A Peek Into Quantum Computing and its Applications

Agustin Garcia Flores & Nathan Reynolds

PHY432: Computational Methods in Physics
Dr. Oliver Beckstein

1 Problem

The project explores the fascinating world of quantum computing, focusing on key concepts and approaches such as quantum entanglement, teleportation, and cryptography. At the heart of quantum computing is the phenomena of quantum entanglement, a unique quantum mechanical property in which particles become correlated in such a way that the state of one (regardless of distance) instantly correlates with that of another.

This project seeks to investigate how quantum entanglement is used in computing to do jobs that conventional computers struggle with. In particular, we focus on the simplest examples of quantum entanglement: the 2-qubit Bell states, or Einstein-Podolsky-Rosen (EPR) pairs, and their M-qubit generalizations, Greenberger-Horne-Zeilinger (GHZ) states. There are four different Bell states [1]:

$$|\beta_{00}\rangle = \frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right); \tag{1a}$$

$$|\beta_{10}\rangle = \frac{1}{\sqrt{2}}\left(|00\rangle - |11\rangle\right); \tag{1b}$$

$$|\beta_{01}\rangle = \frac{1}{\sqrt{2}}\left(|01\rangle + |10\rangle\right); \tag{1c}$$

$$|\beta_{11}\rangle = \frac{1}{\sqrt{2}}\left(|01\rangle - |10\rangle\right). \tag{1d}$$

We call the first two states **maximally** entangled, while the last two are only **partially** entangled.

From [2], we see that the M-qubit generalization of Equation 1a is

$$\left|\beta_{00}^{M}\right\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle^{\otimes M} + |1\rangle^{\otimes M}\right), \tag{2a}$$

where $\otimes M$ denotes the tensor product of M qubits. We can then infer that the M-qubit generalizations of Equations 1b, 1c, and 1d are

$$\left|\beta_{10}^{M}\right\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle^{\otimes M} - |1\rangle^{\otimes M}\right); \tag{2b}$$

$$\left|\beta_{01}^{M}\right\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle^{\otimes \lceil M/2\rceil} \otimes |1\rangle^{\otimes \lfloor M/2\rfloor} + |1\rangle^{\otimes \lceil M/2\rceil} \otimes |0\rangle^{\otimes \lfloor M/2\rfloor}\right);\ \text{and} \tag{2c}$$

$$\left|\beta_{11}^{M}\right\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle^{\otimes \lceil M/2\rceil} \otimes |1\rangle^{\otimes \lfloor M/2\rfloor} - |1\rangle^{\otimes \lceil M/2\rceil} \otimes |0\rangle^{\otimes \lfloor M/2\rfloor}\right)\ \text{respectively,} \tag{2d}$$

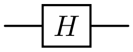
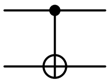
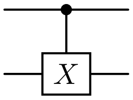
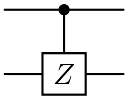
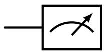
Hadamard		$\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$
Controlled-NOT	 or 	$\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{bmatrix}$
Controlled-Z		$\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & -1\end{bmatrix}$
Measurement		Projection onto $\|0\rangle$ or $\|1\rangle$
Qubit	—	Wire carrying a single qubit, with time flowing from left to right
Classical bit	═	Wire carrying a single classical bit

Table 1: A collection demonstrating the gates we anticipate using often in quantum circuits. The name of the gate is shown on the left column, their corresponding representation is shown in the middle column, and their mathematical expression (or an explanation) is shown in the right column. [1]

where we use the floor and ceiling of $M/2$ to avoid issues when M is odd. The states represented in these last four equations are called GHZ states.

To achieve these states on a quantum computer, we utilize quantum circuits. Certain schematic symbols are commonly used to represent unitary transforms, which are useful in the design of quantum circuits. We show the ones necessary for the quantum circuits we expect to produce in Table 1.

