Tina Hu, Maria Berova
MATH 136
June 4, 2023

Introduction to Quantum Computing

Contents

1. Introduction

Classical computing, also known as binary computing, utilizes transistors and bits to store and transmit information. A bit in classical computing has state that is either 1 (signal) or 0 (no signal). A transistor, acting like a switch, represents a bit by being on (1) or off (0). At any single point in time, a set of n transistors can only represent one state (each transistor has two possible states, so there exist 2^n possible states).

The computing power of a classical computer grows linearly with number of transistors used, and therefore with the size of the computer. In order to speed up a process, one must increase the number of transistors running parallel to each other.

Finally, classical gates implemented with transistors (such as the well-known NOT, AND, OR gates) are not reversible.

Unlike classical computing, quantum computing uses quantum bits, also known as qubits. Each qubit's state is in a superposition (or more simply, combination) of the "0" state and "1" state. This does not mean that the qubit is a "mix" of 0 and 1, or "in-between" 0

and 1. In fact, the true value of the qubit is unknown until we **measure** it, and there is a certain probability of it being 0 or 1. A quantum computer utilizes quantum bits rather than transistors. Thus, since each qubit represents "both 1 and 0", a set of n qubits represents a superposition of all of the 2^n possible states. So, in theory, we can use the same amount of particles as in a classical computer to store much more information. Note, however, that only one state can be extracted from our set of n qubits.

Quantum gates applied to qubits are reversible. Therefore, we can apply some transformation to our set of qubits, and later undo it, allowing us to see past results.

1.1. **How Quantum Computing Works: A Brief Analogy.** To give us more intuition regarding how a quantum computer differs from a classical computer, imagine we would like to calculate some function $f(x)$ at $x = a$. If we are using a classical computing method, the computer would be given the input a, and definitely return $f(a)$. In quantum computing, however, the input would be a superposition of all possible x in the domain of f, and after operations are performed, the output would be a superposition of all possible $f(x)$. To obtain $f(a)$, one would need to use wave interference to amplify the probability of $f(a)$ being measured (as only one bit worth of information can be extracted from a qubit).

Generally, quantum computing is governed by Schrodinger's equation and some quantum theory (both of which are not discussed in this paper). A qubit is not a definite type of particle, it is just a description of some properties of a hypothetical particle that can be used for quantum computing. Qubits used for computing can be naturally occurring or synthetically made. Some examples of qubits include trapped atoms and ions, photons, and superconducting circuits. (As an interesting side note, the particles usually used for qubits in a quantum computer are chosen such that they exhibit wave-particle duality, which is why we can use interference to amplify results.)

The three main properties of a qubit are:

1) **Entanglement**: in a system of multiple entangled qubits, the measurement of one qubit affects the measurement of the others. Alternatively, the state of a system of multiple entangled qubits cannot be represented by the states of the individual qubits – the system as a whole takes on a completely different, unique form.
2) **Interference**: as previously mentioned, the real particle used as a qubit in a quantum computer exhibits wave-particle duality, so we can use wave-like interference to amplify certain component of its state
3) **Superposition**: the value that a qubit represents (0 or 1) is unknown until measurement

Note that none of these properties are present in classical bits. For our linear algebra viewpoint, the property of entanglement is especially interesting, and will be discussed later. However, the other two properties are included only to provide more insight for the reader.

2. Main postulates and Notation

1) We represent the state of each qubit as a unit vector in $\mathbb{C}^2$. These vectors x have a "ket" representation $|x\rangle$. The basis vectors for the state of a qubit in $\mathbb{C}$ are $|0\rangle := \begin{pmatrix} 1 & 0 \end{pmatrix}^T$ and $|1\rangle := \begin{pmatrix} 0 & 1 \end{pmatrix}^T$. Furthermore, each vector x has a conjugate transpose

which we call "bra" (to get the name bra/ket notation): $\langle x|$. For example: $\langle 0| := (1\ 0)$ and $\langle 1| := (0\ 1)$. So we get the convenient bra/ket notation, giving us $\langle x||x\rangle := \langle x|x\rangle$ = inner product of vector $|x\rangle$ with itself.

