Introduction to Quantum Computing

Final report on the INTEREST program

Joint Institute for Nuclear Research
Meshcheryakov Laboratory of Information Technologies

Wave 4: 24May – 02 July 2021
Supervisors: Dr. Mihai Dima & Prof. Gheorghe Adam
Student: David Godos Valencia[1]

[1]Instituto de Física – Universidad Nacional Autónoma de México (IF-UNAM)
Mexico

Abstract

In the wave 4 project “Introduction to Quantum Computing” we have been working during 6 weeks on the basics of quantum computers. For this purpose, we were introduced to the key concepts that reign the behavior of this new technology: Spin and its properties; superposition, entanglement, and interference phenomena; and the qubit concept. We were also introduced to the tools needed to simulate these phenomena: Root CERN programming language and the IBM Qiskit platform. With the Qiskit platform we were able to study quantum gates that conforms the quantum logic and to explore the main quantum algorithms as the Grover’s Algorithm. In this work a brief summary of the theory is given. In addition, the results of the exercises made at class are presented. Finally, in the conclusion the most important aspects of the course are mentioned as well, the learning I got from participating in this project.

Table of Contents

Introduction

Nowadays, computer have helped us to solve many problems of our daily life, to make research and to explore new phenomena. The access to greater computing tools has allowed us to challenge new mysteries. Nevertheless, there are still a lot of problems that can not be solved with normal computing regardless the computation power we have. Those problems are related to the fields of cryptography, chemistry, and the simulation of quantum systems, among others. Fortunately for us, a new computing tool has emerged that could face these problems in the future: Quantum Computing (QC).

The main idea of QC is to use quantum systems to try to reproduce quantum systems properties. It replaces the classical bit with qubits, quantum bits, that can be in a superposition state, entangled, and even interfere with each other. Just the superposition property allows quantum computer to calculate faster than any current computer. Furthermore, entanglement allows communications to be more secure against hacking. In addition, interference is needed in some Quantum Algorithms to properly work and can be used together with entanglement to reduce computation time. The qubits are modified with quantum logical gates, which form quantum circuits as in classical logical circuits. However, classical algorithms do not work for quantum circuits. Therefore, new algorithms must be created to take advantage of quantum properties to solve problems that classical computer cannot.

For all the above, QC is an exciting field to study in. In this project we get an introduction to the basics of QC. The document is divided as follows: first it is presented the Spin as a 2-level quantum system; then, some experiments are shown in order to understand some properties that real qubits have; after that the quantum gates used to compose quantum circuits are presented; and finally, it is shown a quantum algorithm, the Grover’s Algorithm. At the conclusion, a summary of the work is given and some remarks about the project.

Spin Quantum Mechanics

The spin was discovered in 1922 through the Stern-Gerlach experiment. There, it was shown that inhomogeneous magnetic fields deflected neutral particles (hot silver atoms) from its trajectory. It was classically expected that the electrical-neutral particles traveled straight, but this experiment proved that particles have an intrinsic angular momentum like if they were spinning. As the silver atoms were deflected half up and half down, the experiment also shown that there were just two quantum states for the spin: ↑↓. Those systems with only two possible states are known as 2-level quantum systems.

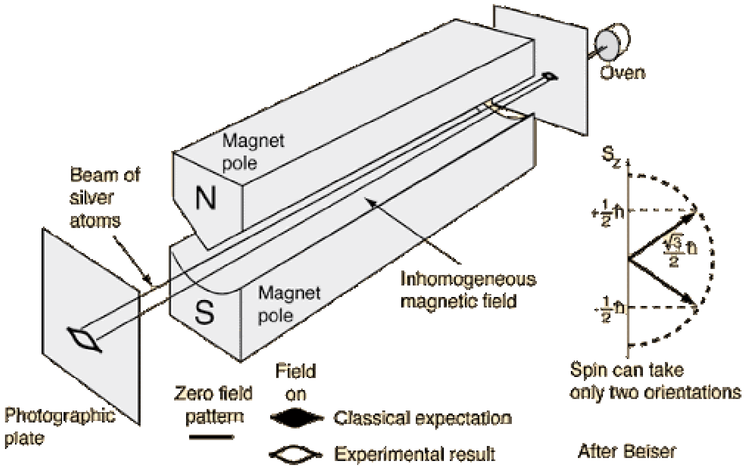
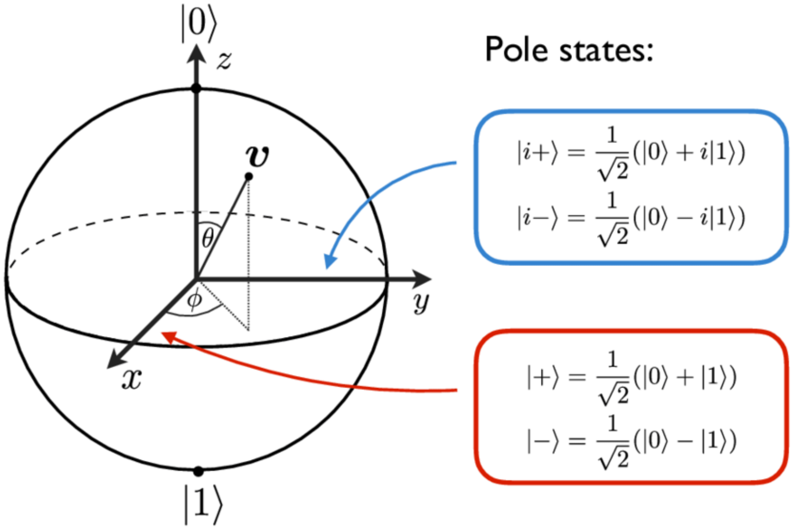