The quantum circuit to create a Bell state, $|\beta_{xy}\rangle$ for $x, y \in \mathbb{Z}_2$, is as follows:

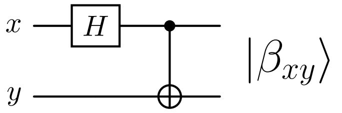


Figure 1: Quantum circuit for entangling two qubits. The final entangled state will be one of the Bell states, depending on the initial conditions of x and y. Note that we say $x, y \in \mathbb{Z}_2$ simply to understand how they relate to the mnemonic in Equation 3. To be more precise, we say that $x, y \in \{|0\rangle, |1\rangle\}$.

where

$$|\beta_{xy}\rangle \equiv \frac{1}{\sqrt{2}}\left(|0, y\rangle + (-1)^x |1, \bar{y}\rangle\right) \tag{3}$$

and $\bar{y}$ is the negation of y. [1]

Similarly, the quantum circuit to create a GHZ state, $|\beta_{xy}^M\rangle$ for $x, y \in \mathbb{Z}_2$ is as follows:

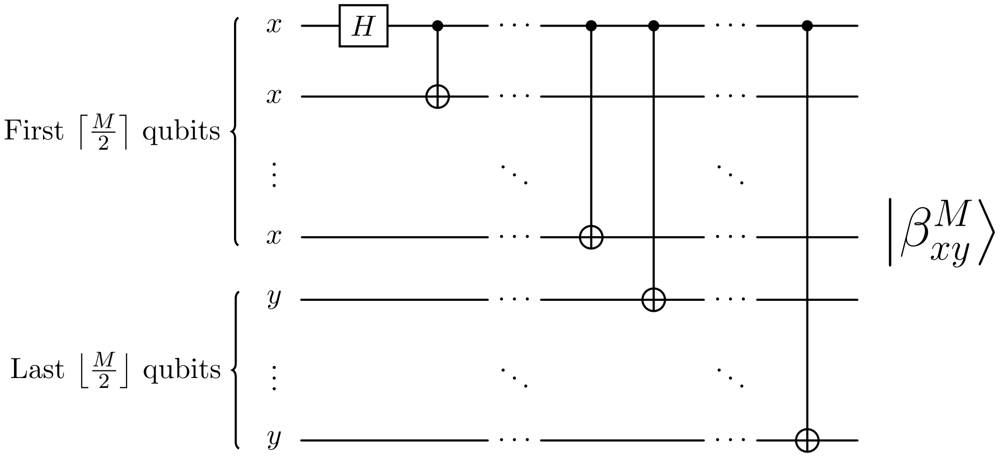

First $\lceil \frac{M}{2} \rceil$ qubits
Last $\lfloor \frac{M}{2} \rfloor$ qubits
x
x
x
y
y
H
$\left|\beta_{xy}^{M}\right\rangle$

Figure 2: Quantum circuit for entangling M qubits. The final entangled state will be one of the GHZ states, depending on the initial conditions of $x, y \in \{|0\rangle, |1\rangle\}$.

where an expression akin to Equation 3 can be derived:

$$\left|\beta_{xy}^{M}\right\rangle \equiv \frac{1}{\sqrt{2}}\left(|0\rangle^{\otimes\lceil M/2\rceil} \otimes |y\rangle^{\otimes\lfloor M/2\rfloor} + (-1)^{x}\,|1\rangle^{\otimes\lceil M/2\rceil} \otimes |\bar{y}\rangle^{\otimes\lfloor M/2\rfloor}\right). \tag{4}$$

Furthermore, the project digs into quantum teleportation, a unique phenomenon that allows quantum data (such as the state of a qubit) to be transferred between distant sites without physically moving the qubit itself. This mechanism uses entangled particles and is critical in quantum communication and network strategies. The quantum circuits necessary to perform quantum teleportation with Bell and GHZ states are shown in Figures 3 and 4, respectively.

