In [1]:

```
%run utils.py
```

COMS 4281 - Intro to Quantum Computing

Problem Set 1, Quantum Info Basics

Due: September 23, 11:59pm

Collaboration is allowed and encouraged (teams of at most 3). Please read the syllabus carefully for the guidlines regarding collaboration. In particular, everyone must write their own solutions in their own words.

Name: Chandan Suri

UNI: CS4090

Collaborated with: AS6413

Problem 1: Bra-Ket Notation Basics

Define $|\psi\rangle = \alpha |0\rangle + \beta |+\rangle$.

1. Write out the state $|\psi\rangle$ as linear combinations of the standard basis states $|0\rangle, |1\rangle$. In order for $|\psi\rangle$ to be a valid quantum state (i.e. be a unit vector), what are the conditions on α, β?
2. Since $\{|+\rangle, |-\rangle\}$ also forms a basis of $\mathbb{C}^2$ (sometimes called the *diagonal basis*), write $|\psi\rangle$ as a linear combinations of $|+\rangle$ and $|-\rangle$.
3. What is the probability of obtaining the $|0\rangle$ and $|1\rangle$ states when measuring $H|\psi\rangle$?

Solution

1.

Since,

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

Using, $|+\rangle = \left(\frac{1}{\sqrt{2}}\right)(|0\rangle + |1\rangle)$ with the equation above and replacing the $|+\rangle$ gives:

$$|\psi\rangle = \alpha |0\rangle + \beta . \left(\frac{1}{\sqrt{2}}\right) . (|0\rangle + |1\rangle) \tag{2}$$

Rearranging the terms gives:

$$|\psi\rangle = \left(\alpha + \frac{\beta}{\sqrt{2}}\right). |0\rangle + \frac{\beta}{\sqrt{2}}. |1\rangle \tag{3}$$

As we can see above that the above equation is a linear combination of the standard basis states $|0\rangle$ and $|1\rangle$.

Also, since the coefficients above with $|0\rangle$ and $|1\rangle$, when squared,are basically probabilities for those two possible states happening, the sum of the coefficients above should be equal to 1.

Therefore,

$$\left|\alpha + \frac{\beta}{\sqrt{2}}\right|^2 + \left|\frac{\beta}{\sqrt{2}}\right|^2 = 1 \tag{4}$$

Solving the above equation,

$$\alpha^2 + \frac{\beta^2}{2} + \frac{2\alpha\beta}{\sqrt{2}} + \frac{\beta^2}{2} = 1 \tag{5}$$

Rearranging the terms above gives us the following:

$$\alpha^2 + \beta^2 + \frac{2.\sqrt{2}\alpha\beta}{\sqrt{2}\sqrt{2}} = 1 \tag{6}$$

Reducing the coefficients for the third term in the equation above gives us:

$$\alpha^2 + \beta^2 + \sqrt{2}\alpha\beta = 1 \tag{7}$$

The above is the condition on α and β that those terms should satisfy!

2.

Since $\{|+\rangle, |-\rangle\}$ is the diagonal basis

and, $|\psi\rangle = \alpha|0\rangle + \beta|+\rangle$.

We can surely write the equation for $|\psi\rangle$ using the diagonal basis.

Also, we know that,

$$|+\rangle = \frac{1}{\sqrt{2}}. (|0\rangle + |1\rangle) \text{ and } |-\rangle = \frac{1}{\sqrt{2}}. (|0\rangle - |1\rangle) \tag{8}$$

To get the equation for $|\psi\rangle$ in the form of diagonal basis, we already have $|+\rangle$, but we also have $|0\rangle$ and no $|-\rangle$.

In order to replace the $|0\rangle$, we will have to somehow combine the equations for $|+\rangle$ and $|-\rangle$ to get the same.

So, let's try and add the equations for $|+\rangle$ and $|-\rangle$:

$$|+\rangle + |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) + \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \tag{9}$$

Let's try and combine the qubits 0 and 1, the $|1\rangle$ will get cancelled out and we will get the following:

$$|+\rangle + |-\rangle = \frac{2}{\sqrt{2}}|0\rangle \tag{10}$$

Multiplying and dividing the RHS by $\sqrt{2}$ and then rearranging the coefficient gives us:

$$\frac{1}{\sqrt{2}}(|+\rangle + |-\rangle) = |0\rangle \tag{11}$$

Thus, using the equation for $|0\rangle$ above, we can replace the same in the equation for $|\psi\rangle$ as,

$$|\psi\rangle = \frac{\alpha}{\sqrt{2}}(|+\rangle + |-\rangle) + \beta\,|+\rangle \tag{12}$$

Rearranging the terms gives us the final relationship:

$$|\psi\rangle = \left(\frac{\alpha}{\sqrt{2}} + \beta\right)|+\rangle + \frac{\alpha}{\sqrt{2}}|-\rangle \tag{13}$$

3.

Since,

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|+\rangle \tag{14}$$

$H\,|\psi\rangle$ can be written as follows:

$$H\,|\psi\rangle = H.\,(\alpha\,|0\rangle + \beta\,|+\rangle) \tag{15}$$

Rearranging the terms and taking the coefficients out of the parenthesis gives us the following:

$$H\,|\psi\rangle = \alpha H\,|0\rangle + \beta H\,|+\rangle \tag{16}$$

Since, we know the effect of the Hamadard matrix on the states is as:

$$H\,|0\rangle = |+\rangle \text{ and } H\,|+\rangle = |0\rangle \tag{17}$$

So, using the equalities above the equation for $H\,|\psi\rangle$ becomes:

$$H\,|\psi\rangle = \alpha\,|+\rangle + \beta\,|0\rangle \tag{18}$$

Opening up the $|+\rangle$ gives us:

$$H\,|\psi\rangle = \frac{\alpha}{\sqrt{2}}.\,(|0\rangle + |1\rangle) + \beta\,|0\rangle \tag{19}$$

Rearranging the terms gives us:

$$H|\psi\rangle = \left(\frac{\alpha}{\sqrt{2}} + \beta\right)|0\rangle + \frac{\alpha}{\sqrt{2}}|1\rangle \tag{20}$$

So, the probabilities are as follows:

Probability of measuring $|0\rangle$ is:

$$Pr(Measuring\,|0\rangle) = \left(\frac{\alpha}{\sqrt{2}} + \beta\right)^2 = \frac{\alpha^2}{2} + \beta^2 + \sqrt{2}\alpha\beta \tag{21}$$

Probability of measuring $|1\rangle$ is:

$$Pr(Measuring\,|1\rangle) = \left(\frac{\alpha}{\sqrt{2}}\right)^2 = \frac{\alpha^2}{2} \tag{22}$$

Problem 2: Outer Products and Projections

Recall that $\langle\psi|\theta\rangle$ denotes a scalar, because is an inner product between two vectors (i.e., you have a row vector followed by a column vector). Now consider $|\psi\rangle\langle\theta|$, which denotes the outer product between the two vectors (i.e., a column vector followed by a row vector). The outer product of two vectors is a matrix.