Figure 1.- Stern-Gerlach experiment diagram.

After the discovery of the spin more experiments followed. They changed the orientation of the magnets and observed the same phenomenon. But when they put consecutive Stern-Gerlach experiments they found some surprises. If they selected just the up spin after an experiment and then sent it through other magnets with the same orientation, they observed just the up spin. Nevertheless, if they changed the orientation of the magnets, then they observed two spin states in the new orientation (X or Y axis). Furthermore, after select one spin in the new basis, they sent it through more magnets with the original orientation (Z axis). What they observed were atoms with spins ↑↓ again. Those results demonstrated that it was not possible to measure the spin in two different bases.

We can consider that the electrically neutral atoms had spin in a **superposition** state ↑↓. This means that its spin had the same probabilities of being an up or down spin and until measurement it was not possible to know. In a superposition state, all possible states had the same probability of being. This property is in any quantum system. In addition, quantum systems can **interfere** with each other and with themselves. As for example in the electron slit experiment it was shown that an electron interfered with itself to form an interference pattern. The **entanglement** state refers when two particles are produced in the same process and each carries information of the other. So, if you measure the state of a particle, then you will know the state of the other particle. These properties are in 2-level quantum systems.

Qubits are chosen to be 2-level quantum systems like the spin. Consequently, we can think about qubits properties as the same to the spin properties. In other words, spin and qubits are equivalents to each other (and to any other 2-level quantum systems). Therefore, everything we have seen in this section applies to qubits too. One graphical way to represent these systems is with the Bloch Sphere were their state and phase can be seen.



$|0\rangle$
z
$\boldsymbol{v}$
θ
ϕ
y
x
$|1\rangle$
Pole states:
$|i+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle)$
$|i-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle)$
$|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$
$|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$

Figure 2.- The Bloch Sphere. In the Z axis are the $|0\rangle$ and $|1\rangle$ states. The X and Y axis have superposition states of previous ones. The angles θ, ϕ represent the phases. Any point in the sphere is a possible quantum state.

Qubit experiments

The following experiments will help us to understand better some characteristics of qubits.

Frequency Scan

In this exercise we want to find the resonant frequency of the qubit. This frequency is the energy difference between its ground and excited states. In the case of superconducting qubits, the quantum systems are designed so the higher levels had larger energy lap than that between the 1st and 2nd state. Therefore, the qubits are constrained into a 2-level system.

In order to find the peak we fit the function $\frac{A}{\pi}\cdot\frac{B}{(x-q_{freq})^2+B^2}+C$ to the data obtained at the IBM Qiskit platform while running the jupyter notebook. In that way the resonant frequency f_0 was obtained as $f_0 = 4.971\ GHz$.

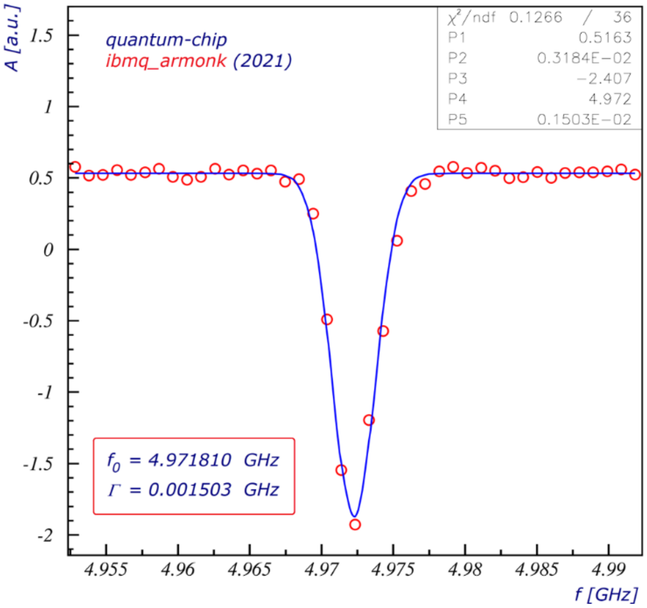


Figure 3.- Pulse amplitude of the qubit vs its frequency. The blue line is the fit performed while the red dots represent the experimental data.

