aligned} x &= cos(\phi)sin(\theta) = cos(\phi)sin(\frac{\pi}{2} + 2k\pi) = cos(\phi) \cdot 1 = cos(\phi) \\ y &= sin(\phi)sin(\theta) = sin(\phi)sin(\frac{\pi}{2} + 2k\pi) = sin(\phi) \cdot 1 = sin(\phi) \\ z &= cos(\theta) = cos(\frac{\pi}{2} + 2k\pi) = 0 \end{aligned} \tag{56}$$

This representation would only use one half (or one hemisphere) of the Bloch sphere, **which does not make sense given the prototypical implementation of the qubit based on**

spin that we saw in the last lecture, a qubit can indeed be represented by a particle which can be either spinning up $|\uparrow\rangle$ or spinning down $|\downarrow\rangle$. Remember that **the Bloch sphere is a real physical object**, not a mere mathematical construction.

Let us try to place $|1\rangle$ on the sphere using the parametrization described in Equation 54:

$$\begin{aligned} x &= cos(\phi)sin(2\theta) = cos(\phi)sin(2 \cdot \frac{\pi}{2} + 2k\pi) = cos(\phi)sin(\pi + 2k\pi) = cos(\phi) \cdot 0 = 0 \\ y &= sin(\phi)sin(2\theta) = sin(\phi)sin(2 \cdot \frac{\pi}{2} + 2k\pi) = sin(\phi)sin(\pi + 2k\pi) = sin(\phi) \cdot 0 = 0 \\ z &= cos(2\theta) = cos(\pi + 2k\pi) = -1 \end{aligned} \tag{57}$$

Which give us indeed a **downward vertical arrow** of length 1 in the Bloch sphere, which makes a lot more sense given the physical implementation of our qubit. From now on, **we will only use the parametrization described in Equation 54**.

Let us summarize the Bloch sphere and its geometry:

- The **"north pole"** corresponds to the state $|0\rangle$.
- The **"south pole"** corresponds to the state $|1\rangle$.
- The angle ϕ is the **"longitude"**, corresponding to the angle that $|\psi\rangle$ makes from the x axis along the equator.
- The angle θ is the **half** of the **"latitude"**, corresponding to the **half** of the angle that $|\psi\rangle$ makes from the z axis.

When a qubit is **measured in the standard basis** $\{|0\rangle, |1\rangle\}$, it collapses to a classical bit ($|0\rangle$ or $|1\rangle$), or equivalently, to the "north pole" or the "south pole". The **probability** of which "pole" the qubit will collapse **depends exclusively on how high or low the qubit's arrow 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 it collapsing to either $|0\rangle$ or $|1\rangle$.

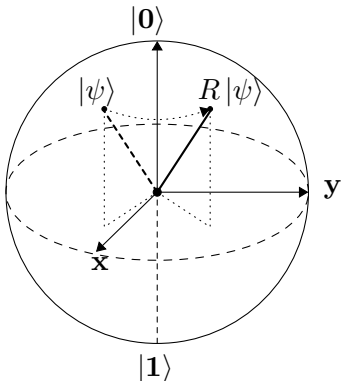


Figure 3: Performing a phase change on $|\psi\rangle$

If we take an arbitrary qubit and **rotate it around the z axis** thus modifying the angle ϕ (its **"longitude"**), notice that the probability of which classical state it will collapse **is not affected**. Such a change is called a **phase change**. In the representation given in Equation 52, this corresponds to altering the parameter $e^{i\phi}$.

Note that just as $|0\rangle$ and $|1\rangle$ sit on opposite sides of the sphere, as we can see on Figure 4, **an arbitrary pair of orthogonal qubits is mapped to antipodal points of the Bloch sphere**.

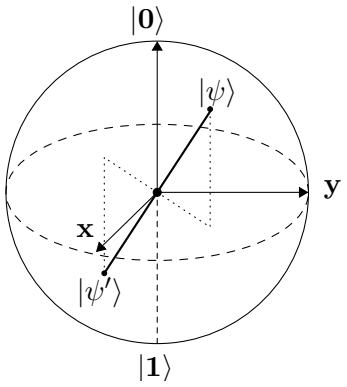


Figure 4: $|\psi\rangle \longrightarrow (\theta,\ \phi)$ and $|\psi'\rangle \longrightarrow (\pi - 2\theta,\ \phi + \pi)$ **does not represent the same state** with the parametrization described in the Equation 54, in fact the states described by those two qubits are **orthogonal**.

The Bloch sphere is interesting in that **every unitary 2-by-2 matrix (a one qubit operation) can be visualized as a way of manipulating the sphere**. We have seen in the previous lecture that every unitary matrix is an isometry, meaning that such a matrix **maps qubits to qubits**, geometrically this corresponds to **rotation** or an **inversion** of the Bloch sphere.

The X, Y, and Z **Pauli matrices** are ways of "flipping" the Bloch sphere 180° around the x, y and z axes respectively. Remember that **X** is nothing more than 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 and vice versa**. The other Pauli matrices work similarly, the Figure 5 shows the action of the Y operation.

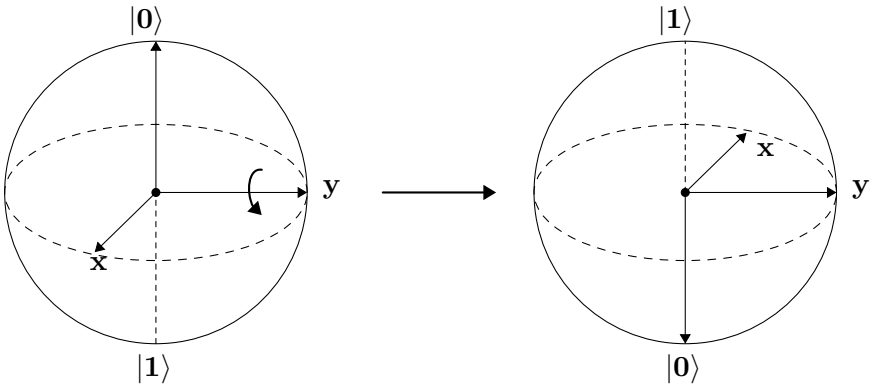


Figure 5: A 180° rotation of the Bloch sphere around y

There are time where we are not interested in performing a total 180° flip but just want to **turn the Bloch sphere α degrees along a particular direction**. To this end we can use a gate called **phase shift** gate, it is defined as:

$$R(\alpha) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\alpha} \end{pmatrix} \tag{58}$$

This gate performs the following operation on any arbitrary qubit:

$$cos(\theta)\,|0\rangle + e^{i\phi}sin(\theta)\,|1\rangle = \begin{pmatrix} cos(\theta) \\ e^{i\phi}sin(\theta) \end{pmatrix} \xrightarrow{R(\alpha)} \begin{pmatrix} cos(\theta) \\ e^{i\alpha}e^{i\phi}sin(\theta) \end{pmatrix} = \begin{pmatrix} cos(\theta) \\ e^{i(\alpha+\phi)}sin(\theta) \end{pmatrix} \tag{59}$$

This correspond to a **rotation that leaves the latitude alone and just change the longitude**. The new state of the qubit will **remain unchanged**, only the phase will change.

There are also times when we want to rotate a particular number of degrees around the x, y and z axis. These three matrices will perform the task:

$$\begin{aligned} R_x(\alpha) &= cos(\alpha)I - i\ sin(\alpha)X \\ &= \begin{pmatrix} cos(\alpha) & 0 \\ 0 & cos(\alpha) \end{pmatrix} - \begin{pmatrix} 0 & i\ sin(\alpha) \\ i\ sin(\alpha) & 0 \end{pmatrix} \\ &= \begin{pmatrix} cos(\alpha) & -i\ sin(\alpha) \\ -i\ sin(\alpha) & cos(\alpha) \end{pmatrix} \end{aligned} \tag{60}$$

$$\begin{aligned} R_y(\alpha) &= cos(\alpha)I - i\ sin(\alpha)Y \\ &= \begin{pmatrix} cos(\alpha) & 0 \\ 0 & cos(\alpha) \end{pmatrix} - \begin{pmatrix} 0 & -i \cdot (i\ sin(\alpha)) \\ i \cdot (i\ sin(\alpha)) & 0 \end{pmatrix} \\ &= \begin{pmatrix} cos(\alpha) & -sin(\alpha) \\ sin(\alpha) & cos(\alpha) \end{pmatrix} \end{aligned} \tag{61}$$

$$\begin{aligned} R_z(\alpha) &= cos(\alpha)I - i\ sin(\alpha)Z \\ &= \begin{pmatrix} cos(\alpha) & 0 \\ 0 & cos(\alpha) \end{pmatrix} - \begin{pmatrix} 1 \cdot (i\ sin(\alpha)) & 0 \\ 0 & -1 \cdot (i\ sin(\alpha)) \end{pmatrix} \\ &= \begin{pmatrix} cos(\alpha) - i\ sin(\alpha) & 0 \\ 0 & cos(\alpha) + i\ sin(\alpha) \end{pmatrix} \\ &= \begin{pmatrix} e^{-i\alpha} & 0 \\ 0 & e^{i\alpha} \end{pmatrix} \end{aligned} \tag{62}$$