1. Show that $|0\rangle\langle 0| = \frac{1}{2}(I+Z)$ and $|+\rangle\langle +| = \frac{1}{2}(I+X)$, where $Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$ and $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$.
2. Let $\{|0\rangle, |1\rangle, \ldots, |n-1\rangle\}$ denote the standard basis for $\mathbb{C}^n$. Show that $I = \sum_{x=0}^{n-1} |x\rangle\langle x|$. This simple identity known as the `completeness relation` can be very useful.
3. Use the completeness relation to show that $|\psi\rangle = \sum_x \langle x|\psi\rangle\, |x\rangle$ for any vector $|\psi\rangle$. Thus the amplitudes of $|\psi\rangle$, in the standard basis, can be written as inner products of $|\psi\rangle$ with the standard basis vectors.

Solution

1.

First, let's prove the equation $|0\rangle\langle 0| = \frac{1}{2}(I+Z)$ as:

Let's take the LHS of the equation above:

LHS:

$$|0\rangle\langle 0| = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \end{pmatrix}$$

Solving the same (matrix multiplication), gives us:

$$|0\rangle\langle 0| = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$$

Now, let's take the RHS of the equation:

RHS:

$$\frac{1}{2}(I + Z) = \frac{1}{2}\left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}\right)$$

Adding the matrices element-wise gives us the following:

$$\frac{1}{2}(I + Z) = \frac{1}{2}\begin{pmatrix} 2 & 0 \\ 0 & 0 \end{pmatrix}$$

This, in turn is equal to:

$$\frac{1}{2}(I + Z) = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$$

So, since $LHS = RHS$ in the above equation. Hence, $|0\rangle\langle 0| = \frac{1}{2}(I + Z)$.

Now, Let's prove the equation $|+\rangle\langle +| = \frac{1}{2}(I + X)$ as:

Let's take the LHS first:

LHS:

Since,

$$|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = \frac{1}{\sqrt{2}}\left(\begin{pmatrix} 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix}\right) = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

Therefore,

$$|+\rangle\langle +| = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix} \cdot \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \end{pmatrix}$$

Combining coefficients and multiplying matrices is done as follows:

$$|+\rangle\langle +| = \left(\frac{1}{\sqrt{2}}\right)^2 \begin{pmatrix} 1 \\ 1 \end{pmatrix} \begin{pmatrix} 1 & 1 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

Now, let's solve for RHS:

RHS:

$$\frac{1}{2}(I+X) = \frac{1}{2}\left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\right)$$

Adding the terms of the matrices element-wise gives us:

$$\frac{1}{2}(I+X) = \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

So, since $LHS = RHS$ in the above equation. Hence, $|+\rangle\langle+| = \frac{1}{2}(I+X)$.

Thus, both the equations have been proven.

2.

We need to show that:

$$I = \sum_{x=0}^{n-1} |x\rangle\langle x|$$

Let's build our intuition for the proof,

When n = 2, the equation becomes:

$$I = \sum_{x=0}^{1} |x\rangle\langle x| = |0\rangle\langle 0| + |1\rangle\langle 1|$$

Since,

$$|0\rangle\langle 0| = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$$

and,

$$|1\rangle\langle 1| = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$$

The above equation becomes:

$$I = \sum_{x=0}^{1} |x\rangle\langle x| = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

As we can see, we actually get the identity matrix above.

As, $|x\rangle\langle x|$ means that the xth row and xth column entry in the matrix will be 1 while all other will be zero.

Thus, all these 1s will lie on the main diagonal and as it's for all the rows and columns starting at (0, 0) and ending at (n-1, n-1), we will cover all the main diagonal entries with 1s which is basically the identity matrix. It's as shown below:

$$I=\sum_{x=0}^{n-1}|x\rangle\langle x| = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix}_{n\times n} + \begin{pmatrix} 0 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix}_{n\times n} + \cdots + \begin{pmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}$$

This becomes an identity matrix as each entry when the summation sign is opened becomes a single 1 in the xth row and column.

Thus, the completeness relation basically tells us that $\sum_{x=0}^{n-1}|x\rangle\langle x|$ is an identity matrix which is perfectly true.

3.

To prove that $|\psi\rangle = \sum_x \langle x|\psi\rangle\,|x\rangle$ for any vector $|\psi\rangle$.

Let's start with $|\psi\rangle$:

Multiplying by the identity matrix should give us the same result, so we can do the following:

$$|\psi\rangle = I\,|\psi\rangle$$

Next, using the completeness relation $I=\sum_{x=0}^{n-1}|x\rangle\langle x|$ above:

$$|\psi\rangle = I\,|\psi\rangle = \left(\sum_{x=0}^{n-1}|x\rangle\langle x|\right)|\psi\rangle$$

Taking the $|\psi\rangle$ inside the summation will give us:

$$|\psi\rangle = I\,|\psi\rangle = \sum_{x=0}^{n-1}\left(|x\rangle\langle x|\,|\psi\rangle\right)$$

As the matrix-vector product is associative, we can do the following:

$$|\psi\rangle = I\,|\psi\rangle = \sum_{x=0}^{n-1}|x\rangle\left(\langle x|\,|\psi\rangle\right)$$

Since, the quantity inside the brackets in the latter part is a scalar, we can move it in the front as follows:

$$|\psi\rangle = I\,|\psi\rangle = \sum_{x=0}^{n-1}\left(\langle x|\psi\rangle\right)|x\rangle$$

Therefore,

$$|\psi\rangle = I\,|\psi\rangle = \sum_{x=0}^{n-1}\langle x|\psi\rangle\,|x\rangle$$

Problem 3: Composite Quantum Systems

Here are some questions to help you get used to the tensor product.

Let $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{m \times m}$. Recall that the tensor product $A \otimes B$ can be given by:

$$A \otimes B = \begin{bmatrix} A_{11}B & A_{12}B & \cdots & A_{1n}B \\ A_{11}B & A_{12}B & \cdots & A_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1}B & A_{n2}B & \cdots & A_{nn}B \end{bmatrix}$$

is an $nm \times nm$ dimensional matrix. This is also known as the *Kronecker product* which is a way to represent the tensor product with respect to a given basis.

1. Give the explicit matrix representations for the matrices $I \otimes H$, $H \otimes I$ and $H \otimes H$.
2. Compute $(I \otimes H) |0, 1\rangle$, $(H \otimes I) |0, 1\rangle$ and $(H \otimes H) |0, 1\rangle$, expressing the results in the standard basis (i.e. $\sum_{i,j \in \{0,1\}} c_{ij} |i\rangle |j\rangle$ and some constants $c_{ij} \in \mathbb{C}$)

Solution

1.