Rabi qubit excitation

The π pulse determines the transition between the 2 states of the qubit. In this experiment we want to know what the is amplitude needed for this transition. The procedure was the following: we start with small amplitude pulses and then we increase the amplitude progressively. We checked after each pulse if the qubit flipped. The results were fit into a sine curve given by $A\cos\left(\frac{2\pi x}{d}-\phi\right)+B$; where d is the drive period. The A is the probability for obtaining $|1\rangle$ as final state. The first amplitude that produces the total swing is saved.

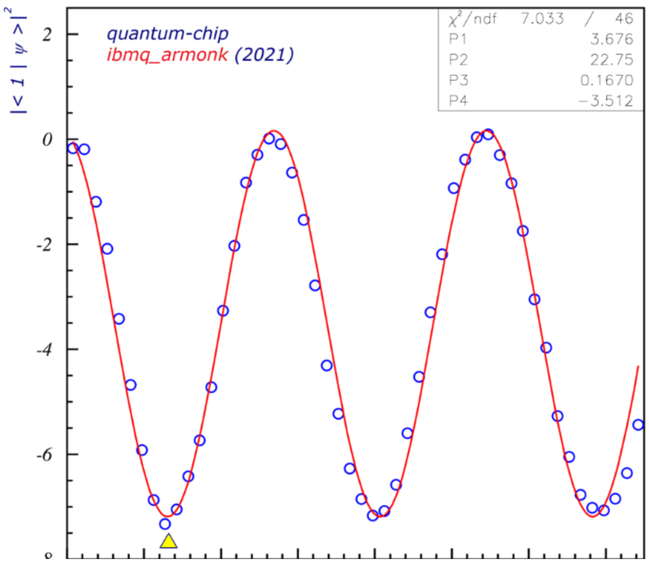


Figure 4.- π pulse amplitude of the qubit vs probability of flip. The red line is the sine fit curve.

We can see in the figure 4 that for this case the amplitude of the π pulse was about 0.112.

0/1 Discriminator

In this exercise we work with a qubit in a superposition state and we want to know in which state the qubit is after applying a π pulse. After performing the experiment in the Qiskit platform we obtain two different clusters form by populations of the up and down states. We can also try to stablish a borderline discrimination of the states by using a discriminator.

With the help of the borderline, we could also study the mixture between states. We only need to get the perpendicular line to the border and project the state population in that line. After that, we can plot histograms of both populations and adjust normal distributions. With those distributions we can see how good the discrimination is.

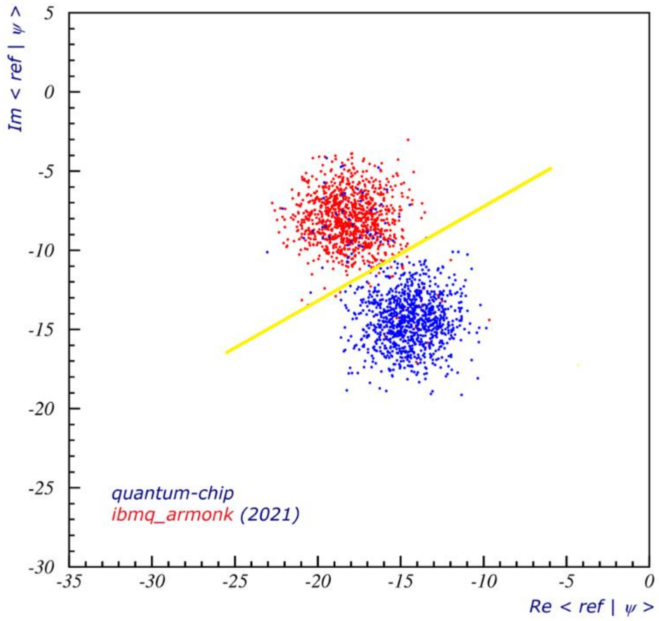


Figure 5.- Real vs Imaginary part of the bracket $\langle ref|\psi\rangle$.

Qubit relaxation Time T₁

The qubits are not ideal and, in that sense, they can not be always in an excited state. As all quantum states they tend to be in the less energetic state. The relaxation time is a parameter used to know how many time passes before our $|1\rangle$ state decays to $|0\rangle$. In this exercise we apply a 1 pulse, then we wait after we make a measurement of the state of the qubit. We vary the time delay and measure. The signal vs delay time is plotted and we get an exponential (relaxation) decay.

