

# Report on INTEREST programme

## Introduction to quantum computing

NRNU MEPhI, Zubova Natalya

July 2021

### 1 Introduction

Quantum computing is solving problems by manipulating quantum objects: atoms, molecules, photons, electrons and specially created macrostructures. Using them allows scientists to achieve two quantum phenomena - superposition and entanglement. Quantum computing is a very promising area of research. In the future, researchers will be able to synthesize new materials, drugs, as well as simulate complex molecules and solve optimization problems that are not currently available for the most powerful computers.

### 2 Spin quantum mechanics

Apart from the degrees of freedom of being in space, the particles also have an inner degree of freedom called "spin". We say that a particle has spin  $s$  if its inner degree of freedom is described by a representation of the rotations group of dimension  $2s + 1$ . For example, "spin  $1/2$ " can be described by a representation of dimension 2, and "spin 1" can be described by a representation of dimension 3. In order to make this abstract statement clearer, we will look at several examples.

- Electrons have spin  $1/2$ , hence  $180^\circ$  difference in polarization ("up" and "down") means orthogonality.
- Photons have spin 1, hence  $90^\circ$  difference in linear polarization means orthogonality.

If we position two polarizers one after the other in the angles that were noted above, no particles will pass through.

A qubit (or quantum bit) is the quantum-mechanical analog of a classical bit. In classical computing, the information is encoded in bits, where each bit can have the value zero or one. In quantum computing, the information is encoded in qubits. A qubit is a two-level quantum system where the two basis qubit states are usually written as  $|0\rangle$  and  $|1\rangle$ . A qubit can be in state  $|0\rangle$ ,  $|1\rangle$

or (unlike a classical bit) in a linear combination of both states. The name of this phenomenon is superposition.

Any two-level quantum-mechanical system can be used as a qubit. Multilevel systems can be used as well, if they possess two states that can be effectively decoupled from the rest (e.g., ground state and first excited state of a nonlinear oscillator). Examples of qubit include the spin of the electron in which the two levels can be taken as spin up and spin down; or the polarization of a single photon in which the two states can be taken to be the vertical polarization and the horizontal polarization, two of the (multiple) discrete energy levels of an ion, a superconducting Transmon qubit or the nuclear spin states of an atom etc.

### 3 QuLogic

A quantum gate or quantum logic gate is a rudimentary quantum circuit operating on a small number of qubits. They are the analogues for quantum computers to classical logic gates for conventional digital computers. Quantum logic gates are reversible, unlike many classical logic gates. Some universal classical logic gates, such as the Toffoli gate, provide reversibility and can be directly mapped onto quantum logic gates. Quantum logic gates are represented by unitary matrices (Figure 1).

Quantum gates also can be represented as rotations around the Bloch sphere (Figure 2).

$$\begin{array}{l}
 \text{X Gate} \\
 \text{Bit-flip, Not} \\
 \text{Z Gate} \\
 \text{Phase-flip} \\
 \text{H Gate} \\
 \text{Hadamard} \\
 \text{T Gate} \\
 \text{Controlled Not} \\
 \text{Controlled X} \\
 \text{CNot} \\
 \text{Swap}
 \end{array}
 \begin{array}{l}
 \boxed{\text{X}} \\
 \boxed{\text{Z}} \\
 \boxed{\text{H}} \\
 \boxed{\text{T}} \\
 \begin{array}{c} \bullet \\ | \\ \boxed{\text{X}} \\ | \\ \oplus \end{array} \\
 \begin{array}{c} \times \\ | \\ | \\ \times \end{array}
 \end{array}
 \equiv
 \begin{array}{l}
 \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \\
 \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \\
 \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \\
 \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \\
 \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} \\
 \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}
 \end{array}
 =
 \begin{array}{l}
 \beta|0\rangle + \alpha|1\rangle \\
 \alpha|0\rangle - \beta|1\rangle \\
 \frac{\alpha + \beta|0\rangle + \alpha - \beta|1\rangle}{\sqrt{2}} \\
 \alpha|0\rangle + e^{i\pi/4}\beta|1\rangle \\
 a|00\rangle + b|01\rangle + \\
 d|10\rangle + c|11\rangle \\
 a|00\rangle + c|01\rangle + \\
 b|10\rangle + d|11\rangle
 \end{array}$$

Figure 1: Main quantum gates

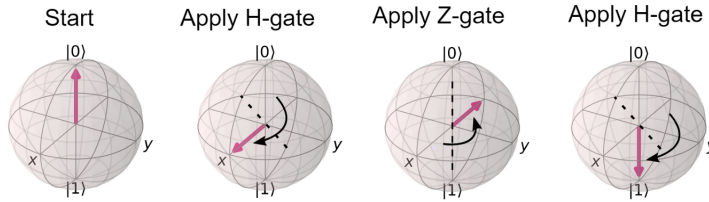


Figure 2: Bloch sphere

## 4 Qubit experiments

In this section, we'll look at qubit experiments conducted on the IBM Q Experience platform, which allows access to real quantum computers.