Solving for $I \otimes H$ as follows:

$$I \otimes H = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Using the tensor product notation above we will multiply H by all the elements of I one by one and we take the coefficient out as well. After this, we get the following:

$$I \otimes H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} & 0 \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \\ 0 \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} & 1 \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \end{pmatrix}$$

Now, multiplying the elements of I with the H matrices shows above gives us:

$$I \otimes H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{pmatrix}$$

Solving for $H \otimes I$ as follows:

$$H \otimes I = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Using the tensor product notation above we will multiply I by all the elements of H one by one and we take the coefficient out as well. After this, we get the following:

$$H \otimes I = \frac{1}{\sqrt{2}} \begin{pmatrix} 1\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} & 1\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\ 1\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} & -1\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \end{pmatrix}$$

Now, multiplying the elements of H with the I matrices shows above gives us:

$$H \otimes I = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{pmatrix}$$

Solving for $H \otimes H$ as follows:

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

Rearranging the coefficients as that won't affect the tensor product,

$$H \otimes H = \frac{1}{\sqrt{2}} \cdot \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Using the tensor product notation above we will multiply H by all the elements of H one by one and we take the coefficient out as well. After this, we get the following:

$$H \otimes H = \frac{1}{2} \begin{pmatrix} 1\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} & 1\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \\ 1\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} & -1\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \end{pmatrix}$$

Now, multiplying the elements of H with the H matrices shows above gives us:

$$H \otimes H = \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}$$

2.

$(I \otimes H)\,|0,1\rangle$ can be re-written as $(I \otimes H)\,|01\rangle$.

Now, we have already calculated $(I \otimes H)$ above, so, we just need to multiply the same with the $|01\rangle$ which can be done as follows:

$$(I \otimes H)|01\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

Multiplying the two matrices above gives us the following:

$$(I \otimes H)|01\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \\ 0 \\ 0 \end{pmatrix}$$

This can be re-written as:

$$(I \otimes H)|01\rangle = \frac{1}{\sqrt{2}} \left[\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} - \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \right]$$

The first column matrix above is actually $|00\rangle$ and the second one is $|01\rangle$.

So, it can be written as the following:

$$(I \otimes H)|01\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |01\rangle)$$

Moving forward with the second one:

$(H \otimes I)|0,1\rangle$ can be re-written as $(H \otimes I)|01\rangle$.

Now, we have already calculated $(H \otimes I)$ above, so, we just need to multiply the same with the $|01\rangle$ which can be done as follows:

$$(H \otimes I)|01\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

Multiplying the two matrices above gives us the following:

$$(H \otimes I)|01\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}$$

This can be re-written as:

$$(H \otimes I) |01\rangle = \frac{1}{\sqrt{2}} \left[\begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \right]$$

The first column matrix above is actually $|01\rangle$ and the second one is $|11\rangle$.

So, it can be written as the following:

$$(I \otimes H) |01\rangle = \frac{1}{\sqrt{2}} (|01\rangle + |11\rangle)$$

Moving forward with the last one:

$(H \otimes H) |0, 1\rangle$ can be re-written as $(H \otimes H) |01\rangle$.

Now, we have already calculated $(H \otimes H)$ above, so, we just need to multiply the same with the $|01\rangle$ which can be done as follows:

$$(H \otimes H) |01\rangle = \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

Multiplying the two matrices above gives us the following:

$$(H \otimes H) |01\rangle = \frac{1}{2} \begin{pmatrix} 1 \\ -1 \\ 1 \\ -1 \end{pmatrix}$$

This can be re-written as:

$$(H \otimes H) |01\rangle = \frac{1}{2} \left[\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} - \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \right]$$

The first column matrix above is actually $|00\rangle$, second one is $|01\rangle$, third one is $|10\rangle$ and the fourth one is $|11\rangle$.

So, it can be written as the following:

$$(H \otimes H) |01\rangle = \frac{1}{2} (|00\rangle - |01\rangle + |10\rangle - |11\rangle)$$

Problem 4: Quantum Entanglement

For each of the following quantum states, say whether they are entangled or unentangled across the specified bipartition, and justify your answer.

1. $|\Phi\rangle = \frac{1}{\sqrt{2}}\Big(|00\rangle + |11\rangle\Big) = \begin{pmatrix} 1/\sqrt{2} \\ 0 \\ 0 \\ 1/\sqrt{2} \end{pmatrix}$ is a two-qubit state in $\mathbb{C}^2 \otimes \mathbb{C}^2$, called the `EPR pair`. Are the two qubits entangled?
2. $|\psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle$ is an n-qubit state where $|x\rangle = |x_1\rangle \otimes |x_2\rangle \otimes \cdots \otimes |x_n\rangle$ is a basis state for $(\mathbb{C}^2)^{\otimes n}$. Let's say we divide the n qubits into the first j and the last $n-j$ qubits. Is the state entangled across the bipartition $(\mathbb{C}^2)^{\otimes j} \otimes (\mathbb{C}^2)^{\otimes (n-j)}$?
3. Let $|\Phi\rangle$ be the EPR pair. Let $H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ denote the Hadamard gate. Is the state $(I \otimes H)\,|\Phi\rangle$ entangled?
4. Is the state $CNOT\,|\Phi\rangle$ entangled?

Solution

1.

To see whether 2 qubits in the EPR pair are entangled, let's assume that the two qubits are not entangled which would mean that we can show that as a tensor product of two quantum states,

let's say those are as follows:

$$|c\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \text{ and } |d\rangle = \gamma\,|0\rangle + \delta\,|1\rangle$$

Let's say we could show the EPR pair as the tensor product of $|c\rangle$ and $|d\rangle$, let's solve the tensor product now,

$$|c\rangle \otimes |d\rangle = (\alpha\,|0\rangle + \beta\,|1\rangle) \otimes (\gamma\,|0\rangle + \delta\,|1\rangle)$$

Let's solve this further,

$$|c\rangle \otimes |d\rangle = \left[\alpha\begin{pmatrix} 1 \\ 0 \end{pmatrix} + \beta\begin{pmatrix} 0 \\ 1 \end{pmatrix}\right] \otimes \left[\gamma\begin{pmatrix} 1 \\ 0 \end{pmatrix} + \delta\begin{pmatrix} 0 \\ 1 \end{pmatrix}\right] = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \otimes \begin{pmatrix} \gamma \\ \delta \end{pmatrix} = \begin{pmatrix} \alpha\gamma \\ \alpha\delta \\ \beta\gamma \\ \beta\delta \end{pmatrix}$$

Now, if the product above is equivalent to the $|\Phi\rangle$, then the following equations should hold which we get by equating each of the elements,

$$\alpha\gamma = \frac{1}{\sqrt{2}}$$

$$\beta\delta = \frac{1}{\sqrt{2}}$$

$$\alpha\delta = 0$$

$$\beta\gamma = 0$$

For all the above equations to be true, either of the 4 combinations shown below should hold which are as follows with the reason why they can or cannot hold:

$\alpha = 0, \beta = 0$	Not Possible since the first 2 equations above won't hold!
$\alpha = 0, \gamma = 0$	Not Possible since the first equation above won't hold!
$\delta = 0, \beta = 0$	Not Possible since the second equation above won't hold!
$\delta = 0, \gamma = 0$	Not Possible since the first 2 equations above won't hold!