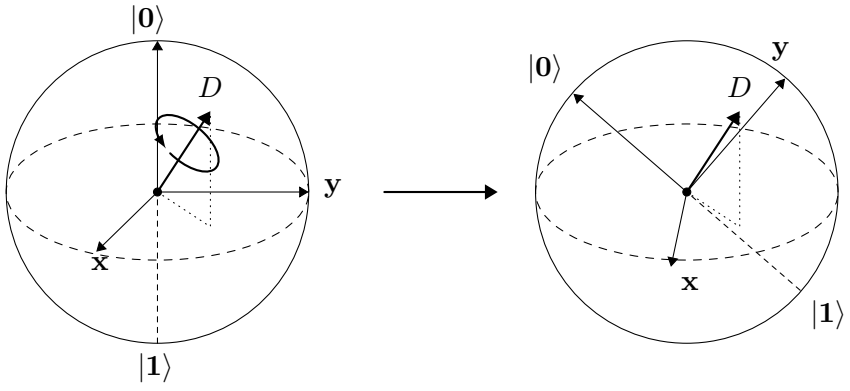


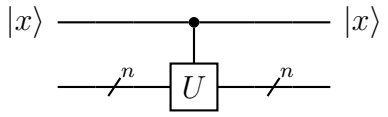
Figure 6: A rotation of the Bloch sphere around D

More generally, there are rotations around axes beside x, y and z. Let $D = (D_x, D_y, D_z)$ be a 3D vector of size 1 from the origin. This determines an axis on the Bloch sphere around which we can spin (see Figure 6). The rotation matrix is given by:

$$R_D(\alpha) = cos(\alpha)I - i\ sin(\alpha)(D_xX + D_yY + D_zZ) \tag{63}$$

As we have just seen, **the Bloch sphere is a very valuable tool when it comes to understanding qubits and one-qubit operations**. There is currently no very intuitive way to visualize more than one qubit manipulation, indeed **it still is a current research challenge to develop new ways to visualize 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.

There are still other quantum gates, for every n-qubit unitary operation U, we can create a unitary $(n+1)$-qubit operation referred as **controlled-U** or ${}^{C}U$.



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 being a one-qubit operation such as:

$$U = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \tag{64}$$

Then the associated controlled-U can be seen as:

$${}^{C}U = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & a & b \\ 0 & 0 & c & d \end{pmatrix} \tag{65}$$

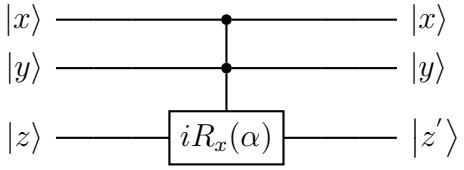
The same construction works for matrices larger than 2-by-2.

Exercise 1.10. Show that the constructed ${}^{C}U$ works as it should when the top qubit $|x\rangle$ is set to $|0\rangle$ or set to $|1\rangle$.

We know that every classical logical circuit can be simulated using only AND and NOT, we thus say the $\{AND,\ NOT\}$ form a set of **universal logical gates**. But are there sets of quantum gates that can simulate al quantum gates ? In other words, are there universal quantum gates ? The answer is yes. One set of universal quantum gates is:

$$\{H,\ {}^{C}NOT,\ R(cos^{-1}(\frac{3}{5})) \tag{66}$$

that is, the **Hadamard** gate, the **controlled-NOT** gate and **this specific phase shift gate**. There is also a quantum gate called the **Deutsch gate**, denoted $D(\alpha)$, depicted as:



Which is simply a **rotation of α around the x axis of the Bloch sphere with two control qubits** and look very similar to the Toffoli gate. When α is not a rational multiple of π, $D(\alpha)$ is by itself a **universal quantum gate**, in other words, $D(\alpha)$ will be able to mimic every other quantum gate.

Exercise 1.11. Show that the Toffoli gate is nothing more than $D(\frac{\pi}{2})$.

In this lecture, we demonstrate many of the operations that can be performed with quantum gates, however **there are limitations** to what can be done with them. We already know that **every operation must be reversible**, but another limitation is a consequence of 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**. We can then "cut" and "past" quantum states but we cannot "copy" and "paste" them. **Move is possible. Copy is impossible**.

What is the difficulty ? How would such a cloning operation look ? Let $\mathbb{V}$ represent a quantum system. As we intend to clone states in this system we thus deal with two systems and then let us "double" this vector space and deal with $\mathbb{V} \otimes \mathbb{V}$. A potential cloning operation C would be a linear map (therefore unitary):

$$C : \mathbb{V} \otimes \mathbb{V} \longrightarrow \mathbb{V} \otimes \mathbb{V} \tag{67}$$

That should take an arbitrary state $|x\rangle$ in the first system and, perhaps, nothing in the second system and clone $|x\rangle$:

$$C(|x\rangle \otimes 0) = (|x\rangle \otimes |x\rangle) \tag{68}$$

This seem like a harmless enough operation, but it is ? Let show how it behaves on the basic states:

$$C(|0\rangle \otimes |0\rangle) = |0\rangle \otimes |0\rangle \text{ and } C(|1\rangle \otimes |0\rangle) = |1\rangle \tag{69}$$

Because C **must be linear**, meaning that:

$$C(|\phi\rangle + |\psi\rangle) = C(|\phi\rangle) + C(|\psi\rangle) \text{ and } C(c \cdot |\phi\rangle) = c \cdot C(|\phi\rangle) \tag{70}$$

We should have that:

$$C((c_0 |0\rangle + c_1 |1\rangle) \otimes |0\rangle) = c_0 |0\rangle \otimes |0\rangle + c_1 |1\rangle \otimes |1\rangle \tag{71}$$

for an arbitrary quantum state, i.e, an arbitrary **superposition** of $|0\rangle$ and $|1\rangle$. Suppose we start with $\frac{|x\rangle+|y\rangle}{\sqrt{2}}$. Cloning such a state would mean that

$$C\left(\frac{|x\rangle + |y\rangle}{\sqrt{2}} \otimes 0\right) = \left(\frac{|x\rangle + |y\rangle}{\sqrt{2}} \otimes \frac{|x\rangle + |y\rangle}{\sqrt{2}}\right) \tag{72}$$

However, if we insist that C **is a quantum operation**, then C **must be linear**. If C was linear, then

$$\begin{aligned} C\left(\frac{|x\rangle + |y\rangle}{\sqrt{2}} \otimes 0\right) &= C(\frac{1}{\sqrt{2}}(|x\rangle + |y\rangle) \otimes 0) \\ &= \frac{1}{\sqrt{2}} \cdot C((|x\rangle + |y\rangle) \otimes 0) \\ &= \frac{1}{\sqrt{2}} \cdot (C(|x\rangle \otimes 0 + |y\rangle \otimes 0)) \\ &= \frac{1}{\sqrt{2}} \cdot (C(|x\rangle \otimes 0) + C(|y\rangle \otimes 0)) \\ &= \frac{1}{\sqrt{2}} \cdot ((|x\rangle \otimes |x\rangle) + (|y\rangle \otimes |y\rangle)) \\ &= \frac{(|x\rangle \otimes |x\rangle) + (|y\rangle \otimes |y\rangle)}{\sqrt{2}} \end{aligned} \tag{73}$$

But

$$\left(\frac{|x\rangle + |y\rangle}{\sqrt{2}} \otimes \frac{|x\rangle + |y\rangle}{\sqrt{2}}\right) \neq \frac{(|x\rangle \otimes |x\rangle) + (|y\rangle \otimes |y\rangle)}{\sqrt{2}} \tag{74}$$

Therefore, C **is not a linear map**, and hence **is not permitted**.