The function we fit is given by $Ae^{\frac{-x}{T_1}} + c$. We get a relaxation time of $T_1 = 6.734\ \mu s$.

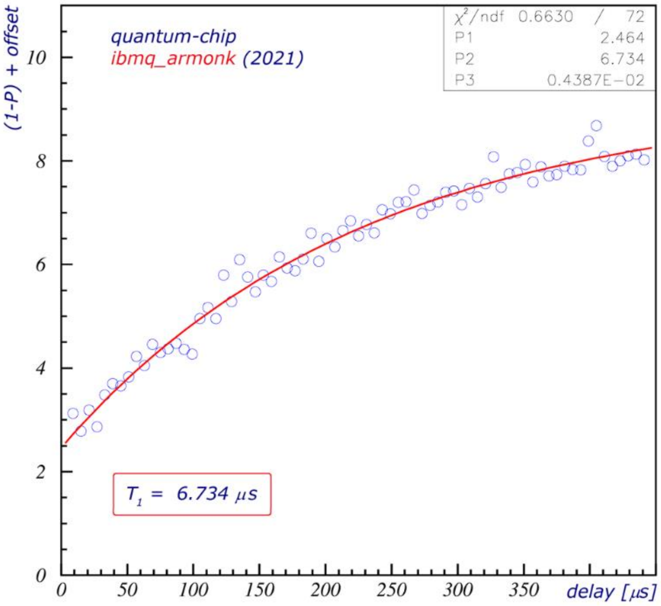


Figure 6.- Delay time vs (1-P) plus an offset for the relaxation time. The fit is shown in the red line.

Ramsey experiment

Now, for the Ramsey experiment we first apply a $\frac{\pi}{2}$ pulse, wait a time and then we apply another $\frac{\pi}{2}$ pulse. We then measure the qubits using frequencies equal to that of the pulses. We want to observe the oscillations at the difference in frequency between f$_0$.

The function was fitted with the following equation $Acos\left(2\pi d_f x - C\right) + B$, where $f_0 = f_{rough} + C$. And then we obtained a difference of about $\Delta f = 0.0113\ GHz$.

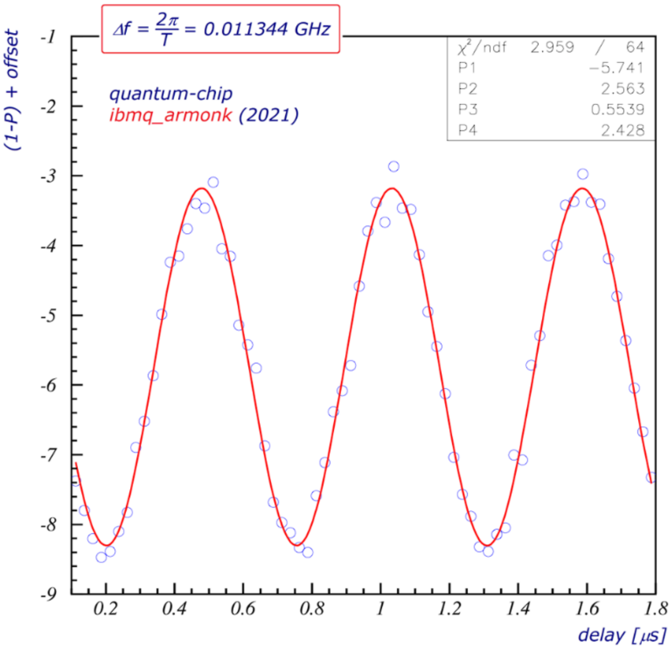


Figure 7.- Delay time vs (1-P) plus an offset for the difference in frequency. The fit is shown in the red line.

Hanh Echo experiment T$_2$

Along the T$_1$ parameter it exists the T$_2$ parameter or coherence time. In order to