2) Individual quantum states of qubits combine through the tensor product (defined and discussed in detail later).
3) Qubits evolve over time naturally, and the evolutions are dictated by unitary transformations. This is to say, the state of a qubit changes with respect to time even when left alone. The state change of qubit is described by the Schrodinger's equation. According to this website, each evolution of a qubit can be represented by a unitary transformation. Thus, when we create a quantum computer, we want to mostly use unitary gates to reflect the way qubits naturally evolve.
4) Measuring a quantum state collapses/projects the quantum state to one of the bases associated with the measuring device, with some probability associated with each basis. Importantly, measurement is one of the few non-unitary operations which we are allowed to discuss in the context of quantum computers (we also discuss this further in the paper).

3. Qubits basics

3.1. **Tensor Product (aka Outer Product).** Before we dive into the math of quantum computing, we will first go over an important mathematical construct used in quantum mechanics: the tensor product.

Tensor products are used to combine the state of multiple individual qubits into the state of a system of the qubits. Given two vectors $\vec{v} = \begin{pmatrix} a & b & c \end{pmatrix}^T$ and $\vec{w} = \begin{pmatrix} d & e \end{pmatrix}^T$, we define the tensor product

$$\vec{v} \otimes \vec{w} = \begin{pmatrix} a \\ b \\ c \end{pmatrix} \begin{pmatrix} d & e \end{pmatrix} = \begin{pmatrix} ad & ae \\ bd & be \\ cd & ce \end{pmatrix}$$

We can also reshape the matrix (or define a isomorphism from the matrix representation to a column vector representation) and get

$$\vec{v} \otimes \vec{w} = \begin{pmatrix} ad & ae & bd & be & cd & ce \end{pmatrix}^T$$

Unlike Cartesian product (in which the elements and dimension of given vector spaces add up), the dimension of given vector spaces multiplies in tensor product. If we let V be a 3 dimensional vector space with basis $\{\vec{v_1}, \vec{v_2}, \vec{v_3}\}$ and W be a 2 dimensional vector space with basis $\{\vec{w_1}, \vec{w_2}\}$, then the set of new basis for $V \otimes W$ is

	$\vec{w}_1$	$\vec{w}_2$
$\vec{v}_1$	$\vec{v}_1 \otimes \vec{w}_1$	$\vec{v}_1 \otimes \vec{w}_2$
$\vec{v}_2$	$\vec{v}_2 \otimes \vec{w}_1$	$\vec{v}_2 \otimes \vec{w}_2$
$\vec{v}_3$	$\vec{v}_3 \otimes \vec{w}_1$	$\vec{v}_3 \otimes \vec{w}_2$

Therefore, our new space $V \otimes W$ has a dimension of $3 \times 2 = 6$.

Some properties of tensor products are:

- $(\vec{v_1} + \vec{v_2}) \otimes \vec{v_3} = \vec{v_1} \otimes \vec{v_3} + \vec{v_2} \otimes \vec{v_3}$
- $\vec{v_3} \otimes (\vec{v_1} + \vec{v_2}) = \vec{v_3} \otimes \vec{v_1} + \vec{v_3} \otimes \vec{v_2}$
- $a\vec{v_1} \otimes b\vec{v_2} = ab(\vec{v_1} \otimes \vec{v_2})$, a,b are scalars

Which may be verified using the definition of the tensor product.

Now, suppose we have two spaces in which we represent the states of two individual vectors. As mentioned in the introduction, these spaces have bases "0" and "1", so they are two-dimensional. After combining the two spaces using the tensor product, we will have a four-dimensional space. Thus, as previously alluded, the power of a quantum computer increases exponentially – the dimensions of computing spaces are multiplied when we combine them.