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

$$T : \mathbb{V} \otimes \mathbb{V} \longrightarrow \mathbb{V} \otimes \mathbb{V} \tag{75}$$

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

$$T(|x\rangle \otimes 0) = (0 \otimes |x\rangle) \tag{76}$$

We do not run into the same problem as earlier if we transport a superposition of states:

$$\begin{aligned} T\left(\frac{|x\rangle + |y\rangle}{\sqrt{2}} \otimes 0\right) &= T(\frac{1}{\sqrt{2}}(|x\rangle + |y\rangle) \otimes 0) \\ &= \frac{1}{\sqrt{2}} \cdot T((|x\rangle + |y\rangle) \otimes 0) \\ &= \frac{1}{\sqrt{2}} \cdot T((|x\rangle \otimes 0) + (|y\rangle \otimes 0)) \\ &= \frac{1}{\sqrt{2}} \cdot (T(|x\rangle \otimes 0) + T(|y\rangle \otimes 0)) \\ &= \frac{1}{\sqrt{2}} \cdot ((0 \otimes |x\rangle) + (0 \otimes |y\rangle)) \\ &= \frac{(0 \otimes (|x\rangle + |y\rangle))}{\sqrt{2}} \\ &= 0 \otimes \frac{(|x\rangle + |y\rangle)}{\sqrt{2}} \end{aligned} \tag{77}$$

Which is exactly what we would expect from a transporting operation.

You might see **an apparent contradiction in what we have stated**. On one and we have stated that the Toffoli (CCNOT) and Fredkin (CSWAP) gates **can mimic the fanout gate, thus duplicating states**. And on the other hand, **the no-cloning theorem say that no quantum gate 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:

$$(x,\ 1,\ 0) \longrightarrow (x,\ \neg x,\ x) \tag{78}$$

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

$$|x\rangle \otimes |y\rangle \otimes |z\rangle = |xyz\rangle = |x,\ 1,\ 0\rangle = \begin{matrix} \mathbf{000} \\ \mathbf{001} \\ \mathbf{010} \\ \mathbf{011} \\ \mathbf{100} \\ \mathbf{101} \\ \mathbf{110} \\ \mathbf{111} \end{matrix} \begin{pmatrix} 0 \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \end{pmatrix} \tag{79}$$

Multiplying this state with the Fredkin gate gives us

$$\begin{array}{c} \\ \mathbf{000} \\ \mathbf{001} \\ \mathbf{010} \\ \mathbf{011} \\ \mathbf{100} \\ \mathbf{101} \\ \mathbf{110} \\ \mathbf{111} \end{array} \begin{array}{c} \begin{array}{cccccccc} \mathbf{000} & \mathbf{001} & \mathbf{010} & \mathbf{011} & \mathbf{100} & \mathbf{101} & \mathbf{110} & \mathbf{111} \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{array} * \begin{pmatrix} 0 \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \\ 0 \end{pmatrix} \tag{80}$$

The resulting state is

$$\frac{|0,\ 1,\ 0\rangle + |1,\ 0,\ 1\rangle}{\sqrt{2}} \tag{81}$$

So whereas on a **classical** bit $|x\rangle$, the Fredkin gate performs the fanout operation, on a superposition of states the Fredkin gate performs the following very strange operation:

$$\left(\frac{|0\rangle + |1\rangle}{\sqrt{2}},\ 1,\ 0\right) \longrightarrow \frac{|0,\ 1,\ 0\rangle + |1,\ 0,\ 1\rangle}{\sqrt{2}} \tag{82}$$

This strange operation **is not a fanout operation**. Thus the no-cloning theorem safely stands.

Exercise 1.12. Do a similar analysis on for the Toffoli gate. Show that the the way we set the Toffoli gate to perform a fanout operation does not clone a superposition of states.

What we have learned 1.4. In this section we have formalized the concept of quantum gates, introduced a new way to visualize single-qubit operations and showed the limitations of quantum gates, in particular we will retain that:

- **Quantum gates** are unitary operators acting on one or more qubits.
- The **polar representation of a qubit** $|\psi\rangle$ is written $|\psi\rangle = cos\theta\,|0\rangle + e^{i\phi}sin\theta\,|1\rangle$ and can then be described by two real numbers $(\theta,\ \phi)$ with $0 \leq \theta \leq \frac{\pi}{2}$ and $0 \leq \phi < 2\pi$.
- Any qubit $(\theta,\ \phi)$ can be represented on the **Bloch sphere** at the polar coordinates $(2\theta,\ \phi)$, therefore **orthogonal qubits are represented at antipodal positions on the sphere**.
- Single-qubit operations can be represented as **rotations of the Bloch sphere**.
- The **No-Cloning theorem** states that **you cannot copy quantum states**.
- **Transporting** or **moving** quantum states from a system to another is **possible**.

2 Quantum Algorithms

> Computer science is no more about computers than astronomy is about telescopes
>
> *E.W. Dijkstra*

Algorithms are often **developed long before the machines they are supposed to run on**. Classical algorithms predate classical computers by millennia, and similarly, there exist several quantum algorithms before any large-scale quantum computers have seen the day of light.

All quantum algorithm work with the following basic pattern:

- The system will **start** with qubits **in a particular state**.
- From there the system is put in a **superposition of states**.
- This is followed by **acting on superposition** with several unitary operations.
- And finally, a **measurement** of the qubits.

Of course, there will be several variation of this pattern. However it will be helpful to keep this general scheme in mind as we proceed.

2.1 Deutsch's Algorithm

The simplest quantum algorithm is the Deutsch's algorithm, which is designed to solve a slightly contrived (deliberately created rather than arising naturally or spontaneously) problem.

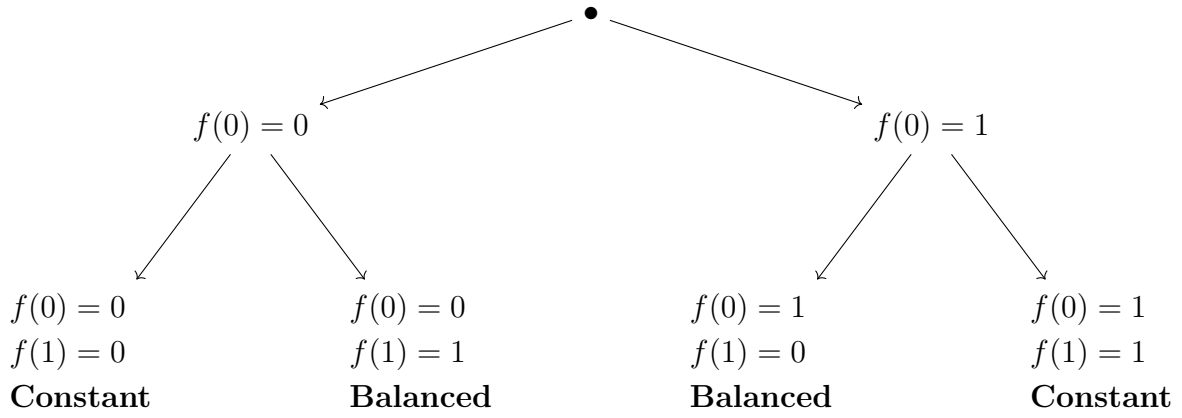
This algorithm is **concerned with functions** from the set $\{0,\ 1\}$ to the set $\{0,\ 1\}$. There are four such functions that might be visualized as:

We call a function $f : \{0,\ 1\} \longrightarrow \{0,\ 1\}$

- **Balanced** if $f(0) \neq f(1)$, in other words if $f(0) = 0$ and $f(1) = 1$ or $f(0) = 1$ and $f(1) = 0$ then f is balanced.
- **Constant** if $f(0) = f(1)$, in other words if $f(0) = f(1) = 0$ or $f(0) = f(1) = 1$ then f is constant.

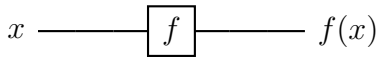
Deutsch's algorithm solves the following problem:
Given a function $f : \{0,\ 1\} \longrightarrow \{0,\ 1\}$ as a black box, where we can evaluate an input but cannot "look inside" and "see" how the function is defined, determine if the function is balanced or constant.

With a classical computer, we would have to evaluate first f with one input (0), then evaluate f on the second input (1), and finally compare the outputs, making it two calls to the function f. The following decision tree shows what a classical computer must do:

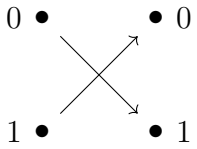


The point is that **with a classical computer**, f must **be evaluated twice**. Can we do better with a quantum computer?

A quantum computer can be in a superposition of two basic states at the same time. **We shall use this superposition of states to evaluate both inputs in one time**. In classical computing, evaluating a given function f corresponds to performing the following operation:



As we saw in the previous courses, such **a function can be thought of as a matrix acting on the input**. For instance, the function:



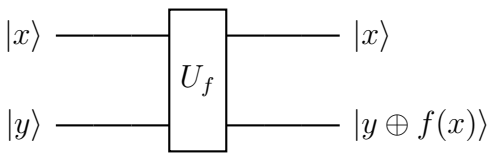
is equivalent to the matrix:

$$\begin{array}{c} \\ \mathbf{0} \\ \mathbf{1} \end{array} \begin{array}{c} \begin{array}{cc} \mathbf{0} & \mathbf{1} \end{array} \\ \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \end{array} \tag{83}$$

Exercise 2.1. Describe the matrices for the other three functions from $\{0,\ 1\}$ to $\{0,\ 1\}$.