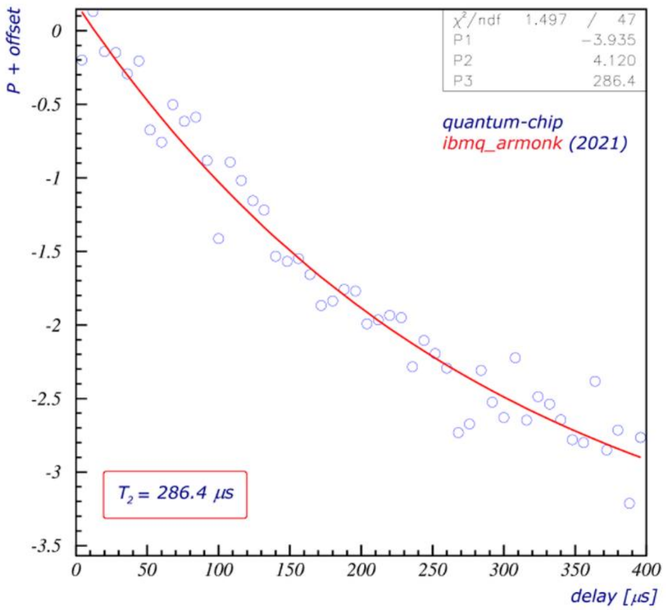


Figure 8.- Delay time vs (P) plus an offset for the coherence time T_2. The fit is shown in the red line.

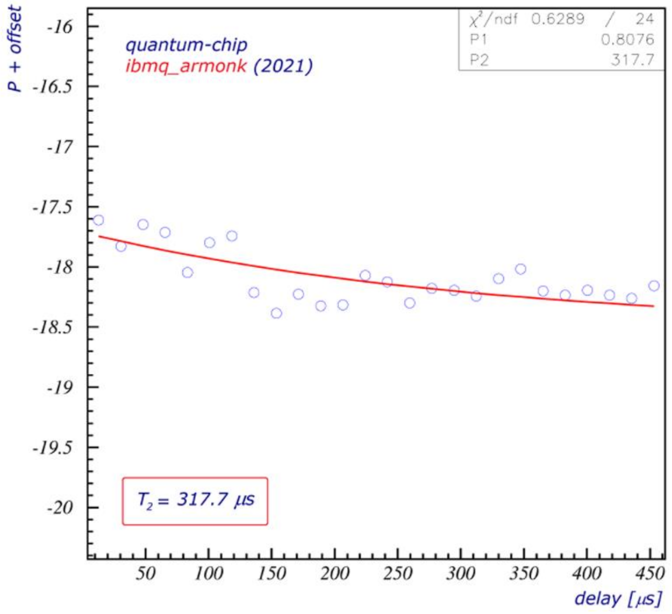


Figure 9.- Delay time vs (P) plus an offset for the coherence time T_{2DD}. The fit is shown in the red line.

perform this exercise, we follow the steps of the Ramsey experiment but adding a π pulse between the two $\frac{\pi}{2}$ pulses. This pulse reverses the accumulation of phase and results in an echo when we measure the qubit. Now the decay time gives of the exponential decay gives T_2. Then, using for the fit the same equation $Ae^{\frac{-x}{T_2}} + c$, we get that $T_2 = 286.4\ \mu s$.

Dynamical Decoupling T_{2DD}

For the last exercise we modified the Hahn experiment by applying several consecutive π pulses. This is the dynamical decoupling, and it cancels different noise frequency, lengthening qubit coherence times. At doing so and adjusting the same equation as above $Ae^{\frac{-x}{T_{2DD}}} + c$; we get the coherence time of $T_{2DD} = 317.7\ \mu s$.

QuLogic

Classical bits are modified through classical logical gates. Therefore, to modified qubits we need to use quantum logical gates. In the classical case we had the following gates: OR, AND, NOT, NOR, NAND, XOR, and XNOR. Those gates had its own truth table that relies on the value of the inputs to give an output. In contrast, quantum gates do not act the same as the classical ones. Its behavior can be seen like changing the position of the qubits in the Bloch Sphere. The main quantum gates and their function are:

- **Identity:** This gate is the identity gate denoted by I and its equivalent to the identity matrix in its function.
- **Hadamard:** This gate is special for us. It allows us to put a qubit into a superposition state or from a superposition state return to a ↑ or ↓ state. It can be see in matrix notation as: $H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = \frac{1}{\sqrt{2}}(|0\rangle\langle 0| + |0\rangle\langle 1| + |1\rangle\langle 0| - |1\rangle\langle 1|)$. It is easy to see that $HH = I,\ H|0\rangle = |+\rangle$ and $H|1\rangle = |-\rangle$.
- **Pauli (X, Y, Z):** These gates are applied to just one qubit and they act like rotations by π radians around the axis. The X gate rotates around the x axis, in other words: $X|1\rangle = |0\rangle$ and $X|0\rangle = |1\rangle$. In the matrix form it is represented by $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, and in Dirac's notation it is $X = |0\rangle\langle 1| + |1\rangle\langle 0|$. The Y and Z axis are represented in matrix and Dirac's notation as: $Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} = -i|0\rangle\langle 1| + i|1\rangle\langle 0|$, $Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = |0\rangle\langle 0| - |1\rangle\langle 1|$.
- **P-gate:** The phase gate performs a rotation around the Z axis. It needs a number φ that represents the rotation. It is denoted as $P(\phi) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{pmatrix}$.
- **CNOT:** It is a control gate, and it is different from all the above because it needs two qubits (the control and the target). Only if the state of the control bit is $|1\rangle$, then an X gate is applied to the target. The CNOT gate with the H gate allows us to create entangled qubits. The notation of the gate is in the figure 10, where the qubit q_0 is the control and q_1 is the target.