3.2. **Systems of one qubit.** We have seen that quantum state spaces are described in terms of vectors, matrices, and the bra/ket notation. For a quantum bit, we fix the basis $\{|0\rangle, |1\rangle\}$ (associated with the orthonormal basis of the measuring device). The basis vectors represent the classical bits 0 and 1. This allows us to express quantum bits as being in a "superposition" of 0 and 1, for example, the bit $a|0\rangle + b|1\rangle$ represents neither zero nor one. While a quantum bit can store infinitely many states, we may only extract one classical bit of information from it. This is because when a qubit is measured, the measurement changes the state to one of the basis vectors. However, if given the state of the qubit, we may attempt to predict what state will be extracted from a qubit by measurement. If we are given a qubit

$$a|0\rangle + b|1\rangle$$

Then the probability of measurement returning $|0\rangle$ is a^2, and the probability of measurement returning $|1\rangle$ is b^2. This holds because the vector is a unit vector (see postulates), meaning $a^2 + b^2 = 1$. Therefore the squared components represent probabilities of measurement, adding to one.

3.3. **Systems of multiple qubits.** When more than one qubit is introduced, the system becomes more complicated to work with. In fact, the measurement of one qubit in the system may affect the measurement of any other qubit in the same system! This property is called entanglement. The state of multiple qubits (recall, the state of each individual qubit is modeled by the basis $\{|0\rangle, |1\rangle\}$) combines through the tensor product. Given two qubits, q_1 with state $a|0\rangle + b|1\rangle$ and q_2 with state $c|0\rangle + d|1\rangle$, we get their joint state to be $(a|0\rangle + b|1\rangle) \otimes (c|0\rangle + d|1\rangle)$

$$= (a|0\rangle \otimes c|0\rangle) + (a|0\rangle \otimes d|1\rangle) + (b|1\rangle \otimes c|0\rangle) + (b|1\rangle \otimes d|1\rangle)$$

$$= ac(|0\rangle \otimes |0\rangle) + ad(|0\rangle \otimes |1\rangle) + bc(|1\rangle \otimes |0\rangle) + bd(|1\rangle \otimes |1\rangle)$$

To obtain a cleaner result, we use the notation $|x\rangle \otimes |y\rangle = |xy\rangle$, so we have

$$= ac|00\rangle + ad|01\rangle + bc|10\rangle + bd|11\rangle$$

where $\{|00\rangle, |01\rangle\, |10\rangle, |11\rangle\}$ are the new basis for the state of a 2-qubit system

$$|00\rangle = \begin{pmatrix}1\\0\\0\\0\end{pmatrix}, |01\rangle = \begin{pmatrix}0\\1\\0\\0\end{pmatrix}, |11\rangle = \begin{pmatrix}0\\0\\1\\0\end{pmatrix}, |10\rangle = \begin{pmatrix}0\\0\\0\\1\end{pmatrix},$$

We can interpret this notation as telling us the possible combinations of the states of each qubit in the system. For example, $|ab\rangle$ tells us that the first qubit has state a and the second, state b. Of course, the only possible states for each qubit are 1 and 0, so our basis vectors are just all the combinations of those two states.

As another alternative, we can write the state of a multiple-qubit system by directly using the tensor product. For example, let qubit $a = (a_1, a_2)^T$ and qubit $b = (b_1, b_2)^T$, in the basis $\{|0\rangle, |1\rangle\}$. Then we use the tensor product to obtain a matrix representing the states of the two vectors:

$$a \otimes b = \begin{pmatrix}a_1\\a_2\end{pmatrix} \otimes (b_1, b_2) = \begin{pmatrix}a_1b_1 & a_1b_2\\a_2b_1 & a_2b_2\end{pmatrix} = \begin{pmatrix}a_1b_1 & a_1b_2 & a_2b_1 & a_2b_2\end{pmatrix}^T$$

This is a very useful representation because it allows us to view transformations of quantum states as matrix multiplication (see problem 1).