However this will not be enough for a quantum system. Such a system requires that **every gate must be unitary (and thus reversible)**. Given the output, we must be able to find the input, and because we don't know if f is balanced or constant we can't always do it.

If f is the name of the function then the following **black box U_f will be the quantum gate that we shall employ to evaluate input**:



The **top input** $|x\rangle$ will be the **qubit value that we wish to evaluate**, and the **bottom input** $|y\rangle$ **controls the output**.

The gate U_f makes the following state operation $|x,\ y\rangle \longrightarrow |x,\ y \oplus f(x)\rangle$, if $y = 0$ this simplifies $|x,\ 0\rangle$ to $|x,\ 0 \oplus f(x)\rangle = |x,\ f(x)\rangle$. In quantum systems, evaluating f **is equivalent to multiplying a state by the unitary matrix** U_f, for example with f such that $f(0) = 1$ and $f(1) = 0$:

$$f = \begin{array}{c} \\ \mathbf{0} \\ \mathbf{1} \end{array} \begin{array}{c} \begin{array}{cc} \mathbf{0} & \mathbf{1} \end{array} \\ \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \end{array} \text{ we get } U_f = \begin{array}{c} \\ \mathbf{00} \\ \mathbf{01} \\ \mathbf{10} \\ \mathbf{11} \end{array} \begin{array}{c} \begin{array}{cccc} \mathbf{00} & \mathbf{01} & \mathbf{10} & \mathbf{11} \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{array} \tag{84}$$

Remember that the **top column name corresponds to the input** $|x,\ y\rangle$ and the **left-hand row name corresponds to the outputs** $|x',\ y'\rangle$. A "1" in the xy column and the $x'y'$ row **means that for the input** $|x,\ y\rangle$**, the output will be** $|x',\ y'\rangle$.

Exercise 2.2. Give the unitary matrices that corresponds to the four functions from $\{0,\ 1\}$ to $\{0,\ 1\}$. Show for each of the matrices that its adjoint is also its inverse and hence all are unitary.

Let us remind ourselves of the task at hand. **We are given such a matrix that express a function but we cannot "look inside" the matrix to "see" how it is defined**.

$$U_f = \begin{array}{c} \\ \mathbf{00} \\ \mathbf{01} \\ \mathbf{10} \\ \mathbf{11} \end{array} \begin{array}{c} \begin{array}{cccc} \mathbf{00} & \mathbf{01} & \mathbf{10} & \mathbf{11} \end{array} \\ \begin{pmatrix} ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \end{pmatrix} \end{array} \tag{85}$$

We are asked to **determine if the function is balanced or constant**.

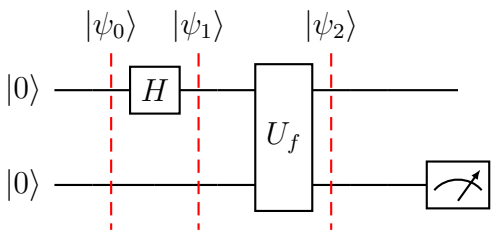
Rather than evaluating f twice, we shall try our trick if **superposition of states**. Instead of having the top input $|x\rangle$ to be either in state $|0\rangle$ or in state $|1\rangle$, we shall put the top input in state

$$|x\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \tag{86}$$

which is **"half-way" between** $|0\rangle$ **and** $|1\rangle$. The **Hadamard** matrix can place a qubit in such state:

$$H\ *\ |0\rangle = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \tag{87}$$

Let us then put all of our inputs to $|0\rangle$ for the moment and put an **Hadamard gate** on the top qubit $|x\rangle$, we thus obtain the following circuit:



The $|\psi_j\rangle$ above the circuit will be used to **describe the state of the qubits at each time step**. In term of matrices this circuit corresponds to

$$U_f\ *\ (H \otimes I)\ *\ (|0\rangle \otimes |0\rangle) = U_f(H \otimes I)(|0,\ 0\rangle) \tag{88}$$

The tensor product $|0,\ 0\rangle$ can be written as

$$\begin{matrix}\mathbf{00}\\ \mathbf{01}\\ \mathbf{10}\\ \mathbf{11}\end{matrix}\begin{pmatrix}1\\0\\0\\0\end{pmatrix} \tag{89}$$

And then the entire circuit is written as

$$U_f\ *\ (H \otimes I)\ *\ \begin{pmatrix}1\\0\\0\\0\end{pmatrix} \tag{90}$$

We shall carefully **examine the states of the system at every time click**. The system starts in

$$|\psi_0\rangle = |0\rangle \otimes |0\rangle = |0,\ 0\rangle \tag{91}$$

We then **apply the Hadamard matrix only on the top input**, leaving the bottom input alone, we get

$$|\psi_1\rangle = \left[\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right] \otimes |0\rangle = \frac{|0,\ 0\rangle + |1,\ 0\rangle}{\sqrt{2}} \tag{92}$$

After **multiplying with $\mathbf{U_f}$**, we have

$$|\psi_2\rangle = \frac{|0,\ f(0)\rangle + |1,\ f(1)\rangle}{\sqrt{2}} \tag{93}$$

For instance, for the function f such that $f(0) = 1$ and $f(1) = 0$, $|\psi_2\rangle$ would be

$$|\psi_2\rangle = \begin{matrix}\\ \mathbf{00}\\ \mathbf{01}\\ \mathbf{10}\\ \mathbf{11}\end{matrix}\begin{matrix}\begin{matrix}\mathbf{00} & \mathbf{01} & \mathbf{10} & \mathbf{11}\end{matrix}\\ \begin{pmatrix}0&1&0&0\\1&0&0&0\\0&0&1&0\\0&0&0&1\end{pmatrix}\end{matrix} * \begin{matrix}\\ \mathbf{00}\\ \mathbf{01}\\ \mathbf{10}\\ \mathbf{11}\end{matrix}\begin{pmatrix}\frac{1}{\sqrt{2}}\\0\\\frac{1}{\sqrt{2}}\\0\end{pmatrix} = \begin{matrix}\\ \mathbf{00}\\ \mathbf{01}\\ \mathbf{10}\\ \mathbf{11}\end{matrix}\begin{pmatrix}0\\\frac{1}{\sqrt{2}}\\\frac{1}{\sqrt{2}}\\0\end{pmatrix} = \frac{|0,\ 1\rangle + |1,\ 0\rangle}{\sqrt{2}} \tag{94}$$

Exercise 2.3. Using the matrices calculated in the exercise 2.2, determine the state of $|\psi_2\rangle$ for the other three functions.

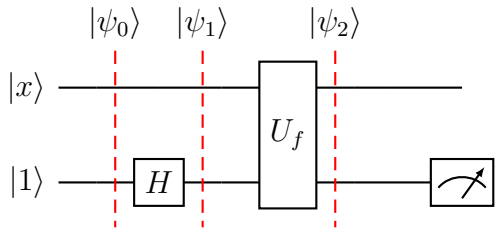
If we measure the **top qubit**, there will be **a 50-50 chance** of finding it in state $|0\rangle$ or in state $|1\rangle$. Similarly **there is no real information to be gotten by measuring the bottom qubit** since we will either get the result of $f(0)$ or $f(1)$ but still not both. We need a batter trick.

Rather than leaving the bottom qubit in state $|0\rangle$, let us put $|y\rangle$ in the **superposition state**:

$$\frac{|0\rangle - |1\rangle}{\sqrt{2}} = \begin{pmatrix}\frac{1}{\sqrt{2}}\\-\frac{1}{\sqrt{2}}\end{pmatrix} = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle \tag{95}$$

Notice the minus sign. Even though there us a negation, **this state is also "half-way" between $|0\rangle$ and $|1\rangle$**. On this qubit, θ has not changed, the qubit $|y\rangle$ **is still on the "equator" on the Bloch sphere, only the phase ϕ has changed**.

This change of phase will help us get our desired results. We can get this superposition of states by multiplying state $|1\rangle$ with the Hadamard matrix. We **shall leave the top qubit as an ambiguous** $|x\rangle$ for the moment.