### 4.1 Qubit frequency scan

The qubit frequency is the difference in energy between the ground and excited states, which we label the  $|0\rangle$  and  $|1\rangle$  states, respectively. Though superconducting qubits exhibit a number of energy levels, they are engineered for non-linearity to keep the higher levels out of frequency reach, thereby essentially constraining the qubit to a two-level system.

We define the frequency range that will be swept in search of the qubit. Since this can be arbitrarily broad, we restrict ourselves to a window of 40 MHz around the estimated qubit frequency. We step the frequency in units of 1 MHz. We will create a pulse schedule by defining this frequency as a parameter. At each frequency, we will send a drive pulse of that frequency to the qubit and measure immediately after the pulse. Pulse envelopes used are the same for all frequencies.

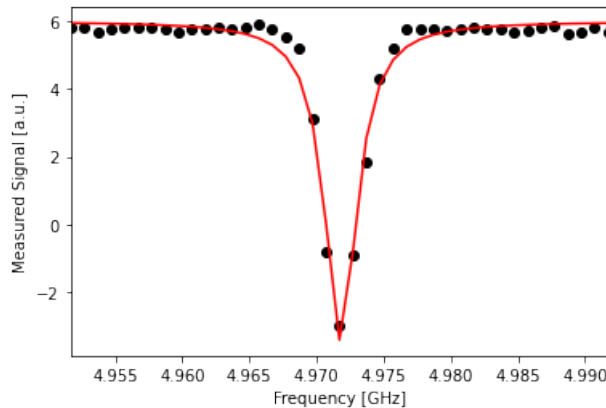


Figure 3: Qubit frequency scan

The peak near the center corresponds to the location of the qubit frequency. The signal shows power-broadening, which is a signature that we are able to drive the qubit off-resonance as we get close to the center frequency. To get the value of the peak frequency, we will fit the values to a resonance response curve, which is typically a Lorentzian shape. We get that a qubit frequency is equal to 4.97181 GHz.

## 4.2 Rabi qubit excitation

A  $\pi$  pulse determines the transition  $|0\rangle - |1\rangle$  - i.e. an X gate (inverter) action. This experiment determines the amplitude needed for the transition.

We will change the drive amplitude in small increments and measuring the state of the qubit each time. We expect to see oscillations which are commonly named Rabi oscillations, as the qubit goes from  $|0\rangle$  to  $|1\rangle$  and back (Figure 4). Amplitude is equal to 0.23400325324347926.

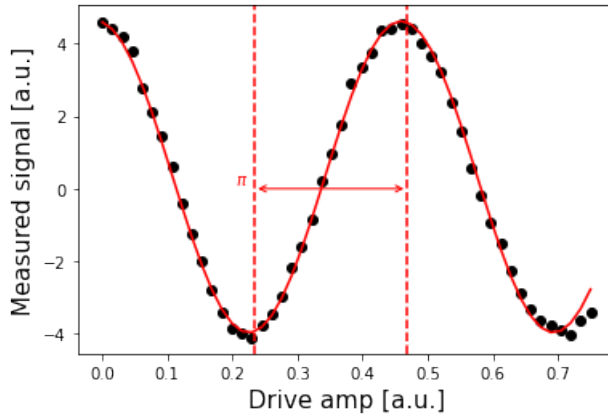


Figure 4: Amplitude

## 4.3 Determining 0 vs 1

Once our  $\pi$  pulses have been calibrated, we can now create the state  $|1\rangle$  with good probability. We can use this to find out what the states  $|0\rangle$  and  $|1\rangle$  look like in our measurements, by repeatedly preparing them and plotting the measured signal. This is what we use to build a discriminator, which is simply a function which takes a measured andkerneled complex value ( $meas_{level}=1$ ) and classifies it as a 0 or a 1 ( $meas_{level}=2$ ).