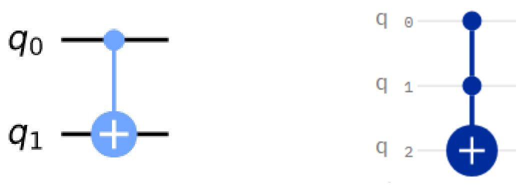
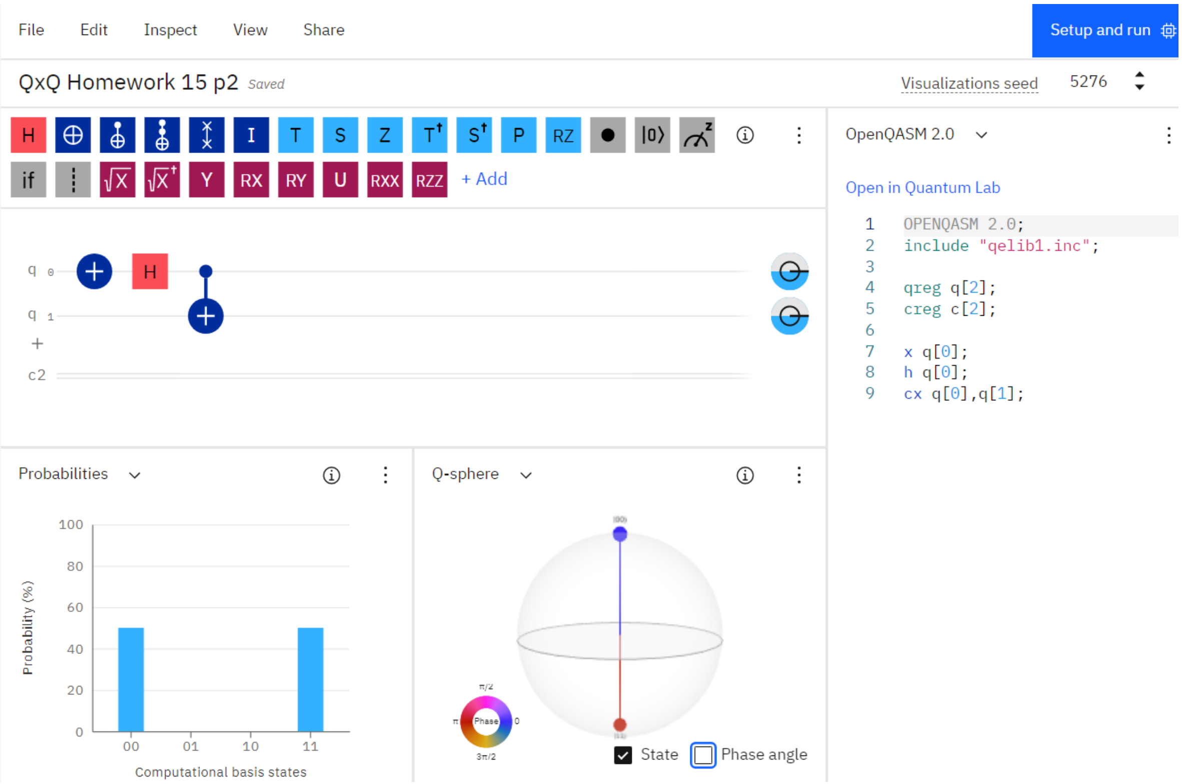


Figure 10.- At the left the CNOT gate represented graphically in the IBM Qiskit platform, at the right it is shown the CCNOT gate.

- **CCNOT:** Also known as Toffoli gate. It is a control control gate, like the CNOT gat but it needs three qubits (two control and the target). Only if the state of the control bits is |1,1⟩, then an X gate is applied to the target. The notation of the gate is in the figure 10, where the qubits q_0 and q_1 are the control and q_2 is the target.

The IBM Qiskit platform gives us the opportunity to build quantum circuits and to run them in quantum computers or in a simulator. The IBM Quantum Composer is where you can build graphically those circuits by dropping the blocks of quantum gates to the qubits. The platform looks like this:



```
OPENQASM 2.0;
include "qelib1.inc";

qreg q[2];
creg c[2];

x q[0];
h q[0];
cx q[0],q[1];
```

Figure 11.- IBM Quantum Composer interface.