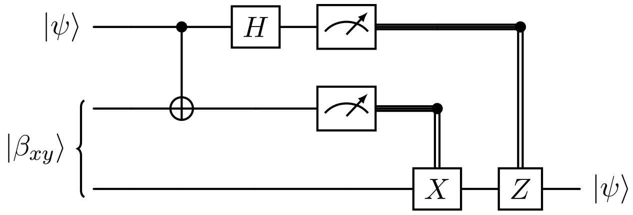


Figure 3: Quantum circuit for teleporting a qubit in an arbitrary state, $|\psi\rangle$, using some Bell state, $|\beta_{xy}\rangle$. The meters represent measurement, and the double lines coming out of them carry classical bits, which can then be used to make corrections to the resulting state, thus ensuring that we end up with the initial state $|\psi\rangle$. [1]

We also look into quantum cryptography, namely the BB84 protocol. Charles Bennett and Gilles Brassard developed the BB84 protocol in 1984, a fundamental technique for quantum key distribution (QKD) that uses quantum mechanics principles to assure secure communication. In this protocol, two users, say Alice and Bob, exchange quantum states via a quantum channel. These states are encoded using two separate bases, with the choice of base randomized for each bit sent. During the method, Alice broadcasts qubits in random states, which Bob then measures using randomly selected bases. Following transmission, they publicly compare their bases, without

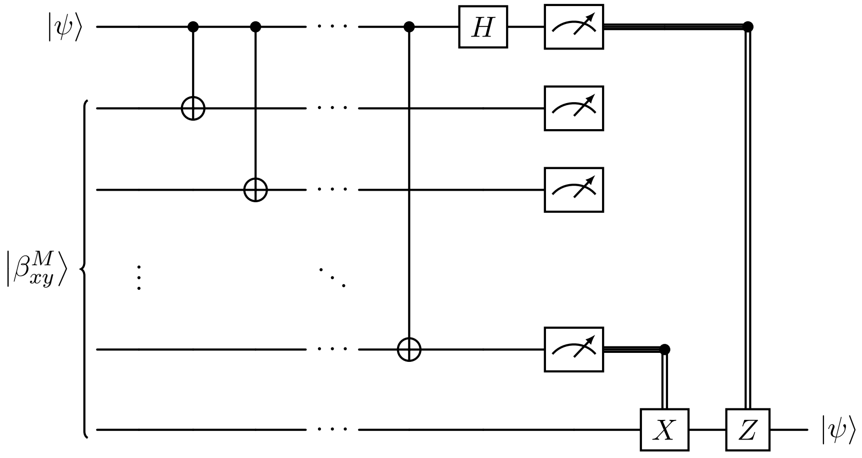


Figure 4: Quantum circuit for teleporting a qubit in an arbitrary state, $|\psi\rangle$, using some GHZ state, $\left|\beta_{xy}^{M}\right\rangle$. Adapted from [3]

releasing the specific bits, and delete the bits for which their bases did not match. This results in the generation of the filtered key. The protocol's security is based on the quantum principle that the act of observation impacts the state of a qubit, which means that any attempt to eavesdrop will invariably modify the state of the qubits, making any interception detectable. [4]

The two bases that Alice and Bob will choose from are the **Computational Basis (Z-Basis)** and the **Hadamard Basis (X-Basis)**. The computational basis consists of the states $|0\rangle$ and $|1\rangle$, which correspond to the north and south poles of the Bloch sphere, respectively. In quantum computing, these states are often used to represent the classical binary states 0 and 1. The hadamard basis consists of the states $|+\rangle$ and $|-\rangle$, where $|+\rangle \equiv |\beta_{00}\rangle$ (Equation 1a) and $|-\rangle \equiv |\beta_{10}\rangle$ (Equation 1b). These states lie on the equator of the Bloch sphere, orthogonal to the computational basis states.

The BB84 protocol has been enhanced to increase its security and feasibility. One of these enhancements is the use of decoy states to detect and prevent sophisticated eavesdropping tactics. These advancements have solidified BB84 as a dependable protocol for secure quantum communication, ensuring the security and anonymity of the key distribution method. In order to ensure true randomness during the protocol, we will investigate the use of quantum physics to generate valid random numbers. By placing a qubit in superposition, allowing it to exist in multiple potential states at the same time, and then measuring it, one can obtain truly random results, a desirable characteristic in resilient cryptographic algorithms. The quantum circuit demonstrating this is shown in Figure 5.