Therefore, there's no possibility of $|\Phi\rangle$ being a tensor product of two others. Since, that's not possible, the two quibits in the EPR pair are entangled.

2.

We are given that $|\psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle$ is a n-qubit state where $|x\rangle = |x_1\rangle \otimes |x_2\rangle \otimes \cdots \otimes |x_n\rangle$ is the basis state.

Now, if we divide the state such that we have 2 partitions with one of them having j qubits and the other one having $n-j$ qubits.

We can re-write that having the form:

$$|z\rangle = |z_1\rangle \otimes |z_2\rangle$$

where z_1 is the term corresponding to j qubits and z_2 is the term corresponding to $n-j$ qubits.

As there are 2^n terms in total, for the 1st term above, there are 2^j terms from which we can factor out z_2 which will give us the following:

$$\left(\sum_{x\in\{0,1\}^j} |x\rangle\right) \otimes z_2$$

Now, if we compare the same with the equation for $|\psi\rangle$, we can see that we can re-write the same as (where p is the combination of $n-j$ qubits):

$$\frac{1}{\sqrt{2^n}}\left(\sum_{x\in\{0,1\}^{n-j}}\left(\sum_{x\in\{0,1\}^j} |x\rangle\right) \otimes |p\rangle\right)$$

This can be re-written as:

$$\frac{1}{\sqrt{2^n}}\left(\sum_{x\in\{0,1\}^j}|x\rangle\right)\otimes\left(\sum_{x\in\{0,1\}^{n-j}}|p\rangle\right)$$

As we can see above, j is arbitrary, we can partition the n qubits any way we want and it will be a tensor product of the form shown above. This would mean that the bi-partition formed will always be seperable. Thus, in turn, means that the state across the bipartition $(\mathbb{C}^2)^{\otimes j}\otimes(\mathbb{C}^2)^{\otimes(n-j)}$ is not entangled.

3.

From problem 3 and 1st part, we know that:

$$I\otimes H=\frac{1}{\sqrt{2}}\begin{pmatrix}1&1&0&0\\1&-1&0&0\\0&0&1&1\\0&0&1&-1\end{pmatrix}$$

So, $I\otimes H\,|\Phi\rangle$ can be computed as follows:

$$I\otimes H\,|\Phi\rangle=\frac{1}{\sqrt{2}}\frac{1}{\sqrt{2}}\begin{pmatrix}1&1&0&0\\1&-1&0&0\\0&0&1&1\\0&0&1&-1\end{pmatrix}\begin{pmatrix}1\\0\\0\\1\end{pmatrix}$$

Solving this further gives us the following:

$$I\otimes H\,|\Phi\rangle=\frac{1}{2}\begin{pmatrix}1\\1\\1\\-1\end{pmatrix}$$

This can also be re-written as:

$$I\otimes H\,|\Phi\rangle=\frac{1}{2}(|00\rangle+|01\rangle+|10\rangle-|11\rangle)$$

Just by looking at the above equation we can see that there's not way to factor out the qubits such that we can show that as a tensor product.

However, let's look at this in the formal way!

Let's assume that the equation above can be shown as a tensor product of two states as shown below:

$$|c\rangle=\alpha\,|0\rangle+\beta\,|1\rangle\ \text{ and }\ |d\rangle=\gamma\,|0\rangle+\delta\,|1\rangle$$

$$|c\rangle\otimes|d\rangle=(\alpha\,|0\rangle+\beta\,|1\rangle)\otimes(\gamma\,|0\rangle+\delta\,|1\rangle)$$

$$|c\rangle \otimes |d\rangle = \left[\alpha\begin{pmatrix}1\\0\end{pmatrix} + \beta\begin{pmatrix}0\\1\end{pmatrix}\right] \otimes \left[\gamma\begin{pmatrix}1\\0\end{pmatrix} + \delta\begin{pmatrix}0\\1\end{pmatrix}\right] = \begin{pmatrix}\alpha\\\beta\end{pmatrix} \otimes \begin{pmatrix}\gamma\\\delta\end{pmatrix} = \begin{pmatrix}\alpha\gamma\\\alpha\delta\\\beta\gamma\\\beta\delta\end{pmatrix}$$

This would mean that the following equalities should hold:

$$\alpha\gamma = \frac{1}{2}$$

$$\beta\delta = -\frac{1}{2}$$

$$\alpha\delta = \frac{1}{2}$$

$$\beta\gamma = \frac{1}{2}$$

For the 2nd equality above to be true, either β or δ should be negative (but not both). Let's consider the two cases below:

1. If β is negative, for $\beta\gamma$ to be positive, γ should be negative. But, if γ is negative, then for $\alpha\gamma$ to be positive, α should be negative. In turn, if α is negative, then δ should also be negative in order for $\alpha\delta$ to be positive. As both β and δ shouldn't be positive, then, there's no way this case is valid where β is negative but δ is positive.

2. If δ is negative, for $\alpha\delta$ to be positive, α should be negative. But, if α is negative, then for $\alpha\gamma$ to be positive, γ should be negative. In turn, if γ is negative, then β should also be negative in order for $\beta\gamma$ to be positive. As both β and δ shouldn't be positive, then, there's no way this case is valid where δ is negative but β is positive.

As both cases are not possible, that means we cannot represent the quantum state $I \otimes H\,|\Phi\rangle$ as a tensor product of two states. Thus, the state is not separable and therefore, is entangled.

4.

To find whether $CNOT\,|\Phi\rangle$ is entangled or not, let's try and solve the same as follows:

$$CNOT\,|\Phi\rangle = CNOT\frac{1}{\sqrt{2}}\begin{pmatrix}1\\0\\0\\1\end{pmatrix} = \frac{1}{\sqrt{2}}CNOT\begin{pmatrix}1\\0\\0\\1\end{pmatrix}$$

This can be written as:

$$CNOT\,|\Phi\rangle = \frac{1}{\sqrt{2}}CNOT(|00\rangle + |11\rangle)$$

Applying CNOT, according to the 1st qubit, the second would change which would make the same as follows:

$$CNOT\,|\Phi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle)$$

This, can be written as the tensor product with $|0\rangle$ as follows:

$$CNOT\,|\Phi\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle = (\frac{|0\rangle + |1\rangle}{\sqrt{2}}) \otimes |0\rangle$$

As we can see above, we could show the state as a tensor product of two others, thus, these are separable!

Thus, $CNOT\,|\Phi\rangle$ is not entangled.

Problem 5: Implement a Quantum Circuit

In this problem, you will implement a simple quantum circuit that constructs the $|\psi\rangle = \frac{1}{\sqrt{2}}(|000\rangle - |111\rangle)$ from the all zeroes state. In other words, you will find a circuit C such that

$$C\,|000\rangle = |\psi\rangle$$

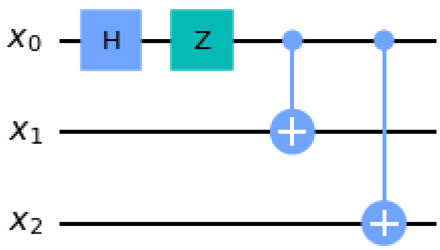
In this problem, you will use the Qiskit library to implement, visualize, and analyze the circuit C.