In term of matrices, this circuit is written as

$$U_f \ * \ (I \otimes H) \ * \ |x, \ 1\rangle \tag{96}$$

Let us look carefully at how the states of the qubits change:

$$|\psi_0\rangle = |x, \ 1\rangle \tag{97}$$

After we **applied Hadamard matrix**, we have

$$|\psi_1\rangle = |x\rangle \otimes \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right] = \frac{|x, \ 0\rangle - |x, \ 1\rangle}{\sqrt{2}} \tag{98}$$

Applying $\mathbf{U_f}$, we get

$$|\psi_2\rangle = |x\rangle \otimes \left[\frac{|0 \oplus f(x)\rangle - |1 \oplus f(x)\rangle}{\sqrt{2}}\right] = |x\rangle \otimes \left[\frac{|f(x)\rangle - \overline{|f(x)\rangle}}{\sqrt{2}}\right] \tag{99}$$

where $\overline{f(x)}$ **means the opposite of** $f(x)$. Therefore, we have

$$|\psi_2\rangle = \begin{cases} |x\rangle \otimes \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right], \text{ if } f(x) = 0 \\ |x\rangle \otimes \left[\frac{|1\rangle - |0\rangle}{\sqrt{2}}\right], \text{ if } f(x) = 1 \end{cases} \tag{100}$$

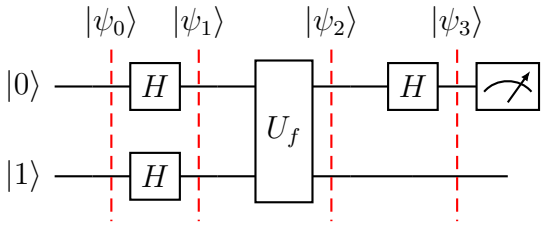
Remembering that $a - b = (-1)(b - a)$, we might write this as

$$|\psi_2\rangle = (-1)^{f(x)} |x\rangle \otimes \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right] = (-1)^{f(x)} \frac{|x, \ 0\rangle - |x, \ 1\rangle}{\sqrt{2}} \tag{101}$$

What would happens if **we measure either the top or the bottom state** ?
Again this does not really help us. We do not gain any information if we measure the top qubit or the bottom qubit.

- The **top qubit** will be in state $|x\rangle$.
- The **bottom qubit** will either be in state $|0\rangle$ or in state $|1\rangle$.

We need something more. Let us **combine our two attempts**, Deutsch's algorithms works by **put both the top and bottom qubit into a superposition**. We will also put the results of the top qubit a Hadamard matrix.



In term of matrices, this becomes

$$(H \otimes I) \ * \ U_f \ * \ (H \otimes H) \ * \ |0, \ 1\rangle \tag{102}$$

or equivalently

$$(H \otimes I) \ * \ U_f \ * \ (H \otimes H) \ * \ \begin{matrix} \mathbf{00} \\ \mathbf{01} \\ \mathbf{10} \\ \mathbf{11} \end{matrix} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \tag{103}$$

At each point of the algorithm, the state are as follows:

$$|\psi_0\rangle = |0,\ 1\rangle \tag{104}$$

$$|\psi_1\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}} = \frac{+|0,\ 0\rangle - |0,\ 1\rangle + |1,\ 0\rangle - |1,\ 1\rangle}{2} = \begin{matrix} \mathbf{00} \\ \mathbf{01} \\ \mathbf{10} \\ \mathbf{11} \end{matrix} \begin{pmatrix} +\frac{1}{2} \\ -\frac{1}{2} \\ +\frac{1}{2} \\ -\frac{1}{2} \end{pmatrix} \tag{105}$$

We saw from our last attempt at solving this problem that when we put the **bottom qubit in superposition** and then multiply by U_f, we will be in the following superposition:

$$(-1)^{f(x)} |x\rangle \otimes \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right] \tag{106}$$

Now, with $|x\rangle$ **in superposition this time**, we have:

$$|\psi_2\rangle = \left[\frac{(-1)^{f(0)} |0\rangle + (-1)^{f(1)} |1\rangle}{\sqrt{2}}\right] \otimes \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right] \tag{107}$$

For example, if $f(0) = 1$ and $f(1) = 0$, the top qubit $|x\rangle$ becomes

$$\frac{(-1) |0\rangle + (+1) |1\rangle}{\sqrt{2}} = (-1) \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right] \tag{108}$$

Exercise 2.4. Describe for the other three functions what $|\psi_2\rangle$ would be.

For a general function f, let us look carefully at

$$(-1)^{f(0)} |0\rangle + (-1)^{f(1)} |1\rangle \tag{109}$$

If f is **constant**, in other words if $f(0) = f(1)$ then this becomes either

$$+1(|0\rangle + |1\rangle) \quad \text{or} \quad -1(|0\rangle + |1\rangle) \tag{110}$$

If f is **balanced**, in other words if $f(0) \neq f(1)$ then it becomes either

$$+1(|0\rangle - |1\rangle) \quad \text{or} \quad -1(|0\rangle - |1\rangle) \tag{111}$$

Summing up, we have

$$|\psi_2\rangle = \begin{cases} (\pm 1) \left[\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right] \otimes \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right], & \text{if } f \text{ is } \mathbf{constant}. \\ (\pm 1) \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right] \otimes \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right], & \text{if } f \text{ is } \mathbf{balanced}. \end{cases} \tag{112}$$

Remembering that **the Hadamard matrix is its own inverse** that takes $\frac{|0\rangle+|1\rangle}{\sqrt{2}}$ to $|0\rangle$ and takes $\frac{|0\rangle-|1\rangle}{\sqrt{2}}$ to $|1\rangle$, when we apply the Hadamard matrix to the **top qubit** $|x\rangle$ we get

$$|\psi_3\rangle = \begin{cases} (\pm 1)\,|0\rangle \otimes \left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right], \text{ if } f \text{ is } \textbf{constant}. \\ (\pm 1)\,|1\rangle \otimes \left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right], \text{ if } f \text{ is } \textbf{balanced}. \end{cases} \tag{113}$$

For example, if $f(0) = 1$ and $f(1) = 0$, then we get

$$|\psi_3\rangle = -1\,|1\rangle \otimes \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right] \tag{114}$$

Exercise 2.5. For each of the other three functions, calculate the value of $|\psi_3\rangle$.

Now **simply measure the top qubit**.

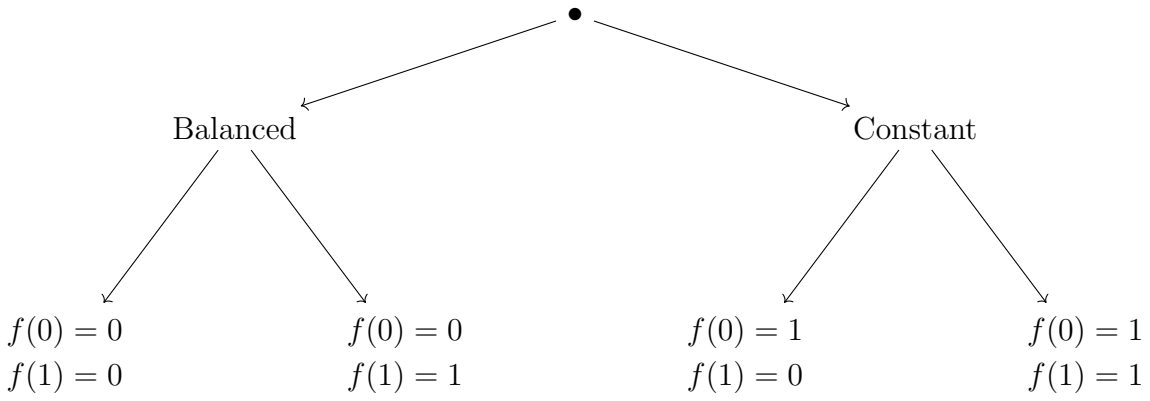
- If it is in state $|\mathbf{0}\rangle$ then f is **constant**.
- If it is in state $|\mathbf{1}\rangle$ then f is **balanced**.

This was all accomplished with **only one function evaluation** as opposed to the two evaluations that the classical algorithm demands.

Notice that the $\pm\mathbf{1}$ tells us **even more information**, namely, which of the the two balanced functions or two constant functions we have, **measurement will not grant us this information**. Upon measuring, if the function is balanced, we will measure $|1\rangle$ regardless if the state was $(-1)\,|1\rangle$ or $(+1)\,|1\rangle$.

You might be bothered by the fact that **the output of the top qubit of U_f should not change from being the same as the input**. However, the inclusion of the **Hadamard** matrix changes things around, this is the essence of the fact that **the top and bottom qubits are entangled**.

Did we perform a magic trick here? Did we gain information that was not there? Not really. There are **four possible functions**, and with a classical computer we needed **2 bits** of information to determine which of the four function we were given.