3.4. **Example: Entangled State.** A system of qubits is entangled if the state of the system cannot be described using the individual states of the qubits in the system. One example of this entanglement is the system of two qubits with state

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

We will show that this state cannot be obtained by combining the states of two individual qubits (through the tensor product). For a contradiction, let's assume there exist two qubits with state $(a|0\rangle + b|1\rangle)$ and $(c|0\rangle + d|1\rangle)$ such that their combined state results in the aforementioned system state.

From the previous section, we have

$$(a|0\rangle + b|1\rangle) \otimes (c|0\rangle + d|1\rangle)$$

$$= ac|00\rangle + ad|01\rangle + bc|10\rangle + bd|11\rangle$$

However, it is impossible for the below to hold true

$$ac = \frac{1}{\sqrt{2}}$$

$$ad = 0$$

$$bc = 0$$

$$bd = \frac{1}{\sqrt{2}}$$

so the state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ is entangled.

3.5. **Example: Entangled State Measurement.** Another property of an entangled system is that the measurement of one qubit affects the measurement of the others. Let us work through an example of measurement of a 2-qubit system. Suppose the state of our system is

$$a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle$$

For clarity, we may rewrite this system as

$$u|0\rangle \otimes (a/u|0\rangle + b/u|1\rangle) + v|1\rangle(c/v|0\rangle + d/v|1\rangle)$$

Where $u = \sqrt{a^2 + b^2}, v = \sqrt{c^2 + d^2}$. Now we can clearly see that measuring the first bit will have a $u^2 = a^2 + b^2$ probability of returning 0. This would project the state of the system to

$$u|0\rangle \otimes (a/u|0\rangle + b/u|1\rangle) = a|00\rangle + b|01\rangle$$

This now means that the probability of measuring the second bit as 0 is a^2, and measuring it as 1 is b^2. Similarly, measuring the first bit will have a $v^2 = c^2 + d^2$ probability of returning 1. This would project the state of the system to

$$v|1\rangle(c/v|0\rangle + d/v|1\rangle) = c|10\rangle + d|11\rangle$$

Which means the the probability of measuring the second bit as 0 is c^2, and measuring it as 1 is d^2. The possible results of two consecutive measurements (when considering the second measurement, we take the first as granted) can be written as

1) First bit measured as 0 (this has an $a^2 + b^2$ chance of happening)
   (a) Second bit measured as 0 (this has a a^2 chance of happening)
   (b) Second bit measured as 1 (this has a b^2 chance of happening)
2) First bit measured as 1 (this has a $c^2 + d^2$ chance of happening
   (a) Second bit measured as 0 (this has a c^2 chance of happening)
   (b) Second bit measured as 1 (this has a d^2 chance of happening)

Hopefully, it is now easy to see how the measurement of one qubit affects the measurement of the other.

4. Quantum Gates

Quantum gates are operators which transform the state of a system of qubits. These are necessary to introduce if we want to manipulate our qubits to store data inside a potential quantum computer. One property of a quantum gate is crucial to the linear algebra perspective, and is discussed below.

4.1. **Quantum gates are unitary.** A unitary transformation is one which preserves the norm and is invertible.

As mentioned in the introduction, the natural evolution of qubits is usually modeled through unitary transformations. We would like to be able to represent both this evolution, and any other special transformation, with the same system. Therefore, it is convenient to restrict ourselves to strictly unitary gates.

Restricting ourselves to unitary gates also allows us to continue predicting the probabilities based on the components of the vector representing the state of our system. For example,

the state vector $a|0\rangle + b|1\rangle$ is purposefully normalized so we can say the state has an a^2 probability of being measured as $|0\rangle$ and a b^2 probability of being measured as $|1\rangle$. Any quantum gate/unitary transformation will preserve the norm of this vector, so it will remain normalized, and our probabilities will continue to add to one. However, if we transform this state into some state which is not normalized, it will become difficult to read off measurement probabilities in the way described above.