1. Design a circuit C to prepare the state $|\psi\rangle$, and write the corresponding Qiskit code between the "BEGIN CODE" and "END CODE" delineations below. You may use any of the gates we have learned in class. We've already created the circuit object, you just need to specify what gates to add.

In [2]:

```
def create_sym_state_circuit():
    qr = QuantumRegister(3, name='x')
    qc = QuantumCircuit(qr)

    # ========= BEGIN CODE =================
    qc.h(0)
    qc.z(0)
    qc.cx(0, 1)
    qc.cx(0, 2)

    # ========= END CODE =================


    return qc

qc = create_sym_state_circuit()
qc.draw(output='mpl')
```

Out[2]:



1. Consult the qiskit documentation and use its API to obtain the output state of the circuit you just created as a **vector** (i.e. a list of amplitudes).
   It doesn't matter for this problem, but keep in mind that qiskit uses the **little-endian** convention (e.g., qubits are ordered right to left; for more info see this). We provide a function *beautify*, which given input `amplitude` and `k` where `amplitude` is a list of 2^k complex numbers and k is an integer, prints the amplitude list as a quantum state. See the code below as an example.

In [3]:
```
# Example for beautify
amplitudes = [complex(-1.55345, 0),   complex(0,0),   complex(-1, 0.1),    compl
print('State vector:', beautify(amplitudes, 2))
```

```
State vector: -1.55 |00> + (-1+0.1i) |10> - 1.1i |11>
```

In the following code block, print the output of the circuit you created in Part 1 above as a vector in beautified form.

In [4]:
```
# ========= BEGIN CODE =================

# Import Aer
from qiskit import Aer

# Run the quantum circuit on a statevector simulator backend
backend = Aer.get_backend('statevector_simulator')

# Create a Quantum Program for execution
job = backend.run(qc)

result = job.result()

outputstate = result.get_statevector(qc, decimals=3)
print('State vector:', beautify(outputstate, 3))

# ========= END CODE =================
```

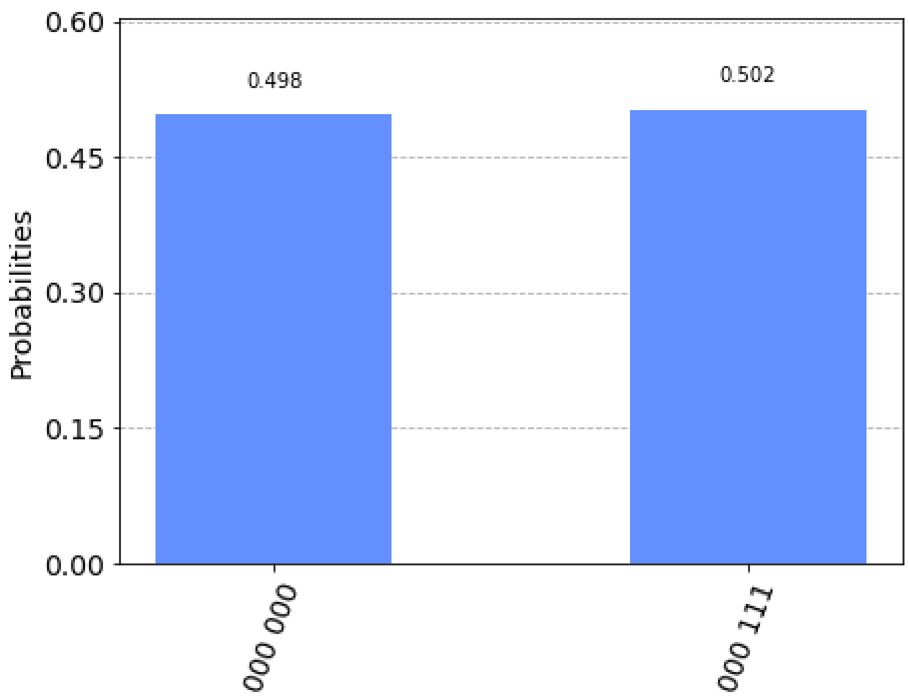
```
State vector: 0.71 |000> - 0.71 |111>
<frozen importlib._bootstrap>:219: RuntimeWarning: scipy._lib.messagestream.Mess
ageStream size changed, may indicate binary incompatibility. Expected 56 from C
header, got 64 from PyObject
```

1. Write code to measure all the qubits of the $|\psi\rangle$ state in the standard basis and visualize the measurement statistics using a histogram.

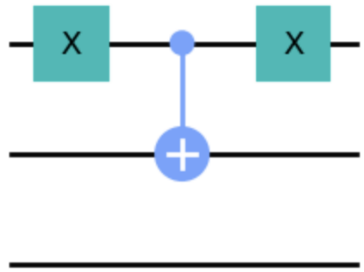
In [6]:

```
# ========= BEGIN CODE =================

# Adding classicial registers to take measurements for all the qubits.
cr = ClassicalRegister(3)
qc.add_register(cr)
qc.measure(range(3), range(3))

# Code for simulator
# Use Aer's qasm_simulator
backend_sim = Aer.get_backend('qasm_simulator')

# Execute the circuit on the qasm simulator.
# We've set the number of repeats of the circuit
# to be 2048.
job_sim = backend_sim.run(transpile(qc, backend_sim), shots=2048)

# Grab the results from the job.
result_sim = job_sim.result()

# Getting the counts
counts = result_sim.get_counts(qc)

# Visualization as a histogram
plot_histogram(counts)

# ========= END CODE =================
```

Out[6]:



1. Consider running the following circuit with $|\psi\rangle$ as input. Let $|\theta\rangle$ denote the output state. Calculate the state $|\theta\rangle$ by computing the intermediate states of the circuit, and write it out below in $\LaTeX$.



Solution

The state going into the circuit above can be written as follows:

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|000\rangle - |111\rangle)$$

Now, to compute the output state, let's compute the state of the circuit at each stage of the circuit shown above.