In the image we can see that at the top are some buttons that represents the different operators we can use while creating a quantum circuit. Below, at the left we can find a probability graph that shows us the results at the end of the quantum circuit; meanwhile, at the right we can see a Bloch Sphere representing our code. At the center it is the code creator section. Each q line represents a qubit and each c line represents a classical bits. If n qubits are used then we are working in a $|q\rangle = |q_1\rangle \otimes |q_2\rangle \otimes |q_3\rangle \otimes \dots \otimes |q_n\rangle$ space. Classical bits are needed to store each result of the code. To set the value to store in the classical bit a measure block is needed. Finally, at the right we found a command section where we can also modify our circuit with code.

It is also possible to upload a Jupyter notebook in the platform and to code with python in the IBM Quantum Lab. Quantum circuits can also be created there with the qiskit python package. It is even possible to draw the circuit in the notebook.

Quantum algorithm

Quantum algorithms take advantage of quantum properties to perform tasks more efficiently or faster than classical algorithms. One example of this is the Grover's Algorithm (GA). It is aim for finding among a list of N items one in particular, reducing the iterations needed to do this. The best part is that the list can be of any kind of items, and it does not have to be organized. Classically for a list of N items one would need on average N/2 iterations; but the Grover's Algorithm reduces the iteration to $\sqrt{N}$.

GA is based on the use of an Oracle. First, our state $|s\rangle$ is put into a superposition state with equal amplitudes for all the possible outcomes $|s\rangle = \frac{1}{\sqrt{N}} \sum_i |\psi_\mathrm{i}\rangle$. Then this oracle can flip the phase of our solution state, lowering the average amplitude. The oracle can be seen as a reflection of the target state $|s\rangle$ denoted by $U_w = 2|w\rangle\langle w| - 1$. Later, a reflection of the state $|s\rangle$ is performed which set all the amplitudes positive again and it is given by $U_s = 2|s\rangle\langle s| - 1$. At doing so, only the solution amplitude increases while the rest decreases. In each iteration the Oracle and then the reflection of the general state is performed. This procedure is known as amplitude amplification. After $\sqrt{N}$ iterations we measure the results to find the solution state.

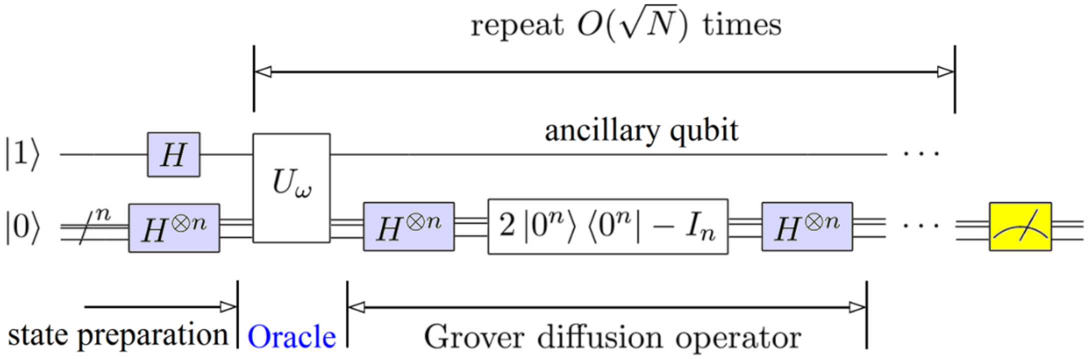


Figure 12.- Grover algorithm diagram.

The only thing we must do is to find an Oracle that can flip our solution state. We will give an example in a simple system of 2 qubits (N=4). We want to detect the $|1{,}1\rangle$ state. We can see that the CCNOT gate can flip the $|-\rangle$ state of the target qubit if the control qubits are in the state $|1,1\rangle$. Therefore, we will use this gate as the oracle. Then we will apply the Diffusion operator $D = -U_s U_w$. Finally we measure the control qubits to find our state.