Finally, unitary transformations are given to be invertible. This is an advantage of quantum computing over classical computing, as we may backtrack to previous states using inverse transformations.

Some core properties of unitary transformations are

- $U^* = U^{-1}$ (which is also unitary)
- The product of unitary matrices is also a unitary matrix (therefore we can combine quantum gates freely!)

4.2. **Example: check if a particular transformation is a possible quantum gate.** A transformation A is unitary if $A^*A = I$. Therefore, given an arbitrary transformation, checking this property is sufficient to determine whether or not it is a valid possible quantum gate. As an example, let

$$A = \begin{pmatrix} 3/5 & -4/5 \\ 4/5 & 3/5 \end{pmatrix}$$

Then

$$A^* = \begin{pmatrix} 3/5 & 4/5 \\ -4/5 & 3/5 \end{pmatrix}$$

We have

$$A^*A = \begin{pmatrix} 9/25 + 16/25 & -12/25 + 12/25 \\ -12/25 + 12/25 & 16/25 + 9/25 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

We see $A^*A = I$, thus A is unitary and is a possible quantum transformation to implement.

4.3. **Problem 1 Statement.** Consider the matrices

$$A = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix},\ B = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix},\ C = \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix},\ D = \begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix}.$$

Find unitary matrices U_1, U_2, U_3, U_4 such that

$$U_1^{-1}AU_1 = B,\ U_2^{-1}BU_2 = C,\ U_3^{-1}CU_3 = D,\ U_4^{-1}DU_4 = A$$

4.4. **Problem 1 Solution.** First, notice that since linear (matrix) transformations must map the zero vector to itself, U_1 and U_3 are impossible to construct – as they require a mapping of the zero vector to a non-zero vector. For U_2, however, let

$$U_2 = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

Then we must solve

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

$$\begin{pmatrix} a+c & b+d \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & a+b \\ 0 & c+d \end{pmatrix}$$

Therefore we have a system of equations

$$\begin{aligned} a + c &= 0 \\ d - a &= 0 \\ c + d &= 0 \end{aligned}$$

Solving this system is trivial. The solution to is given by

$$\begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} x_2 \\ x_1 \\ -x_2 \\ x_2 \end{pmatrix}, \; x_1, x_2 \in \mathbb{C}$$

So

$$U_2 = \begin{pmatrix} x_2 & x_1 \\ -x_2 & x_2 \end{pmatrix}$$

However, this matrix must be unitary. For simplicity, let us assume this matrix is real (so $x_1, x_2 \in \mathbb{R}$). Then, using the property $A^*A = I$ of a unitary matrix, we know that U must satisfy

$$\begin{pmatrix} x_2 & x_1 \\ -x_2 & x_2 \end{pmatrix}^* \begin{pmatrix} x_2 & x_1 \\ -x_2 & x_2 \end{pmatrix} = \begin{pmatrix} x_2 & -x_2 \\ x_1 & x_2 \end{pmatrix} \begin{pmatrix} x_2 & x_1 \\ -x_2 & x_2 \end{pmatrix} = \begin{pmatrix} 2x_2^2 & x_1x_2 - x_2^2 \\ x_1x_2 - x_2^2 & x_1^2 + x_2^2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Therefore, $2x_2^2 = 1$, which implies $x_2 = \pm\frac{1}{\sqrt{2}}$. Next, $x_1x_2 - x_2^2 = 0$, which implies $x_1 = x_2 = \pm\frac{1}{\sqrt{2}}$. This also satisfies the last equation given by the matrix shown above, which is that $x_1^2 + x_2^2 = 1$. So we conclude that

$$U_2 = \begin{pmatrix} \pm\frac{1}{\sqrt{2}} & \pm\frac{1}{\sqrt{2}} \\ \mp\frac{1}{\sqrt{2}} & \pm\frac{1}{\sqrt{2}} \end{pmatrix}$$