Step 1:

After the 1st X gate is applied,

As it's applied on the 1st qubit, it would flip it's state keeping others the same giving the intermediate output as:

$$\frac{1}{\sqrt{2}}(|100\rangle - |011\rangle)$$

Step 2:

After that the CNOT gate is applied,

As the source is the 1st qubit and target is the 2nd qubit, the 2nd qubit is going to change according to the 1st qubit giving the intermediate output as,

$$\frac{1}{\sqrt{2}}(|110\rangle - |011\rangle)$$

Step 3:

Now, when the X gate is applied again (on the 1st qubit again),

The output that we get is:

$$\frac{1}{\sqrt{2}}(|010\rangle - |111\rangle)$$

Thus, the final output can be written as:

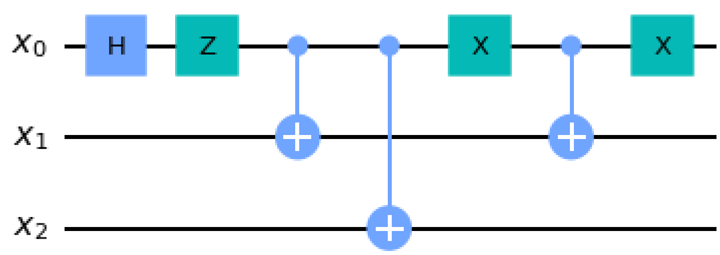
$$|\theta\rangle = \frac{1}{\sqrt{2}}(|010\rangle - |111\rangle)$$

1. Write code to implement a circuit that prepares the state $|\theta\rangle$, measure it in the standard basis and visualize the statistics using a histogram.

In [7]:

```
# Testing out the full circuit
qc = create_sym_state_circuit()
qc.x(0)
qc.cx(0, 1)
qc.x(0)
qc.draw(output='mpl')
```

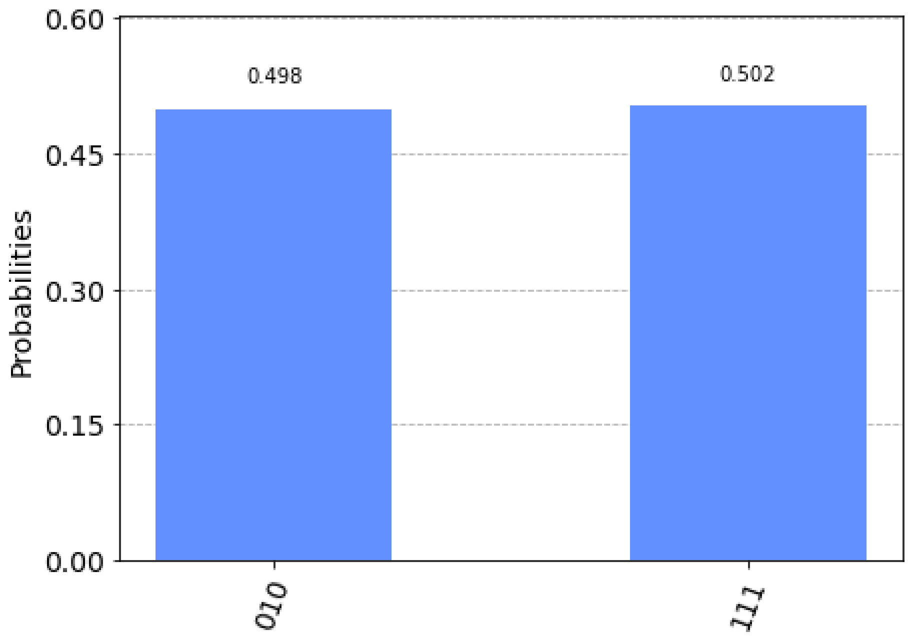
Out[7]:



In [9]:

```
# ========= BEGIN CODE =================

qc = create_sym_state_circuit()
qc.x(0)
qc.cx(0, 1)
qc.x(0)

# Adding classicial registers to take measurements for all the qubits.
cr = ClassicalRegister(3)
qc.add_register(cr)
qc.measure(range(3), range(3))

# Code for simulator
# Use Aer's qasm_simulator
backend_sim = Aer.get_backend('qasm_simulator')

# Execute the circuit on the qasm simulator.
# We've set the number of repeats of the circuit
# to be 2048.
job_sim = backend_sim.run(transpile(qc, backend_sim), shots=2048)

# Grab the results from the job.
result_sim = job_sim.result()

# Getting the counts
counts = result_sim.get_counts(qc)

# Visualization as a histogram
plot_histogram(counts)

# ========= END CODE =================
```

Out[9]:



Problem 6: A Quantum Two-bit Adder

The classical two-bit adder is an **irreversible** function that takes in four bits, (A_0, A_1) and (B_0, B_1), and outputs three bits (C_0, Q_0, Q_1) which is the binary representation of the sum of $2A_1 + A_0$ and $2B_1 + B_0$ (i.e., integers that (A_0, A_1) and (B_0, B_1) represent in binary). For example, on input $(0, 1)$ and $(1, 1)$ the two bit adder should return $(1, 0, 0)$. On input $(1, 1)$ and $(1, 1)$ it should output $(1, 1, 0)$.

You can find a circuit for an irreversible circuit for the two-bit adder here, consisting of XOR, OR, and AND gates (see Wikipedia for gate symbol reference).

In this problem you will implement a **reversible** two-bit adder in Qiskit.

1. First, let's implement reversible versions of the XOR, OR, and AND gates. Recall that every boolean function f can be converted to a reversible transformation T_f using an additional ancilla bit. Since XOR, OR, AND map 2 bits to 1 bit, the reversible functions T_{XOR}, T_{OR}, T_{AND} will map 3 bits to 3 bits. The corresponding matrices are 8×8.

In the functions below, enter the matrix representations of T_{XOR}, T_{OR}, T_{AND} below (replace the entries with the appropriate values). The row/columns are ordered as follows: $|000\rangle, |001\rangle, |010\rangle, \ldots, |111\rangle$.

Your implementations of reversible XOR, OR, and AND will be tested.

In [10]:

```
def create_Tor(qr: QuantumRegister) -> QuantumCircuit:
    assert len(qr) == 3, 'Tor gate should operate on 3 qubits.'
    qc = QuantumCircuit(qr)
    ##### FILL IN THE MATRIX BELOW FOR THE REVERSIBLE OR GATE ##########
    Tor = Operator([
        [1, 0, 0, 0, 0, 0, 0, 0],
```

```
        [0, 1, 0, 0, 0, 0, 0, 0],
        [0, 0, 0, 1, 0, 0, 0, 0],
        [0, 0, 1, 0, 0, 0, 0, 0],
        [0, 0, 0, 0, 0, 1, 0, 0],
        [0, 0, 0, 0, 1, 0, 0, 0],
        [0, 0, 0, 0, 0, 0, 0, 1],
        [0, 0, 0, 0, 0, 0, 1, 0],
    ])
    ####################################################################
    qc.unitary(Tor, [2, 1, 0], label='Tor')
    return qc

def create_Txor(qr: QuantumRegister) -> QuantumCircuit:
    assert len(qr) == 3, 'Txor gate should operate on 3 qubits.'
    qc = QuantumCircuit(qr)
    ##### FILL IN THE MATRIX BELOW FOR THE REVERSIBLE XOR GATE ##########
    Txor = Operator([
        [1, 0, 0, 0, 0, 0, 0, 0],
        [0, 1, 0, 0, 0, 0, 0, 0],
        [0, 0, 0, 1, 0, 0, 0, 0],
        [0, 0, 1, 0, 0, 0, 0, 0],
        [0, 0, 0, 0, 0, 1, 0, 0],
        [0, 0, 0, 0, 1, 0, 0, 0],
        [0, 0, 0, 0, 0, 0, 1, 0],
        [0, 0, 0, 0, 0, 0, 0, 1],
    ])
    ####################################################################
    qc.unitary(Txor, [2, 1, 0], label='Txor')
    return qc

def create_Tand(qr: QuantumRegister) -> QuantumCircuit:
    assert len(qr) == 3, 'Tand gate should operate on 3 qubits.'
    qc = QuantumCircuit(qr)
    ##### FILL IN THE MATRIX BELOW FOR THE REVERSIBLE AND GATE ##########
    Tand = Operator([
        [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],
    ])
    ####################################################################
    qc.unitary(Tand, [2, 1, 0], label='Tand')
    return qc
```