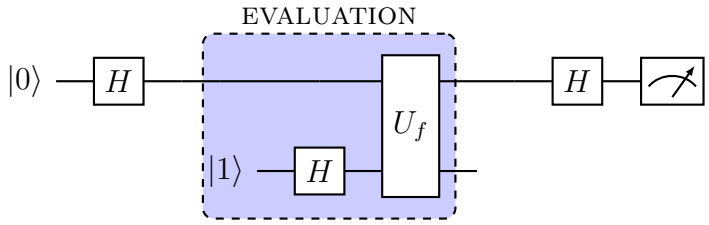


What we are really doing is **changing around the information**. We might determine which of the four function is the case by asking one of the following two questions:

- "Is the function **balanced** or **constant** ?"
- "What is the **value** of the function on 0 ?"

The answer to these two questions uniquely describe each of the four functions, as described in the decision tree above.

The **Hadamard matrices are changing the question that we are asking** (change of basis). The intuition behind the Deutsch's algorithm is that we are really **performing a change of basis** problem. We might rewrite the quantum circuit as



We start in the **canonical basis**. The **first Hadamard matrix** is used as a **change of basis** matrix to go into a balanced superposition of basic states. While in this **non-canonical basis** (Hadamard basis), we **evaluate** f with the **bottom qubit** in a superposition. The **last Hadamard matrix** is used as a change of basis matrix to **revert back to the canonical basis**.

What we have learned 2.1. In this section we discovered our first quantum algorithm, in particular we shall remember that:

- While the Deutsch's algorithm **has no real application**, it however **demonstrate algorithmic quantum speedup** since we can solve a problem with only one function evaluation instead of two necessary on a classical computer.
- **Hadamard** matrix puts a qubit initially in state $|0\rangle$ or $|1\rangle$ in **balanced superposition** meaning that for a qubit $c_0 |0\rangle + c_1 |1\rangle$ we have $|c_0|^2 = |c_1|^2 = 0.5$.

2.2 Shor's Factoring Algorithm

3 Exercises Correction

3.1 Quantum Architecture

3.1.1 Exercise 1.1

Let us begin with the norm $|W|$ of $W = \begin{pmatrix} 3+2i \\ 4-2i \end{pmatrix}$.

$$|W| = \sqrt{\begin{pmatrix} 3-2i & 4+2i \end{pmatrix} \begin{pmatrix} 3+2i \\ 4-2i \end{pmatrix}} = \sqrt{13+20} = \sqrt{33}$$

We can thus write the equivalent physical state as:

$$\frac{W}{\sqrt{33}} = \begin{pmatrix} \frac{3+2i}{\sqrt{33}} \\ \frac{4-2i}{\sqrt{33}} \end{pmatrix} = \frac{3+2i}{\sqrt{33}} |0\rangle + \frac{4-2i}{\sqrt{33}} |1\rangle$$

With a probability of $\frac{13}{33}$ of collapsing to state $|0\rangle$ upon measurement and a probability of $\frac{20}{33}$ of collapsing to state $|1\rangle$ upon measurement.

3.1.2 Exercise 1.2

$$|1\rangle \otimes |0\rangle \otimes |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ 0 \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} \end{pmatrix}$$

$$= \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

$$= \begin{pmatrix} 0 \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \\ 1 \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \end{pmatrix} = \begin{matrix} \mathbf{000} \\ \mathbf{001} \\ \mathbf{010} \\ \mathbf{011} \\ \mathbf{100} \\ \mathbf{101} \\ \mathbf{110} \\ \mathbf{111} \end{matrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

$$|0\rangle \otimes |1\rangle \otimes |1\rangle = \begin{matrix} \mathbf{000} \\ \mathbf{001} \\ \mathbf{010} \\ \mathbf{011} \\ \mathbf{100} \\ \mathbf{101} \\ \mathbf{110} \\ \mathbf{111} \end{matrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$|1\rangle \otimes |1\rangle \otimes |1\rangle = \begin{matrix} \mathbf{000} \\ \mathbf{001} \\ \mathbf{010} \\ \mathbf{011} \\ \mathbf{100} \\ \mathbf{101} \\ \mathbf{110} \\ \mathbf{111} \end{matrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

3.1.3 Exercise 1.3

$$\begin{aligned}
\frac{1}{\sqrt{3}}(|00\rangle - |10\rangle + |11\rangle) &= \frac{1}{\sqrt{3}}(\begin{pmatrix}1\\0\end{pmatrix} \otimes \begin{pmatrix}1\\0\end{pmatrix} - \begin{pmatrix}0\\1\end{pmatrix} \otimes \begin{pmatrix}1\\0\end{pmatrix} + \begin{pmatrix}0\\1\end{pmatrix} \otimes \begin{pmatrix}0\\1\end{pmatrix}) \\
&= \frac{1}{\sqrt{3}}(\begin{pmatrix}1 \cdot \begin{pmatrix}1\\0\end{pmatrix} \\ 0 \cdot \begin{pmatrix}1\\0\end{pmatrix}\end{pmatrix} - \begin{pmatrix}0 \cdot \begin{pmatrix}1\\0\end{pmatrix} \\ 1 \cdot \begin{pmatrix}1\\0\end{pmatrix}\end{pmatrix} + \begin{pmatrix}0 \cdot \begin{pmatrix}0\\1\end{pmatrix} \\ 1 \cdot \begin{pmatrix}0\\1\end{pmatrix}\end{pmatrix}) \\
&= \frac{1}{\sqrt{3}}(\begin{pmatrix}1\\0\\0\\0\end{pmatrix} - \begin{pmatrix}0\\0\\1\\0\end{pmatrix} + \begin{pmatrix}0\\0\\0\\1\end{pmatrix}) \\
&= \frac{1}{\sqrt{3}}\begin{pmatrix}1\\0\\-1\\1\end{pmatrix}
\end{aligned}$$

3.1.4 Exercise 1.4

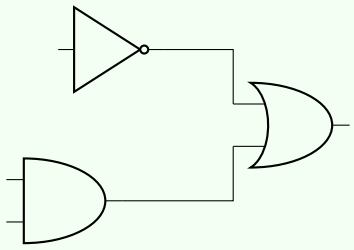
$$\begin{aligned}
3\,|01\rangle + 2\,|11\rangle = 3\,|0 \otimes 1\rangle + 2\,|1 \otimes 1\rangle &= 3 \cdot (\begin{pmatrix}1\\0\end{pmatrix} \otimes \begin{pmatrix}0\\1\end{pmatrix}) + 2 \cdot (\begin{pmatrix}0\\1\end{pmatrix} \otimes \begin{pmatrix}0\\1\end{pmatrix}) \\
&= 3 \cdot \begin{pmatrix}1 \cdot \begin{pmatrix}0\\1\end{pmatrix} \\ 0 \cdot \begin{pmatrix}0\\1\end{pmatrix}\end{pmatrix} + 2 \cdot \begin{pmatrix}0 \cdot \begin{pmatrix}0\\1\end{pmatrix} \\ 1 \cdot \begin{pmatrix}0\\1\end{pmatrix}\end{pmatrix} \\
&= 3 \cdot \begin{pmatrix}0\\1\\0\\0\end{pmatrix} + 2 \cdot \begin{pmatrix}0\\0\\0\\1\end{pmatrix} \\
&= \begin{matrix}\mathbf{00}\\\mathbf{01}\\\mathbf{10}\\\mathbf{11}\end{matrix} \begin{pmatrix}0\\3\\0\\2\end{pmatrix}
\end{aligned}$$

3.1.5 Exercise 1.5

$$\begin{aligned} NOT * OR &= \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} * \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix} \\ &= \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix} \end{aligned}$$

In other words, NOR outputs 1 only if both inputs are 0.

3.1.6 Exercise 1.6



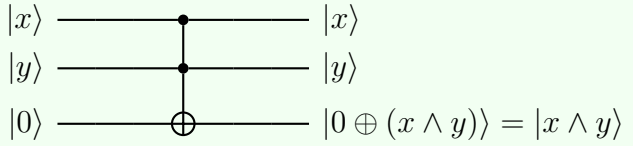
Since we have first two parallel operations NOT and AND, then we apply sequentially an OR gate, we can represent this circuit by the matrix:

$$\begin{aligned} OR \ * \ (NOT \otimes AND) &= OR * \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \otimes \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ &= OR \ * \ \begin{pmatrix} 0 \cdot \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} & 1 \cdot \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ 1 \cdot \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} & 0 \cdot \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix} * \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \\ &= \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \end{aligned}$$

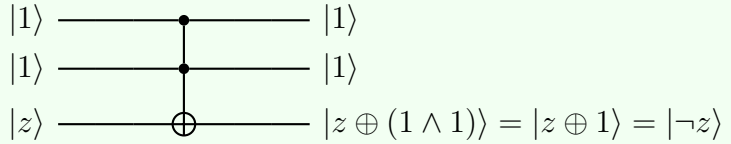
Which is a 2^1-by-2^3 matrix, it makes sense since our circuit takes 3 inputs and have 1 output.