We perform a similar process to find U_4:

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

$$\begin{pmatrix} 0 & 0 \\ a+c & b+d \end{pmatrix} = \begin{pmatrix} a+b & 0 \\ c+d & 0 \end{pmatrix}$$

Therefore we have a system of equations

$$\begin{aligned} b+d &= 0 \\ a-d &= 0 \\ a+b &= 0 \end{aligned}$$

Solving this system is trivial. The solution to is given by

$$\begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} x_2 \\ -x_2 \\ x_1 \\ x_2 \end{pmatrix}, \; x_1, x_2 \in \mathbb{C} \text{ or } \mathbb{R}$$

So

$$U_4 = \begin{pmatrix} x_2 & -x_2 \\ x_1 & x_2 \end{pmatrix}$$

However, this matrix must be unitary. For simplicity, let us assume this matrix is real (so $x_1, x_2 \in \mathbb{R}$). Then, using the property $A^*A = I$ of a unitary matrix, we know that U must satisfy

$$\begin{pmatrix} x_2 & -x_2 \\ x_1 & x_2 \end{pmatrix}^* \begin{pmatrix} x_2 & -x_2 \\ x_1 & x_2 \end{pmatrix} = \begin{pmatrix} x_2 & x_1 \\ -x_2 & x_2 \end{pmatrix} \begin{pmatrix} x_2 & -x_2 \\ x_1 & x_2 \end{pmatrix} = \begin{pmatrix} x_1^2 + x_2^2 & x_1x_2 - x_2^2 \\ x_1x_2 - x_2^2 & 2x_2^2 \end{pmatrix}$$

Therefore, $2x_2^2 = 1$, which implies $x_2 = \pm\frac{1}{\sqrt{2}}$. Next, $x_1x_2 - x_2^2 = 0$, which implies $x_1 = x_2 = \pm\frac{1}{\sqrt{2}}$. This also satisfies the last equation given by the matrix shown above, which is that $x_1^2 + x_2^2 = 1$. So we conclude that

$$U_4 = \begin{pmatrix} \pm\frac{1}{\sqrt{2}} & \mp\frac{1}{\sqrt{2}} \\ \pm\frac{1}{\sqrt{2}} & \pm\frac{1}{\sqrt{2}} \end{pmatrix}$$

Why is this problem relevant? Recall that in theory, we are able to represent the state of a system of two qubits (a_1, a_2), (b_1, b_2) as the two-by-two matrix $\begin{pmatrix} a_1b_1 & a_1b_2 \\ a_2b_1 & a_2b_2 \end{pmatrix}$. Problem 1 shows that we can easily construct valid quantum gates/unitary transformations to transform a given system state to a desired one.

(Note: in this paper, we switch freely between multiple ways of representing qubits, such as the bra/ket notation and the matrix notation above. However, the physical details and possibility of switching between these representations within a real quantum computer is

outside of the scope of this paper. We thus take the ability to use any desired method of representing qubits as granted.)

4.5. **Projection and Measurement: Transformations that Stand Out.** We want quantum gates to be unitary. However, there exist important quantum transformations in quantum computing that are not unitary. These specific transformations are measurement and projection. Measurement in particular is not invertible, as it collapses the state of a qubit to one of its basis vectors, with each basis vector having a certain probability associated with it. Projection also collapses the state of a qubit, but is not probabilistic like measurement.

Projection is especially important in computers where the qubits used are photons. In this case, we may use a polarization filter to project the qubits to a desired basis. Let $\vec{v}$ be the unit vector representing the direction of the polarization, then the photons passing through the filter would be described as a vector $\vec{v}$.