In [11]:

```
#Running test cases on your adder....
test_gates([create_Tor,create_Txor,create_Tand])
```

```
Testing gates...
OR gate: OK.
XOR gate: OK.
AND gate: OK.
```

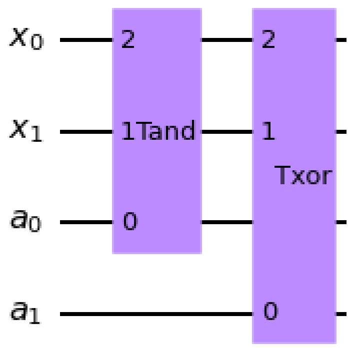
1. You can now put together reversible circuits consisting of T_{XOR}, T_{OR}, and T_{AND} by using the functions `create_Tor`, `create_Txor`, and `create_Tand`, and also a helper

function called append that allows you to append a gate G to a circuit C. The function takes in a circuit C, a function g constructs the gate G, and a list of bits that G operates on. See the code below as an example.

In [12]:

```
### EXAMPLE ONLY ###############
qx = QuantumRegister(2, name="x")
qa = QuantumRegister(2, name="a")
qc = QuantumCircuit(qx,qa)                  #add the X register and A registers
qc = append(qc, create_Tand, [0, 1, 2])     #reversible AND acting on (x0,x1) and
qc = append(qc, create_Txor, [0,1,3])       #reversible XOR acting on (x0,x1) and
qc.draw(output='mpl')
```

Out[12]:



Now, transform the irreversible circuit for the two-bit adder above to a **reversible** circuit C for the two-bit adder. More precisely, the circuit C should act on bits

- (A_0, A_1) representing the first number $A = 2A_1 + A_0$
- (B_0, B_1) representing the second number $B = 2B_1 + B_0$
- (C_0, C_1, C_2) representing the binary representation of $A + B$
- Some number of ancilla bits $(D_0, D_1, \ldots)$

The circuit C should have the behavior: for all inputs $A_0, A_1, B_0, B_1 \in \{0, 1\}$,

$$C\,|A, B, 0, 0\cdots 0\rangle = \left|\underbrace{A}_{\text{2 bits}}, \underbrace{B}_{\text{2 bits}}, \underbrace{A+B}_{\text{3 bits}}, \underbrace{S_{A,B}}_{\text{ancillas}}\right\rangle$$

where A, B are two bits and $A + B$ is represented by three bits. $S_{A,B}$ corresponds to the bits of the ancilla that depends on the inputs A, B. This data corresponds to the "scratch work" of the computation.

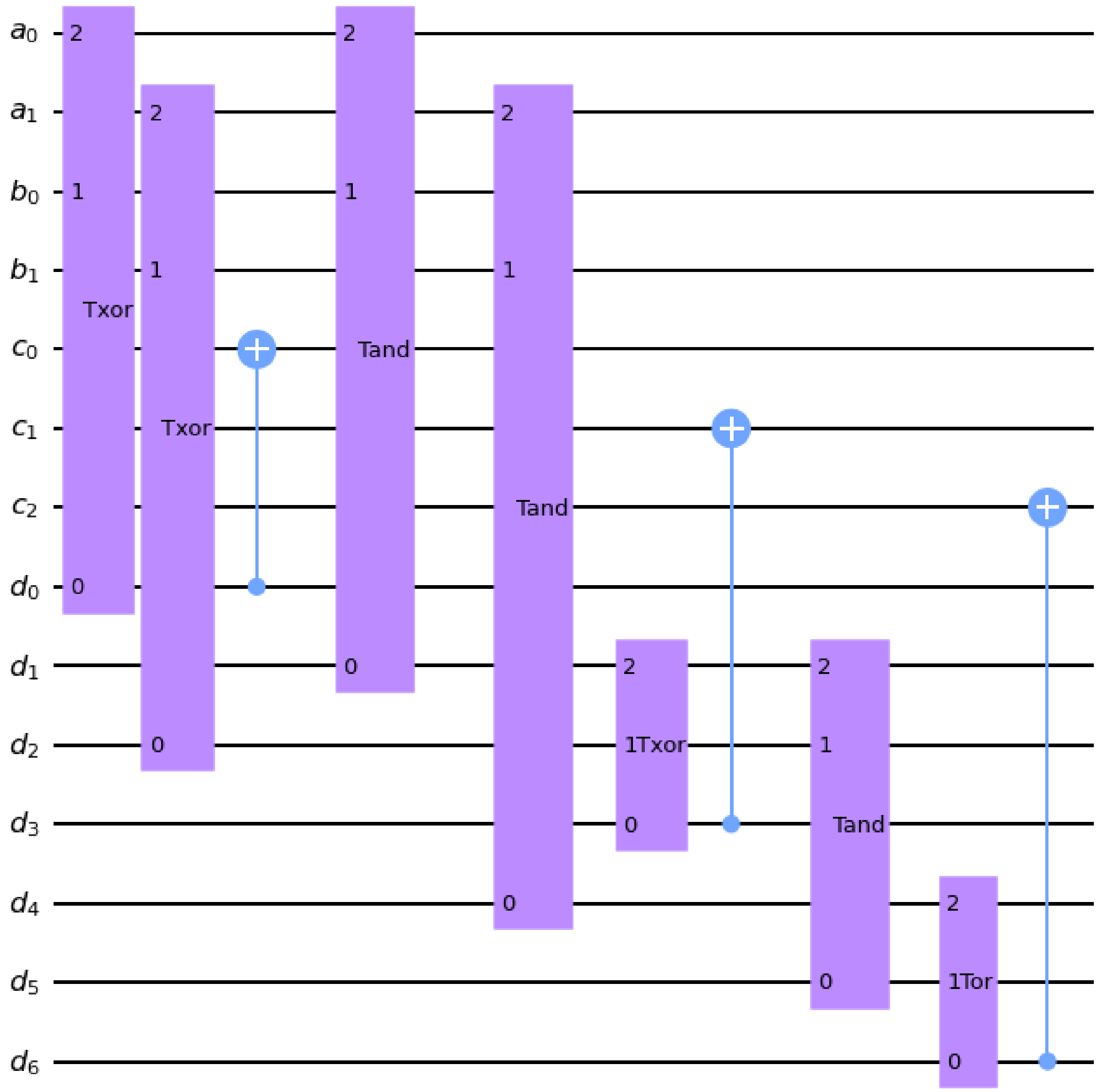
Your circuit C can use T_{XOR}, T_{AND}, T_{OR}, as well as $CNOT$, X, Z and H gates. Choose the appropriate number of ancillas, and then implement your circuit where indicated. The code afterwards will visualize your circuit as well run it on several test cases.