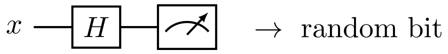


Figure 5: Quantum circuit showing how to put a qubit with initial state $x \in \{|0\rangle, |1\rangle\}$ in superposition then measuring to get a random result. The resulting values will be classical bits, either 0 or 1. Running this circuit multiple times will produce a series of random bits, which can then be used to generate numbers, or as a series to determine the bases of broadcasting or receiving in the BB84 protocol.

In conclusion, this study not only explains the fundamental operation of quantum computers, but it also lays the framework for comprehending their real-world applications in secure communication and other fields, suggesting a significant step toward fulfilling quantum computing technology's full promise.

2 Approach

To conduct the investigations, we use IBM simulators and quantum computers. We may test our code on actual quantum computers by using Qiskit, an open source software development kit that communicates with quantum computers via quantum circuits and algorithms. Specifically, we will first get comfortable with Qiskit by building the quantum circuits shown in Figures 1 and 2. A value of M for the GHZ states has yet to be determined, and will depend on the limitations of IBM quantum computers.

A successful software will produce results that align with our expectations based on our understanding of quantum mechanics. For example, when we entangle two qubits maximally (Equations 1a and 1b) the resulting classical bit value of qubit 1 and qubit 2 will be the same upon measurement; that is, if we measure the first qubit to be in the $|0\rangle$ state, then the second qubit should also be in the $|0\rangle$ state. For the partially entangled states (Equations 1c and 1d), the two measured values should be the negation of one another; i.e., measuring a $|0\rangle$ state for one qubit implies that the other qubit should be measured in the $|1\rangle$ state. Upon measurement, Qiskit returns a list of bits (or qubits) as a string, where bit n is the leftmost bit and bit 0 is the rightmost bit, allowing us to easily verify if the results are what we expect. For example, measuring two qubits in the $|0\rangle$ state will yield "00" where the left zero corresponds to qubit 1 and the right zero corresponds to qubit 0. This will first be verified using an IBM simulator, which only simulates a quantum computer and is thus not accompanied by all the physical noise experienced by real quantum computers. The data will include several strings that correspond to different measurement outcomes, and their frequency of occurrence in our simulation. A functioning code on an IBM simulator would then imply that errors in our data from the quantum computer will be products of noise. A similar, but more sophiscated and computationally complex code will be run to investigate entanglement with GHZ states.

Once the group feels comfortable with Qiskit and its syntax, we will move on to running the quantum circuits shown in Figures 3 and 4, where $|\psi\rangle$ will likely be trivial states such as $|0\rangle$ and $|1\rangle$ due to time limitations. If time permits, the teleportation of more complicated states will also be explored. Success, in this instance, will then be verified by ensuring that the target qubit collapses to the expected value. For instance, if our state to teleport is $|\psi\rangle = |0\rangle$ then the target qubit should collapse to a classical value of 0 every time the circuit is run. If the state to teleport is more complex, we'd have to investigate the states density matrix ($\rho = |\psi\rangle \langle\psi|$) to see what the expected values, and their distributions, should be. For example, if $|\psi\rangle = |+\rangle$, then

$$\rho = |+\rangle \langle+| = \begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix} \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} \end{pmatrix} = \begin{pmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \end{pmatrix},$$

where we look along the main diagonal for probability distributions. In this example, the element in the first row and first column, ρ_{00}, is the probability of the state collapsing to a classical bit value of zero, and the element in the second row and second column, ρ_{11}, is the probability of the state collapsing to a classical bit value of one. Thus, the density matrix of a more complex state can tell us what the distribution of the measurements should be. Once again, in order to avoid faulty code, we will first run our quantum circuits on an IBM simulator and ensure that the resulting data comes out as expected. Only then will the quantum circuits be run on a quantum computer.