We can visualize the two populations which we have prepared on a simple scatter plot, showing results from the ground state program in blue and results from the excited state preparation program in red. We can clearly see that the two populations of  $|0\rangle$  and  $|1\rangle$  form their own clusters (Figure 5). Kerneled measurement results (from  $meas_{level}$ ) are classified (into  $meas_{level}$ ) by applying a

discriminator which optimally separates these two clusters. Optimal separation is simply a line in the IQ plane, equidistant from the average results we plotted above in the large dot, and normal to the line connecting the two dots.

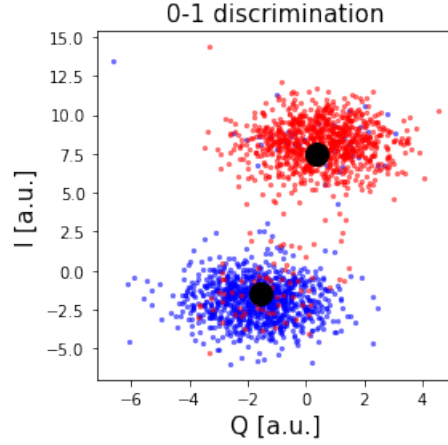


Figure 5: 0-1 discrimination

#### 4.4 $T_1$ determination via Inversion Recovery

$T_1$  is the qubit's relaxation time from the excited state  $|1\rangle$  to the ground state  $|0\rangle$ . This is a major hurdle in quantum computing that limits the exec duration of quantum codes.

The measurement uses a  $\pi$  pulse (defined above): we apply 1 pulse + delay + measure. Then we vary the delay and re-run for a number of delays. We plot the signal vs. delay time = exponential (relaxation) decay.

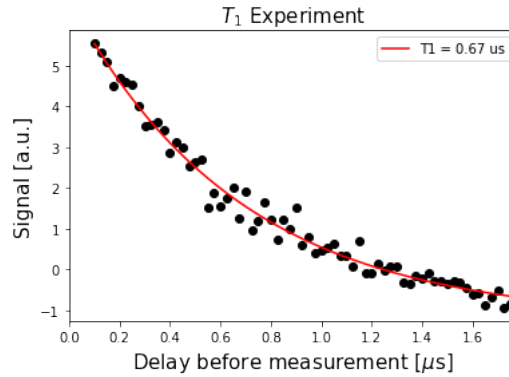


Figure 6:  $T_1$  experiment

## 4.5 Precise frequency measurement with a Ramsey Experiment

We can determine the qubit frequency to better precision. This is done using a Ramsey pulse sequence. In this pulse sequence, we first apply a  $\pi/2$  pulse, wait some time  $\delta t$ , and then apply another  $\pi/2$  pulse. Since we are measuring the signal from the qubit at the same frequency as the pulses, we should observe oscillations at the difference in frequency between the applied pulses and the qubit.

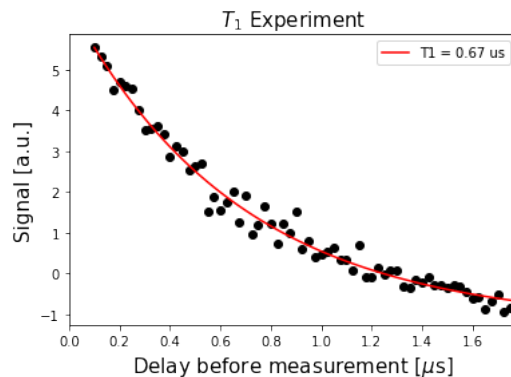


Figure 7: Ramsey experiment

Our updated qubit frequency is now 4.971993 GHz. It used to be 4.971936 GHz.

## 4.6 Measuring $T_2$ using Hahn Echoes

For the  $T_2$  measurement (Hahn echo) we follow Ramsey and add a  $\pi$  pulse between the two  $\pi/2$  pulses. This pulse (at time  $\tau$ ) reverses the accumulation of phase and results in an echo at time  $2\tau$ , when we measure the qubit (with a  $\pi/2$  pulse).

The decay time for the Hahn echo experiment gives the coherence time,  $T_2$ .

## 4.7 Dynamical decoupling

The Hahn  $\pi$  pulse eliminated some of the slow varying noise due to phase accumulation reversal. This can be extended to noise that is not quasi-static - by applying several consecutive  $\pi$  pulses. This is known as dynamical decoupling, and allows us to cancel different noise frequencies - thereby lengthening qubit coherence times.