In [13]:

```
# Todo: fill in the number of ancillary qubits for your circuit
num_anc = 7

def create_two_bit_adder_with_scratch(num_anc):
```

```python
    A = QuantumRegister(2, name="a")
    B = QuantumRegister(2, name="b")
    C = QuantumRegister(3, name="c")
    D = QuantumRegister(num_anc, name="d")
    qc = QuantumCircuit(A,B,C,D)

    #### BEGIN YOUR CODE HERE ############################

    qc = append(qc, create_Txor, [0, 2, 7])
    qc.cx(7, 4)
    qc = append(qc, create_Tand, [0, 2, 8])
    qc = append(qc, create_Txor, [1, 3, 9])
    qc = append(qc, create_Txor, [8, 9, 10])
    qc.cx(10, 5)
    qc = append(qc, create_Tand, [1, 3, 11])
    qc = append(qc, create_Tand, [8, 9, 12])
    qc = append(qc, create_Tor, [11, 12, 13])
    qc.cx(13, 6)

    ##### END YOUR CODE HERE ###############################

    return qc

two_bit_adder_with_scratch = create_two_bit_adder_with_scratch(num_anc=num_anc)
two_bit_adder_with_scratch.draw(output='mpl')
```

Out[13]:



In [14]:

```
# Running test cases on your adder....
test_two_bit_adder(two_bit_adder_with_scratch, num_anc, has_scratch=True)
```

```
Testing two-bit adder with scratch...
OK.
```

1. Now we go one step further to implement a reversible two-bit adder that does the same thing as above except the scratch bits start **and end** in the zero state.

$$C\,|A, B, 0, 0\cdots 0\rangle = |A, B, A+B, 0\cdots 0\rangle$$

In other words, the scratch work is erased.

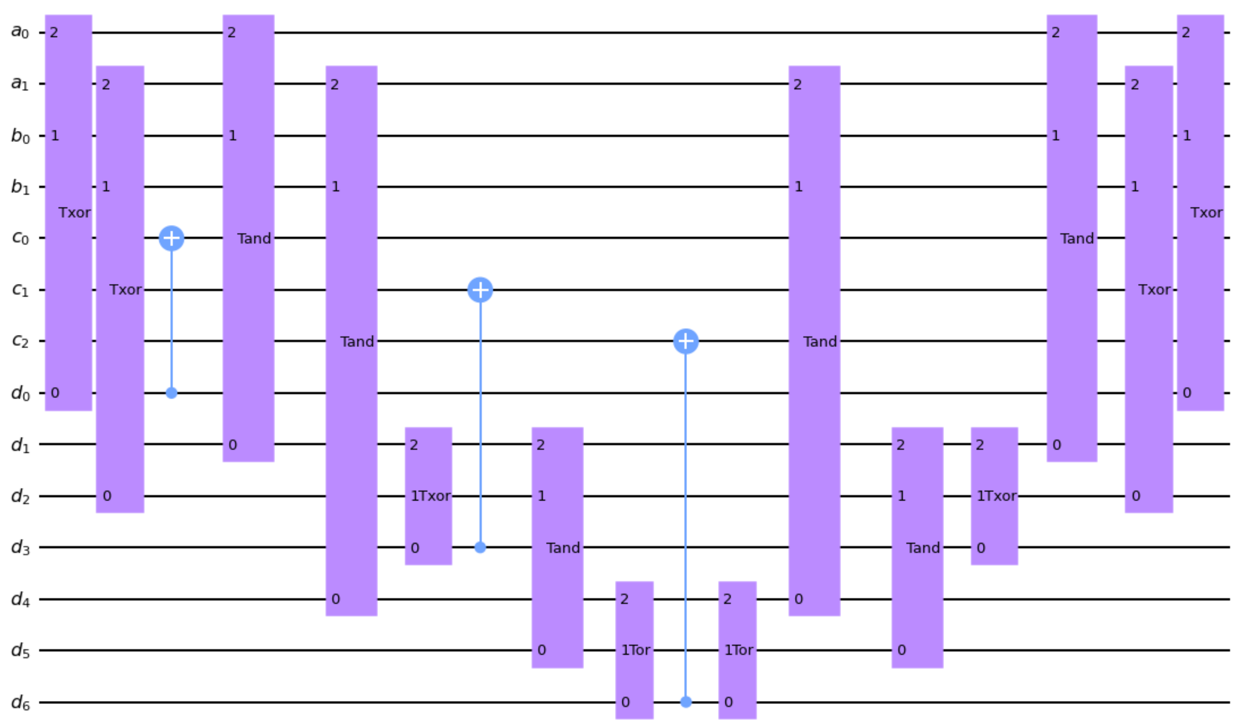
Hint: Use an additional ancilla to save the output, and then reverse the computation.

In [15]:

```
def create_two_bit_adder() -> QuantumCircuit:
    A = QuantumRegister(2, name="a")
```

```
    B = QuantumRegister(2, name="b")
    C = QuantumRegister(3, name="c")
    D = QuantumRegister(num_anc, name="d")
    qc = QuantumCircuit(A,B,C,D)

    #### BEGIN YOUR CODE HERE ############################

    qc = append(qc, create_Txor, [0, 2, 7])
    qc.cx(7, 4)
    qc = append(qc, create_Tand, [0, 2, 8])
    qc = append(qc, create_Txor, [1, 3, 9])
    qc = append(qc, create_Txor, [8, 9, 10])
    qc.cx(10, 5)
    qc = append(qc, create_Tand, [1, 3, 11])
    qc = append(qc, create_Tand, [8, 9, 12])
    qc = append(qc, create_Tor, [11, 12, 13])
    qc.cx(13, 6)

    qc = append(qc, create_Tor, [11, 12, 13])
    qc = append(qc, create_Tand, [8, 9, 12])
    qc = append(qc, create_Tand, [1, 3, 11])
    qc = append(qc, create_Txor, [8, 9, 10])
    qc = append(qc, create_Txor, [1, 3, 9])
    qc = append(qc, create_Tand, [0, 2, 8])
    qc = append(qc, create_Txor, [0, 2, 7])

    ##### END YOUR CODE HERE ###############################

    return qc

two_bit_adder = create_two_bit_adder()
two_bit_adder.draw(output='mpl')
```

Out[15]:



In [16]:

```
#Running test cases on your adder....
test_two_bit_adder(two_bit_adder, num_anc, has_scratch=False)
```

```
Testing two-bit adder without scratch...
OK.
```