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.

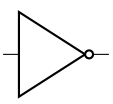
1.2 Classical Gates

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 $S_{input} \in \mathbb{C}^n$, we have then:

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

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



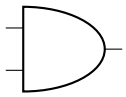
This gate take as input one bit (represented by a 2-by-1 matrix), and outputs one bit. NOT of $|0\rangle$ equals $|1\rangle$ and NOT of $|1\rangle$ equals $|0\rangle$. We can represent such an operation with the matrix:

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

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:

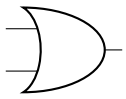
$$AND\,|01\rangle = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |0\rangle \tag{23}$$

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.

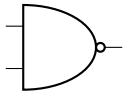
Let us continue to the OR gate.



Which can be represented by:

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

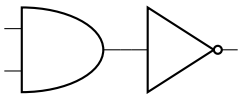
And finally the NAND gate, which is of special importance since **every logical gate can be composed of NAND gates**. It takes 2 bits for input and outputs 1 bit.



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 $x = z = 1$.

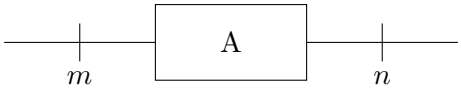
Exercise 1.5. Find a matrix that corresponds to the NOR gate.

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**.

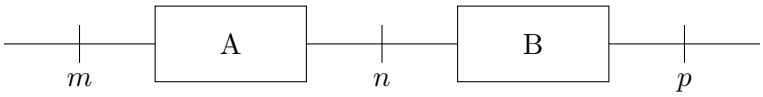
We call this procedure performing **sequential** operations. If matrix A corresponds to performing an operation and matrix B corresponds to another operation, then **the matrix product** $B * A$ (and **not** $A * B$) **corresponds to performing those two operations sequentially in the same order**.

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:

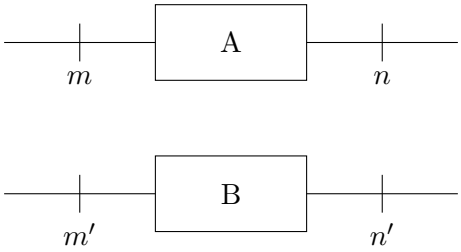


The matrix A would be of size 2^n-by2^m. Say B takes the n outputs of A as input and outputs p bits, thus we obtain:



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:

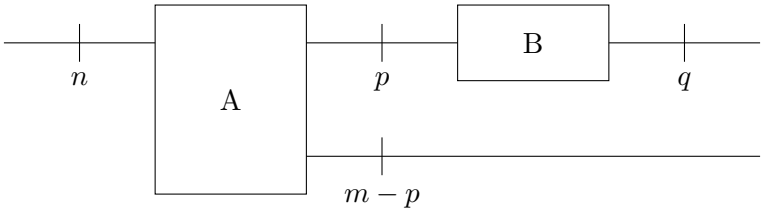


In this case we have A acting on some bits and B acting on others, this would be represented by **the tensor product** $A \otimes B$.

- 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.

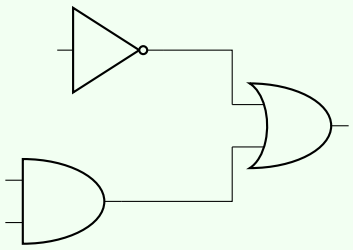


- A is a 2^m-by-2^n matrix.
- B is 2^q-by-2^p matrix.

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

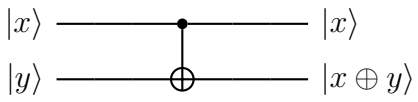
Notion goal 1.3. Not all of the logical gates we saw in the last section would work in quantum computers. As we saw in the last lecture, **all quantum operations that are not measurement are reversible and are represented by unitary matrices**.
For the quantum world we would need reversible gates, which 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.

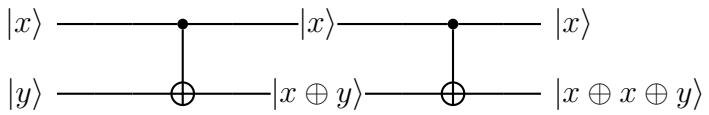
- If $|x\rangle = |0\rangle$ then the bottom output of $|y\rangle$ **will be the same as the input**.
- If $|x\rangle = |1\rangle$ then the bottom output of $|y\rangle$ **will be the opposite**.

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:

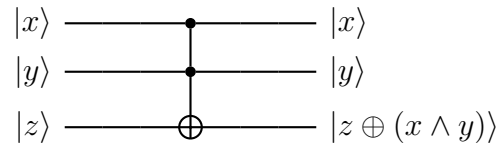
$$\begin{array}{c} \\ \mathbf{00} \\ \mathbf{01} \\ \mathbf{10} \\ \mathbf{11} \end{array} \begin{array}{c} \begin{array}{cccc} \mathbf{00} & \mathbf{01} & \mathbf{10} & \mathbf{11} \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \end{array}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

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**.