3.1.7 Exercise 1.7

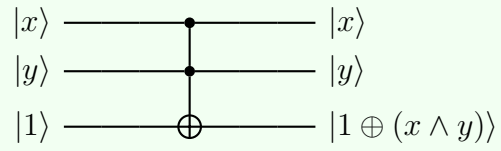
The AND gate can be obtained by setting the bottom $|z\rangle$ input to $|0\rangle$:



The NOT gate is obtained by setting the top two inputs to $|1\rangle$:

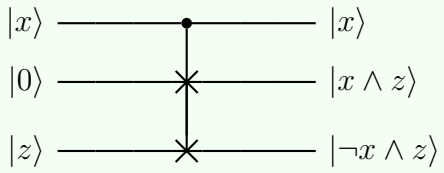


The NAND gate can be obtained by setting the bottom $|z\rangle$ input to $|1\rangle$:

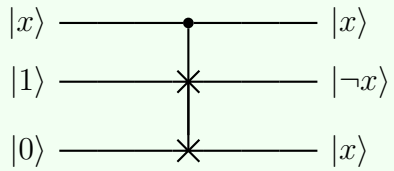


3.1.8 Exercise 1.8

By setting $|y\rangle$ to $|0\rangle$ we obtain the AND gate as follows:



And the NOT gate and the fanout gate can be obtained by setting $|y\rangle$ to $|1\rangle$ and $|z\rangle$ to $|0\rangle$, we thus obtain:



3.1.9 Exercise 1.9

So let us begin with the X gate:

$$X^\dagger = X^T = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = X$$

Then $X * X^\dagger = X^2 = I$ so X is unitary.
Let us see what happens when we apply X to $|\psi\rangle$:

$$X * |\psi\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} * \begin{pmatrix} c_0 \\ c_1 \end{pmatrix} = \begin{pmatrix} c_1 \\ c_0 \end{pmatrix}$$

Let us see the case of the Y gate:

$$Y^\dagger = (\overline{Y})^T = \begin{pmatrix} 0 & \overline{(i)} \\ \overline{(-i)} & 0 \end{pmatrix} = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} = Y$$

Since $Y * Y^\dagger = Y^2 = I$ then Y is unitary.
If we apply Y to $|\psi\rangle$ we get:

$$Y * |\psi\rangle = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} * \begin{pmatrix} c_0 \\ c_1 \end{pmatrix} = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} * \begin{pmatrix} a_0 + b_0 i \\ a_1 + b_1 i \end{pmatrix} = \begin{pmatrix} b_1 - a_1 i \\ -b_0 + a_0 i \end{pmatrix}$$

The Hermitian conjugate of the Z gate is:

$$Z^\dagger = Z^T \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = Z$$

Then multiplying Z by its Hermitian conjugate $Z * Z^\dagger = Z^2 = I$ gives us the identity matrix then Z is unitary. If we apply Z to $|\psi\rangle$ we obtain:

$$Z * |\psi\rangle = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} * \begin{pmatrix} c_0 \\ c_1 \end{pmatrix} = \begin{pmatrix} c_0 \\ -c_1 \end{pmatrix}$$

For the S gate we have:

$$S^\dagger = (\overline{S})^T = \begin{pmatrix} 1 & 0 \\ 0 & (\overline{i}) \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & -i \end{pmatrix}$$

We can then trivially see that $S * S^\dagger = I$ and conclude that S is unitary.
If we apply S to $|\psi\rangle$ we then get:

$$S * |\psi\rangle = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} * \begin{pmatrix} a_0 + b_0 i \\ a_1 + b_1 i \end{pmatrix} = \begin{pmatrix} a_0 + b_0 i \\ -b_1 + a_1 i \end{pmatrix}$$

And finally the Hermitian conjugate of the T gate is:

$$T^\dagger = (\overline{T})^T = \begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{i\pi}{4}} \end{pmatrix} = T^\dagger = (\overline{T})^T = \begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{-i\pi}{4}} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & \frac{1-i}{\sqrt{2}} \end{pmatrix}$$

We can show that it is unitary with the trivial multiplication:

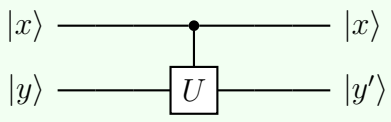
$$T * T^\dagger = \begin{pmatrix} 1 & 0 \\ 0 & \frac{1+i}{\sqrt{2}} \end{pmatrix} * \begin{pmatrix} 1 & 0 \\ 0 & \frac{1-i}{\sqrt{2}} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = I_2$$

And if we apply T to $|\psi\rangle$ we obtain:

$$T * |\psi\rangle = \begin{pmatrix} 1 & 0 \\ 0 & \frac{1+i}{\sqrt{2}} \end{pmatrix} * \begin{pmatrix} a_0 + b_0 i \\ a_1 + b_1 i \end{pmatrix} = \begin{pmatrix} a_0 + b_0 i \\ \frac{(1+i)(a_1+b_1 i)}{\sqrt{2}} \end{pmatrix} = \begin{pmatrix} a_0 + b_0 i \\ \frac{1}{\sqrt{2}} \cdot ((a_1 - b_1) + (a_1 + b_1) \cdot i) \end{pmatrix}$$

3.1.10 Exercise 1.10

In this exercise we consider the following one-qubit operation:



Represented by the following matrix:

$$^C U = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & a & b \\ 0 & 0 & c & d \end{pmatrix}$$

Let us consider a generic qubit $|y\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$ such that $\begin{pmatrix}\alpha\\ \beta\end{pmatrix} \in \mathbb{C}^2$ and $|\alpha|^2 + |\beta|^2 = 1$.

If $|x\rangle = |0\rangle$ then applying ${}^{C}U$ to $|y\rangle$ can be described as:

$$
\begin{aligned}
{}^{C}U \ * \ (|0\rangle \otimes |y\rangle) = {}^{C}U \ * \ |0y\rangle &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & a & b \\ 0 & 0 & c & d \end{pmatrix} \ * \ \begin{pmatrix} 1 \cdot \begin{pmatrix}\alpha\\ \beta\end{pmatrix} \\ 0 \cdot \begin{pmatrix}\alpha\\ \beta\end{pmatrix} \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & a & b \\ 0 & 0 & c & d \end{pmatrix} \ * \ \begin{pmatrix} \alpha \\ \beta \\ 0 \\ 0 \end{pmatrix} \\
&= \begin{pmatrix} \alpha \\ \beta \\ 0 \\ 0 \end{pmatrix} \\
&= I \ * \ (|0\rangle \otimes |y\rangle) \\
&= |0\rangle \otimes |y\rangle
\end{aligned}
$$

As we can see above, when $|x\rangle = |0\rangle$, both qubits remain unchanged, as if $U = I$.

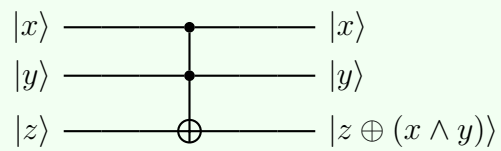
If $|x\rangle = |1\rangle$ then applying ${}^{C}U$ to $|y\rangle$ can be described as:

$$
\begin{aligned}
{}^{C}U \ * \ (|1\rangle \otimes |y\rangle) = {}^{C}U \ * \ |1y\rangle &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & a & b \\ 0 & 0 & c & d \end{pmatrix} \ * \ \begin{pmatrix} 0 \cdot \begin{pmatrix}\alpha\\ \beta\end{pmatrix} \\ 1 \cdot \begin{pmatrix}\alpha\\ \beta\end{pmatrix} \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & a & b \\ 0 & 0 & c & d \end{pmatrix} \ * \ \begin{pmatrix} 0 \\ 0 \\ \alpha \\ \beta \end{pmatrix} \\
&= \begin{pmatrix} 0 \\ 0 \\ a \cdot \alpha + b \cdot \beta \\ c \cdot \alpha + d \cdot \beta \end{pmatrix} \\
&\neq |1\rangle \otimes |y\rangle
\end{aligned}
$$

As we can see above, when $|x\rangle = |1\rangle$, the qubit $|y\rangle$ is modified while the control qubit $|x\rangle$ remain unchanged.