If the associated basis vector of the projection device is either $|0\rangle$ or $|1\rangle$, the projection operator is the outer product $|0\rangle\langle 0|$ and $|1\rangle\langle 1|$. So to get projection of a qubit onto $|0\rangle$, we apply $|0\rangle\langle 0|$ to $a|0\rangle + b|1\rangle$ and would get

$$|0\rangle\langle 0|(a|0\rangle + b|1\rangle) = a|0\rangle\langle 0||0\rangle + b|0\rangle\langle 0||1\rangle$$
$$= a|0\rangle * 1 + b|0\rangle * 0 = a|0\rangle$$

5. Reversible Logic Gates

To build a functional quantum computer, we need to be able to encode standard logical operators such as NOT, AND, OR, XOR, etc, using qubits. We know that any legitimate quantum transformation must be unitary. Therefore, we need to find a way to use unitary transformations to express these operators. For this, the C_{not} (controlled-not) gate is very useful. The C_{not} gate "operates on two qubits as follows: it changes the second bit if the first bit is 1, and leaves the [second] bit unchanged otherwise" (311). The C_{not} gate has representation

$$C_{not} := \begin{matrix} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{matrix} \begin{matrix} \to |00\rangle \\ \to |01\rangle \\ \to |11\rangle \\ \to |10\rangle \end{matrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

for example:

$$C_{not}|00\rangle = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = |00\rangle$$

Note that the C_{not} gate has the form of a row switching elementary matrix and its inverse is itself (row switch and then switch back). Thus, we see that $C_{not} = C_{not}^*$, so the C_{not} gate is unitary and invertible.

Using the C_{not} gate, we may construct another gate called the Toffolli gate (T). Using this gate, we can represent many useful operators, as seen in the following problem.

5.1. **Problem 2 Statement.** Consider the Toffoli gate

$$T : \{0,1\}^3 \to \{0,1\}^3,\ T(a,b,c) := (a,b,(a \cdot b) \otimes c)$$

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

where $\overline{a}$ is the NOT operation, $+$ is the OR operation, $\cdot$ is the AND operation and $\otimes$ is the XOR operation. Express NOT(a), AND(a,b), and OR(a,b) exclusively in terms of the Toffoli gate.

T acts on a system of 3 qubits, which has $2^3 = 8$ possible states:

$$\{|000\rangle, |001\rangle, |010\rangle, |011\rangle\, |100\rangle, |101\rangle, |110\rangle, |111\rangle\}$$

$$= \left\{ \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \right\}$$

Note that the state of qubits a and b would always stay the same while the state of c reflects the result (which would hypothetically have to be measured).

5.2. **Solution to Problem 2.** To express NOT(a), we simply set $b, c = 1$. Then NOT(a) will be given by the third component of $T(a,b,c) = T(a,1,1)$. We can see this by setting a to one or zero. If a is zero, then $(a \cdot b) \otimes c = (0 \cdot 1) \otimes 1 = 0 \otimes 1 = 1$, which is precisely NOT(0). Similarly, if a is one, then $(a \cdot b) \otimes c = (1 \cdot 1) \otimes 1 = 1 \otimes 1 = 0$, which is what we expect from NOT(1).

Let us also view this solution using our established notation. We express the input to the Toffoli gate in ket notation as $|a11\rangle$, with a being the input. If a is $|0\rangle$, then our input is $|011\rangle = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}^T$, so T$|011\rangle = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}^T = |011\rangle$. Reading off the third bit gives us $|1\rangle$, which is the opposite of $a = |0\rangle$. Similarly, if $a = |1\rangle$, we get $T|111\rangle = |110\rangle$, with the third bit c reflecting the opposite of a.

To express AND(a,b), first let us note that for any bit x, XOR$(x,0) = x$. Therefore, we simply set $c = 0$, so AND(a,b) will be given by the third component of $T(a,b,c) = T(a,b,0)$.