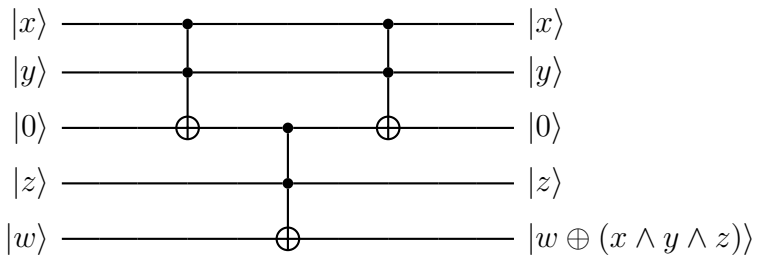
The matrix corresponding this gate is:

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

Let us review what we have seen so far:

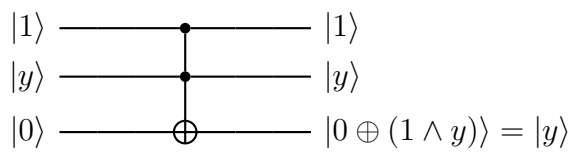
- The **NOT** gate **has no controlling bit**.
- The **CNOT** gate **has one controlling bits**.
- The **Toffoli** gate **has two controlling bits**.

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.

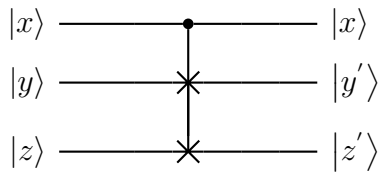
Example 1.5. In order to be able to construct all gates, we must also have a way of producing a fanout of values. In others words, a gate is needed that inputs a value and output two of the same values, this can be obtained by setting the value $|x\rangle$ to $|1\rangle$ and $|z\rangle$ to $|0\rangle$.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

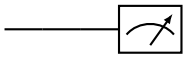
$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \xrightarrow{\sqrt{NOT}} \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} \xrightarrow{\sqrt{NOT}} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |1\rangle \tag{42}$$

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 (complex coefficient) for $|\psi\rangle$ 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.

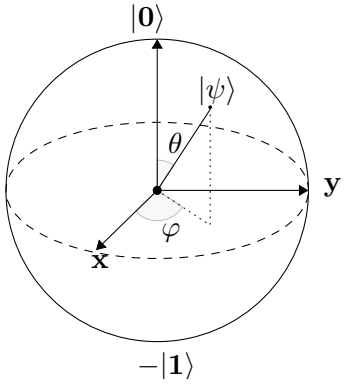


Figure 1: The Bloch sphere

2 Quantum Algorithms

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

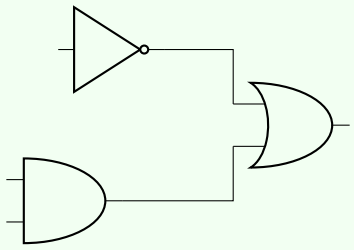
$$\begin{aligned}
3\,|01\rangle + 2\,|11\rangle = 3\,|0 \otimes 1\rangle + 2\,|1 \otimes 1\rangle &= 3 \cdot (\begin{pmatrix}1\\0\end{pmatrix} \otimes \begin{pmatrix}0\\1\end{pmatrix}) + 2 \cdot (\begin{pmatrix}0\\1\end{pmatrix} \otimes \begin{pmatrix}0\\1\end{pmatrix}) \\
&= 3 \cdot \begin{pmatrix}1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

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



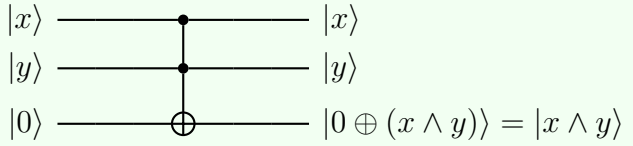
Since we have first two parallel operations NOT and AND, then we apply sequentially an OR gate, we can represent this circuit by the matrix:

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

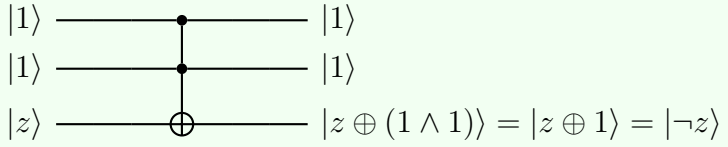
Which is a 2^1-by-2^3 matrix, it makes sense since our circuit takes 3 inputs and have 1 output.

3.1.7 Exercise 1.7

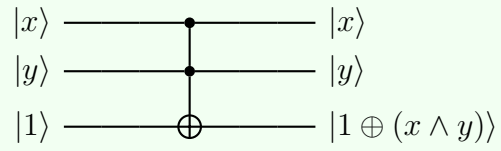
The AND gate can be obtained by setting the bottom $|z\rangle$ input to $|0\rangle$:



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

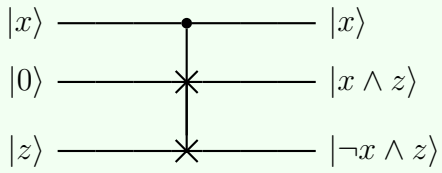


The NAND gate can be obtained by setting the bottom $|z\rangle$ input to $|1\rangle$:

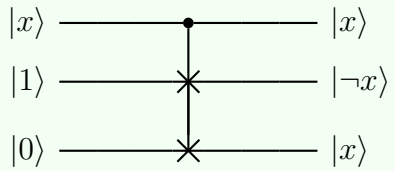


3.1.8 Exercise 1.8

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:

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

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:

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

3.2 Quantum Algorithms