**Introduction to Multiple Qubits and Entanglement**

**Multiple Qubits and Entangled States**

A single bit has two states, and a single qubit has two complex amplitudes. Similarly, two bits have two states:

And to describe the behavior of two qubits, we use four complex amplitudes. We describe these in a 4D vector:

The rules of measurement still hold:

The same implications also hold, such as the normalization condition:

If we have two separate qubits, we can describe their collective state as a tensor product:

The tensor product can actually describe a system of any number of qubits. For example, consider a three-qubit system (note the tensor product is associative):

Notice that for some number defining the number of qubits we have, we must keep track of complex amplitudes, meaning these vectors grow exponentially with the number of qubits. This is why quantum computers with many qubits are too difficult for classical computers to simulate.

Consider an example circuit:

A screenshot of a computer program

Description automatically generated

Each qubit has been initialized in the state , so we should expect to see:

We find the following:

A screenshot of a computer program

Description automatically generated

Notice that:

We have our expected result.

Exercises:

Find the following tensor products:

Write the state as two separate qubits.

Guess:

Also note:

When applying gates to multi-qubit statevectors, we use the tensor product on the matrix operators. For some , we can apply the X gate to and the H gate to as follows:

We then apply this matrix to our 4D statevector. Since this can get quite messy, this notation is often used:

We can actually have Qiskit do this for us by having the Aer simulator compile a single unitary matrix:

A screenshot of a computer program

Description automatically generated

Unsurprisingly, if we want to apply a quantum gate to only one qubit at a time, we take the tensor product of that gate with the identity matrix. For example:

Qiskit validates this calculation:

A screenshot of a computer program

Description automatically generated

Exercises:

Calculate the single qubit unitary (U) created by the sequence of gates: .

Consider a 3-qubit system :

Sidenote:

**VERY IMPORTANT NOTE:**

The Qiskit simulator and textbook calculates

The intuitive way to calculate is backwards. DO NOT DO THIS.

**Multi-Qubit Gates**

#1 The CNOT Gate

This was seen previously. It performs and X-gate on the target qubit if and only if the signal from the control qubit is . The gate is drawn as such:

A screenshot of a computer

Description automatically generated

Remember, in Qiskit, corresponds to the higher digit qubit. This is different in different sources. For this specific CNOT gate, the following matrix is what acts on the statevector:

When the CNOT gate is reversed, the following matrix is used:

The first matrix swaps the amplitudes of and in the statevector:

Let’s consider the effects of this on a qubit in superposition. We put one qubit in the state :

A screenshot of a computer program

Description automatically generated

As expected, since we have in the state and in the state , we have the overall state:

Now consider what happens when we apply CNOT for as the control qubit:

A screenshot of a computer

Description automatically generated

We have the statevector:

This makes sense, given the definition of CNOT we have, being that for as the control, it switches the probability amplitudes of states and . Consider the original state, but with more terms:

If we say that any 4D statevector can be written as the sums of scalars on basis vectors of the form:

It follows that for an overall state we have:

This is an extremely interesting statevector because it’s entangled. This specific state is known as a Bell state. It has a 50% chance of being measured in either or , and a 0% chance of being measured in or . This can be seen by a Qiskit simulation:

A screenshot of a graph

Description automatically generated

This combined qubit state **cannot** be written as the tensor product of two independent qubit states, which has very interesting implications. Although our qubits are in superposition, measuring one of them will inform us about the other’s state and will collapse both superpositions instantaneously at any distance. This is the “spooky action at a distance” 20th century physicists were worried about. For this specific example, it’s easy to see why this is the case.

We will suppose we measure , and that there exists some statevector for which does not imply .

*Proof:*

It is thus shown that measurement of one qubit must collapse the other qubit to the same state as the original qubit, and superposition will thus collapse for any entangled state.

There is no way to use shared quantum states to communicate. I did not understand the reasoning, but I’ll come back to it when I take quantum in school.

Visualization of this statevector is impossible on the Bloch sphere. The position of a Bloch vector along an axis corresponds to the expectation value of measuring in that specific basis. However, there is no single qubit basis for which a specific measurement is guaranteed, which does not describe an entangled state.

A statevector is simply four complex amplitudes, and there are infinitely many ways to map this to an image for visualization. One method is the **Q-sphere**, where each amplitude is represented by a blob on the surface, with the size of the blob being proportional to the amplitude. The color is also proportional to the phase of the amplitude.

A screen shot of a computer

Description automatically generated

Exercises:

Create a quantum circuit to produce the Bell state: . Use the statevector simulator to verify your result. Calculate the unitary of this circuit and use the Qiskit simulator to verify this unitary does in fact perform the desired transformation.

The math dictates that applying CNOT with the control qubit being , and the target qubit being .

A screenshot of a computer program

Description automatically generated

A screenshot of a graph

Description automatically generated

Qiskit agrees with my results.