In ket notation, we have $T|ab0\rangle$ for the input to the AND gate. Below are the possible values of a and b, and their results after being passed through the AND gate.

$$\begin{aligned} |000\rangle &\rightarrow |000\rangle \\ |010\rangle &\rightarrow |010\rangle \\ |100\rangle &\rightarrow |100\rangle \\ |110\rangle &\rightarrow |111\rangle \end{aligned}$$

We see the desired results are given with this choice of c – the toffoli gate returns the value 1 only when the first two inputs are also 1.

To express OR(a, b), we apply the Toffoli gate multiple times. First, we use it to find NOT(a) and NOT(b). Then, we use the gate again to find AND(NOT(a), NOT(b)). We use the gate a final time to find NOT(AND(NOT(a), NOT(b))), which will give us the result of OR(a, b).

The only four possible combinations of $|ab\rangle$ are $|00\rangle, |01\rangle, |10\rangle$, and $|11\rangle$. So, after the first two applications (used to negate a and b), the possible results are

$$\begin{aligned} |00\rangle &\rightarrow |11\rangle \\ |01\rangle &\rightarrow |10\rangle \\ |10\rangle &\rightarrow |01\rangle \\ |11\rangle &\rightarrow |00\rangle \end{aligned}$$

Applying AND gate (to get AND(NOT(a), NOT(b))) we get

$$\begin{aligned} |110\rangle &\rightarrow |111\rangle \\ |100\rangle &\rightarrow |100\rangle \\ |010\rangle &\rightarrow |010\rangle \\ |000\rangle &\rightarrow |000\rangle \end{aligned}$$

Recall that the result of each application of the Toffoli gate is given by the third bit of the system. So, the possible values of AND(NOT(a), NOT(b)) are be $|1\rangle, |0\rangle, |0\rangle$, and $|0\rangle$. Then, applying NOT gate on c we would get $|0\rangle, |1\rangle, |1\rangle$, and $|1\rangle$, which are the results we want to have:

$$\begin{aligned} OR(|00\rangle) &= |0\rangle \\ OR(|01\rangle) &= |1\rangle \\ OR(|10\rangle) &= |1\rangle \\ OR(|11\rangle) &= |1\rangle \end{aligned}$$

6. Conclusion

This paper has covered a few basic properties of quantum computing, and especially, how the states of qubits can be represented using linear algebra. We started by modeling the states of individual qubits as vectors in bra/ket notation, $a|0\rangle + b|1\rangle$. We explained how to model states of systems of multiple qubits by combining individual qubits through the tensor product. Because of the tensor product, qubits may become entangled, which leads to their measurements affecting each other. Furthermore, we presented an example of a few possible quantum gates and their properties. Finally, we covered some very important logic gates which have direct applications to constructing a quantum computer, such as the NOT, AND, and OR logic gates.

7. Further Research

While this paper aims to provide basic examples of the most important concepts of quantum computing, it is based on our "skimming of the surface" of a topic that goes much deeper. To fully understand how a quantum computer works, more research could be conducted on the following questions:

- How are quantum gates, measurement, and transformations implemented physically?
- Can measurement of a quantum state be represented as a projection onto a subspace?
- Why are qubit state modeled by a unit vector in $\mathbb{C}^2$? (as opposed to just $\mathbb{R}^2$ or other vector spaces)
- How can entangled states be used to our advantage? (Keywords for research in this area are **quantum teleportation** and **dense coding**)

8. References and Resources

1) Problems in Quantum Computing
2) The Tensor Product, Demystified
3) An Introduction to Quantum Computing for Non-Physicists
4) What is a qubit
5) Linear Algebra and Quantum Mechanics
6) Quantum Computing Expert Explains One Concept in 5 Levels of Difficulty
7) If all quantum gates must be unitary, what about measurement?
8) Two Qubit Entanglement and Bell Inequalities
9) Classical and Quantum Gates
10) Quantum Computing for Computer Scientists