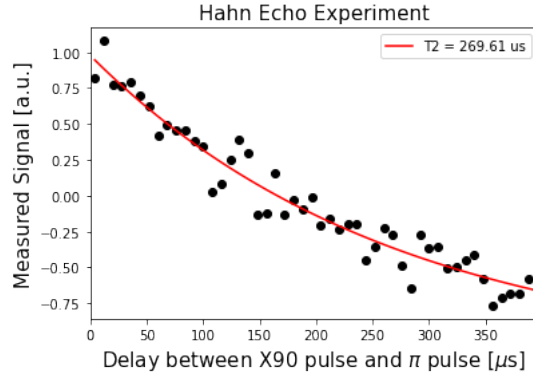


Figure 8: Hahn Echo Experiment

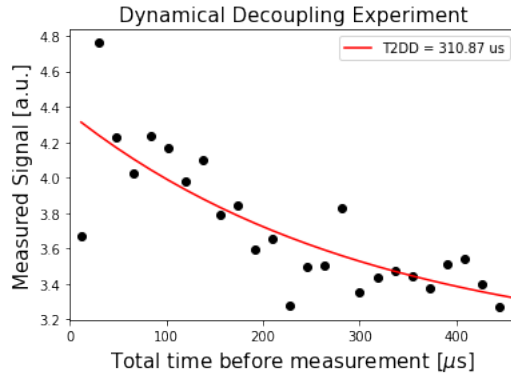


Figure 9: Dynamical decoupling experiment

## 5 Quantum algorithms

In quantum computing, a quantum algorithm is an algorithm which runs on a realistic model of quantum computation, the most commonly used model being the quantum circuit model of computation. A classical (or non-quantum) algorithm is a finite sequence of instructions, or a step-by-step procedure for solving a problem, where each step or instruction can be performed on a classical computer. Similarly, a quantum algorithm is a step-by-step procedure, where each of the steps can be performed on a quantum computer. Although all classical algorithms can also be performed on a quantum computer, the term quantum algorithm is usually used for those algorithms which seem inherently quantum, or use some essential feature of quantum computation such as quantum superposition or quantum entanglement.

Problems that are fundamentally unsolvable by classical algorithms (so called undecidable problems) cannot be solved by quantum algorithms either. The

added value of quantum algorithms is that they can solve some problems significantly faster than classical algorithms. The best-known examples are Shor's algorithm and Grover's algorithm. Shor's algorithm is a quantum algorithm for integer factorization. Simply put, when given an integer  $N$ , it will find its prime factors. It can solve this problem exponentially faster than the best-known classical algorithm can. Grover's algorithm can search an unstructured database or unordered list quadratically faster than the best classical algorithm with this purpose.

## 5.1 Grover's algorithm

In mathematical terms, Grover's algorithm solves the problem of an unstructured search. It is a quantum algorithm for finding the input value  $x_0$  of an Oracle function  $f(x)$  with  $f(x_0) = 1$  and  $f(x) = 0$  for all other values of  $x$ . The strength of the algorithm is that when the Oracle function has  $N$  possible input values, the algorithm only requires  $\mathcal{O}(\sqrt{N})$  evaluations of the Oracle, while the best classical algorithm needs  $\mathcal{O}(N)$  evaluations.

The steps of Grover's algorithm are given as follows:

Initialize the system to the uniform superposition over all states  $|s\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle$ . Perform the following "Grover iteration"  $r(N)$  times: Apply the operator  $U_\omega$ . Apply the Grover diffusion operator  $U_s = 2|s\rangle\langle s| - I$ . Measure the resulting quantum state in the computational basis. For the correctly chosen value of  $r$ , the output will be  $|\omega\rangle$  with probability approaching 1 for  $N \gg 1$ . Analysis shows that this eventual value for  $r(N)$  satisfies  $r(N) \leq \left\lceil \frac{\pi}{4} \sqrt{N} \right\rceil$ .

Implementing the steps for this algorithm can be done using a number of gates linear in the number of qubits. Thus, the gate complexity of this algorithm is  $\mathcal{O}(\log(N)r(N))$ , or  $\mathcal{O}(\log(N))$  per iteration.

Consider the case of Grover's algorithm for  $N = 4$  which is realized with 2 qubits. In this particular case, only one rotation is required to rotate the initial state  $|s\rangle$  to the winner  $|\omega\rangle$ :

- Following the above introduction, in the case  $N = 4$  we have

$$\theta = \arcsin \frac{1}{2} = \frac{\pi}{6}.$$

- After  $t$  steps, we have

$$(U_s U_\omega)^t |s\rangle = \sin \theta_t |\omega\rangle + \cos \theta_t |s'\rangle,$$

where

$$\theta_t = (2t + 1)\theta.$$

- In order to obtain  $|\omega\rangle$  we need  $\theta_t = \frac{\pi}{2}$ , which with  $\theta = \frac{\pi}{6}$  inserted above results to  $t = 1$ . This implies that after  $t = 1$  rotation the searched element is found.