In order to create a random number generator, the quantum circuit shown in Figure 5 will run several times on a quantum computer. An issue we anticipate is how to deal with noise on the IBM quantum computers. The circuit in Figure 5 is intentionally simple in order to limit the circuit's depth and complexity and thus limit the possibility of errors. Resulting strings with 50% of the bits having a value of 0 and 50% of the bits to having value of 1 will indicate success in this instance. Of course, one can apply several different protocols to make the quantum circuit in Figure 5 robust against noise, however that would greatly increase the circuit's depth and complexity, in addition to increasing the amount of time necessary to achieve this. For simplicity, since we are putting each qubit into a fair superposition, we expect 50% of the bits to have a value of 0 and 50% of the bits to have a value of 1.

The bit series generated in the previous part (call it series C) will also be used to randomly generate qubits that Alice will send to Bob in the BB84 protocol. A bit value of 0 means that Alice will send over a qubit in the $|0\rangle$ state, and a bit value of 1 means that Alice will send over a qubit in the $|1\rangle$ state. Two other random series of bits will then be used to determine the choice of bases for Alice and Bob (call them series A and series B, respectively). A bit value of 0 means that they chose the computational basis, and a bit value of 1 means they chose the hadamard basis. This will all be put together into one protocol and then simulated in a quantum circuit. In essence, bits from series C sent and measured in the same basis should yield the same outcome (they will yield the same outcome in the absence of noise). Bits from series C sent and measured in a different basis will *sometimes* ($\sim$ 50% of the time) yield the same outcome in principle. As a result, bits from series C will be kept if and only if Alice and Bob use the same basis. That is, if series $\mathrm{A}_i =$ series B_i for qubit i, then qubit i will be kept. Else, qubit i is omitted. Success, in this instance, is when the qubit value sent by Alice and measured by Bob, in the same basis, is the same. Of course, this will be true using the IBM simulator, but running this code on a quantum computer will occasionally cause the bit value to be different, even using the same basis, due to the amount of noise experienced by the quantum computer.

We may then export the data from the quantum computer and try to identify trends. For example, given the extreme sensitivity of quantum computers to external noise, how do errors scale with the number of qubits used? By combining quantum mechanics properties with Qiskit and IBM equipment, we will be able to investigate each of the problems described.

3 Objectives

1. (a) Perform entanglement between two qubit to create the Bell states, and between M qubits to create all GHZ states using an IBM simulator, then analyze the outcomes to ensure the simulation works as expected.

   (b) Run circuits from part 1(a) on an IBM quantum computer; analyze data to see how different it is from the outcomes in 1(a).

2. (a) Achieve quantum teleportation of $|0\rangle$ and $|1\rangle$ states with Bell states and GHZ states on an IBM simulator, then analyze the outcomes to ensure the simulation works as expected.

   (b) Run circuits from 2(a) on an IBM quantum computer; analyze data to see how different it is from the outcomes in 1(a).

   (c) **(Strech)** Achieve quantum teleportation of some arbitrary $|\psi\rangle$ state with Bell states and GHZ states on an IBM quantum computer; analyze data to see how different it is from the expected outcomes in the density matrix.

3. **(Strech)** Investigate true random number generation by exploiting the properties of quantum mechanics, then analyze the outcomes to ensure the simulation works as expected.

4. **(Strech)** Perform the BB84 protocol of QKD, then analyze the outcomes to ensure the simulation works as expected.

References

[1] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information.* Cambridge University Press, 10th anniversary ed., 2010.

[2] G. J. Mooney, G. A. L. White, C. D. Hill, and L. C. L. Hollenberg, "Generation and verification of 27-qubit greenberger-horne-zeilinger states in a superconducting quantum computer," *Journal of Physics Communications*, vol. 5, p. 095004, Sept. 2021.

[3] E. Jung, M.-R. Hwang, Y. H. Ju, M.-S. Kim, S.-K. Yoo, H. Kim, D. Park, J.-W. Son, S. Tamaryan, and S.-K. Cha, "Greenberger-horne-zeilinger versus w: Quantum teleportation through noisy channels," *Physical Review A*, vol. 78, July 2008.

[4] S. R. M and C. M. B, "Comprehensive analysis of bb84, a quantum key distribution protocol," 2023.