3.1.11 Exercise 1.11

As we saw in this lecture, the Toffoli gate is nothing more than a controlled-controlled-NOT (a NOT gate with 2 control qubits) or CCNOT:



Since the Toffoli gate have 3 input qubits and 3 output qubits it is represented this $2^3 = 8\text{-by-}8 = 2^3$ matrix:

$$TOFFOLI = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Let us assume that $|x\rangle = |y\rangle = 1$ meaning that our Toffoli gate will flip the generic qubit $|\psi\rangle = cos\theta\,|0\rangle + e^{i\phi}sin\theta\,|1\rangle$, if we only take this qubit alone with the NOT operation, we will obtain:

$$|\psi_{flipped}\rangle = NOT\ *\ |\psi\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\ *\ \begin{pmatrix} cos\theta \\ e^{i\phi}sin\theta \end{pmatrix} = \begin{pmatrix} e^{i\phi}sin\theta \\ cos\theta \end{pmatrix} = e^{i\phi}sin\theta\,|0\rangle + cos\theta\,|1\rangle$$

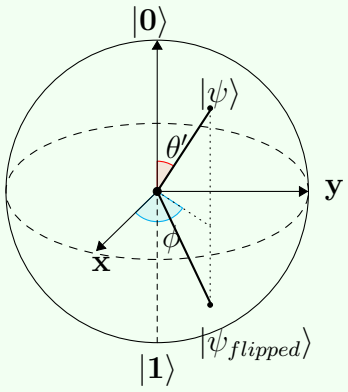
Which can be rewritten as:

$$|\psi_{flipped}\rangle = e^{i\phi}cos(\frac{\pi}{2} - \theta) \cdot |0\rangle + sin(\frac{\pi}{2} - \theta) \cdot |1\rangle$$

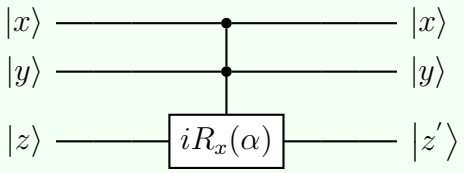
If we map $|\psi\rangle$ and $|\psi_{flipped}\rangle$ on the Bloch sphere where $\theta' = 2\theta$ when $|\psi\rangle = cos\theta\,|0\rangle + e^{i\phi}sin\theta\,|1\rangle$ we obtain the following point on the sphere:

$$|\psi\rangle \xrightarrow{Bloch} (2\theta, \phi) = (\theta', \phi)$$
$$|\psi_{flipped}\rangle \xrightarrow{Bloch} (\pi - 2\theta, \phi) = (\pi - \theta', \phi)$$

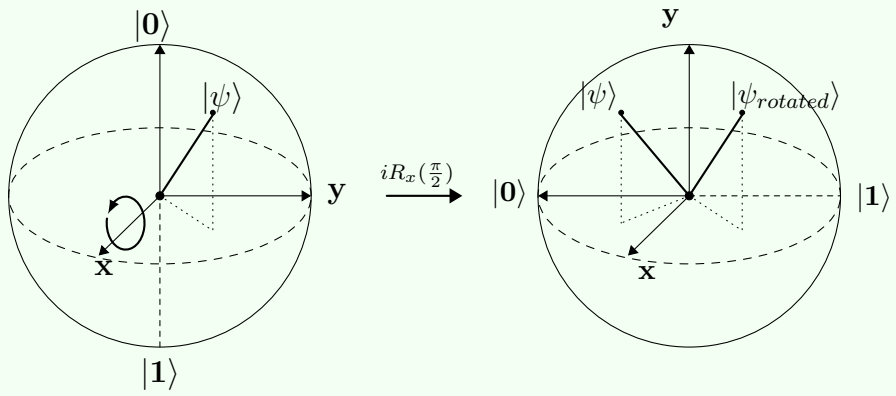
Which give us on the sphere a points $|\psi_{flipped}\rangle$ which is symmetrical at point $|\psi\rangle$ with respect to the plane formed by the x and y axes (the equator), notice that the angle between $|\psi\rangle$ and $|\psi_{flipped}\rangle$ is now of $90° = \frac{\pi}{2}\ rad$.



Now let us consider the Deutsch gate which is a controlled-controlled-$R_x(\alpha)$ or in other words a rotation around the x with two control qubits:



Now, still assuming the two control qubits are both equal to $|1\rangle$, let us see what happens when we apply $R_x(\frac{\pi}{2})$ to $|\psi\rangle$ on the Bloch sphere:



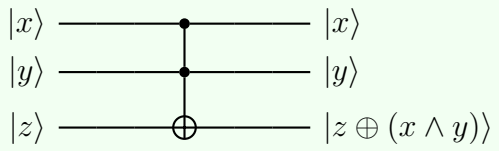
It looks like $|\psi_{flipped}\rangle$ and $|\psi_{rotated}\rangle$ respectively have the same coordinates on the Bloch sphere and share the same symmetry with respect to the plan formed by the x and the y axes, let us very this statement. With the two control qubits equal to $|1\rangle$, applying $D(\frac{\pi}{2})$ means we will apply $iR_x(\frac{\pi}{2})$ to $|\psi\rangle$:

$$
\begin{aligned}
|\psi_{rotated}\rangle = iR_x(\frac{\pi}{2}) \ * \ |\psi\rangle &= \begin{pmatrix} i\ cos\frac{\pi}{2} & sin\frac{\pi}{2} \\ sin\frac{\pi}{2} & i\ cos\frac{\pi}{2} \end{pmatrix} * \begin{pmatrix} cos\theta \\ e^{i\phi}sin\theta \end{pmatrix} \\
&= \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} * \begin{pmatrix} cos\theta \\ e^{i\phi}sin\theta \end{pmatrix} \\
&= \begin{pmatrix} e^{i\phi}sin\theta \\ cos\theta \end{pmatrix} \\
&= |\psi_{flipped}\rangle
\end{aligned}
$$

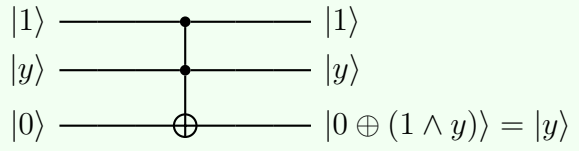
Since we get the same result using either the Toffoli gate and using the Deutsch gate at $D(\frac{\pi}{2})$, we can say that the Toffoli gate is equivalent to $D(\frac{\pi}{2})$.

3.1.12 Exercise 1.12

The Toffoli gate is defined by the circuit



And the following configuration of the Toffoli gate with the state $|1,\ y,\ 0\rangle$ mimics the fanout gates for classical bits:



Let us consider this configuration where $|z\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$ is a superposition of states, giving

us the following starting state of the system:

$$|x\rangle \otimes |y\rangle \otimes |z\rangle = |xyz\rangle = |1,\ y,\ 0\rangle = \begin{matrix}\mathbf{000}\\ \mathbf{001}\\ \mathbf{010}\\ \mathbf{011}\\ \mathbf{100}\\ \mathbf{101}\\ \mathbf{110}\\ \mathbf{111}\end{matrix}\begin{pmatrix}0\\0\\0\\0\\\frac{1}{\sqrt{2}}\\0\\\frac{1}{\sqrt{2}}\\0\end{pmatrix}$$

Multiplying this state with the Toffoli gate gives us

$$\begin{matrix}\\ \mathbf{000}\\ \mathbf{001}\\ \mathbf{010}\\ \mathbf{011}\\ \mathbf{100}\\ \mathbf{101}\\ \mathbf{110}\\ \mathbf{111}\end{matrix}\begin{matrix}\begin{matrix}\mathbf{000} & \mathbf{001} & \mathbf{010} & \mathbf{011} & \mathbf{100} & \mathbf{101} & \mathbf{110} & \mathbf{111}\end{matrix}\\ \begin{pmatrix}1&0&0&0&0&0&0&0\\0&1&0&0&0&0&0&0\\0&0&1&0&0&0&0&0\\0&0&0&1&0&0&0&0\\0&0&0&0&1&0&0&0\\0&0&0&0&0&1&0&0\\0&0&0&0&0&0&0&1\\0&0&0&0&0&0&1&0\end{pmatrix}\end{matrix} * \begin{pmatrix}0\\0\\0\\0\\\frac{1}{\sqrt{2}}\\0\\\frac{1}{\sqrt{2}}\\0\end{pmatrix} = \begin{pmatrix}0\\0\\0\\0\\\frac{1}{\sqrt{2}}\\0\\0\\\frac{1}{\sqrt{2}}\end{pmatrix}$$

Which is equal to:

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

The operation

$$\left(1,\ \frac{|0\rangle + |1\rangle}{\sqrt{2}},\ 0\right) \longrightarrow \frac{|1,\ 0,\ 0\rangle + |1,\ 1,\ 1\rangle}{\sqrt{2}}$$

is not a fanout operation, therefore the Toffoli gate does not violate the no-cloning theorem.

3.2 Quantum Algorithms