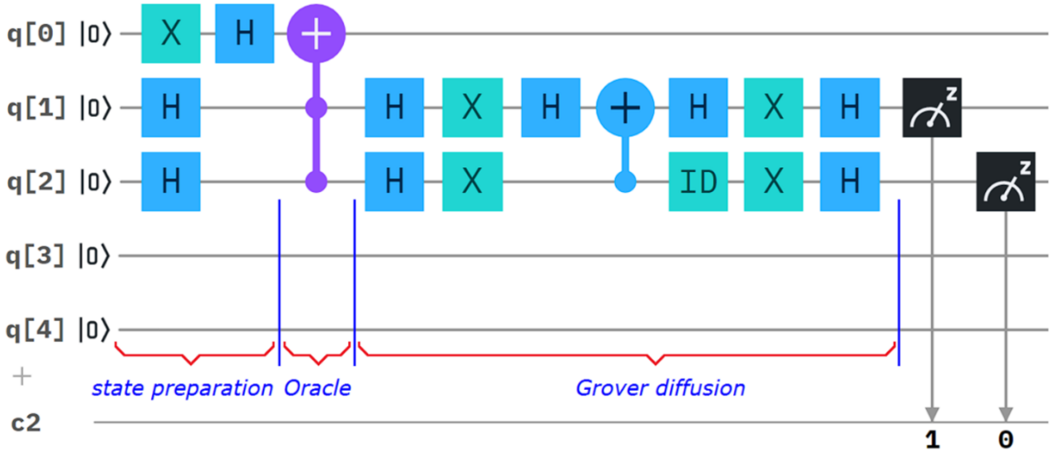
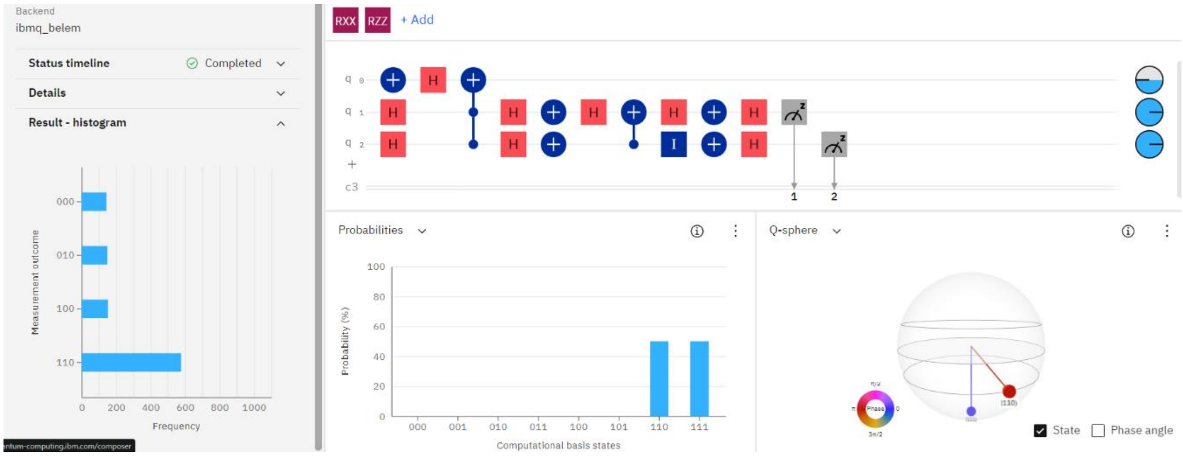


Figure 13.- Grover algorithm circuit for finding the state $|1,1\rangle$.

In the Qiskit circuit the Grover algorithm was reproduced. In the simulator 100% probabilities of finding the result state was obtained. But at checking with a real quantum computer, we got that the other possible results get around 15% of the times and the state $|1,1\rangle$ got 57.7%. Then it was obvious which was the result we were searching for.


Backend
ibmq_belem

Status timeline ✓ Completed

Figure 14.- Grover algorithm circuit results for finding the state $|1,1\rangle$. At the left is the histogram obtained by the ibmq_belem quantum computing after performed the algorithm.

Conclusions

In this work it has been presented the main principles of Quantum Mechanics that reign in the Quantum Computing. For example, the Spin is presented and its properties as a 2-level quantum system: entanglement, superposition, and interference. It was also reviewed the concept of qubit and its notation. In order to understand better the qubits characteristics some experiments were shown. Those experiments were: finding the resonant frequency and the amplitude of its π rotation pulse, the discrimination of states in the superposition state, relaxation time, the Ramsey experiment, and the dynamical decoupling experiment. Later on, we reviewed the quantum gates used in QC, the main blocks in the construction of quantum algorithms. With those blocks we made the Grover's Algorithm in the Qiskit platform.

Personally, I really enjoyed this project. I understood better the basics of QC and I learned more phenomena related to the qubits. Even more, the most memorable part for me were those experiments, because them allowed me to get deeper into the qubit concept and how it is in real life. That is in the sense that we will not work in the future with ideal qubits. I feel that the material had a good level of difficulty, but the advisors made them very easy to comprehend.

Acknowledgments

I deeply thank the Joint Institute of Nuclear Research as well Dr. Mihai Dima and Prof. Gheorghe Adam for the opportunity of participating in this project. Specially I am thankful to Dr Dima for the time and support during the lectures.

Bibliography

1. S. Yanofsky N., *An introduction to Quantum Computing*, arXiv, 0708.0261 (2007).