We will now follow through an example using a specific oracle.

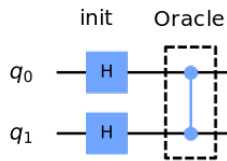
Oracle for  $|\omega\rangle = |11\rangle$  Let's look at the case  $|w\rangle = |11\rangle$ . The oracle  $U_\omega$  in this case acts as follows:

$$U_\omega|s\rangle = U_\omega \frac{1}{2} (|00\rangle + |01\rangle + |10\rangle + |11\rangle) = \frac{1}{2} (|00\rangle + |01\rangle + |10\rangle - |11\rangle).$$

or:

$$U_\omega = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

which you may recognise as the controlled-Z gate. Thus, for this example, our oracle is simply the controlled-Z gate:



Reflection  $U_s$  In order to complete the circuit we need to implement the additional reflection  $U_s = 2|s\rangle\langle s| - \mathbb{1}$ . Since this is a reflection about  $|s\rangle$ , we want to add a negative phase to every state orthogonal to  $|s\rangle$ .

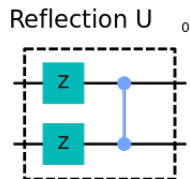
One way we can do this is to use the operation that transforms the state  $|s\rangle \rightarrow |0\rangle$ , which we already know is the Hadamard gate applied to each qubit:

$$H^{\otimes n}|s\rangle = |0\rangle$$

Then we apply a circuit that adds a negative phase to the states orthogonal to  $|0\rangle$ :

$$U_0 \frac{1}{2} (|00\rangle + |01\rangle + |10\rangle + |11\rangle) = \frac{1}{2} (|00\rangle - |01\rangle - |10\rangle - |11\rangle)$$

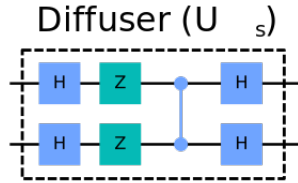
i.e. the signs of each state are flipped except for  $|00\rangle$ . As can easily be verified, one way of implementing  $U_0$  is the following circuit:



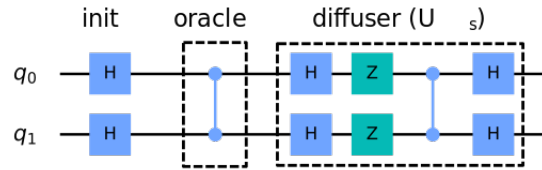
Finally, we do the operation that transforms the state  $|0\rangle \rightarrow |s\rangle$  (the H-gate again):

$$H^{\otimes n}U_0H^{\otimes n} = U_s$$

The complete circuit for  $U_s$  looks like this:



Full Circuit for  $|w\rangle = |11\rangle$  Since in the particular case of  $N = 4$  only one rotation is required we can combine the above components to build the full circuit for Grover's algorithm for the case  $|w\rangle = |11\rangle$ :



We implement Grover's algorithm for the above case of 2 qubits for  $|w\rangle = |11\rangle$ . The results of running the circuit on a simulator and a real quantum computer are shown in Figure 10.

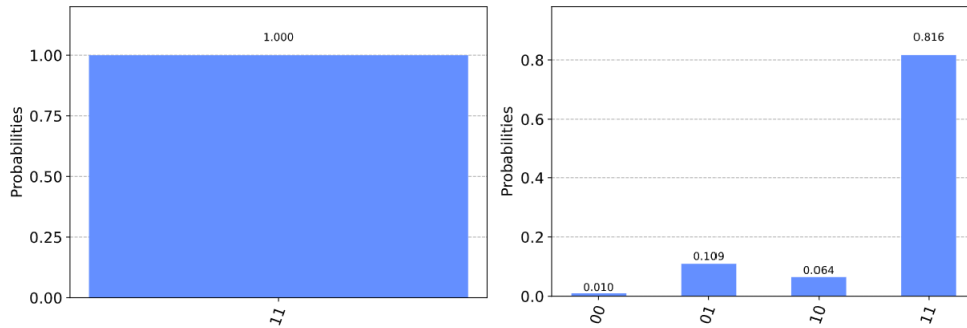


Figure 10: Results of running circuit on a simulator and real device

We confirm that in the majority of the cases the state  $|11\rangle$  is measured. The other results are due to errors in the quantum computation.

## 6 Conclusion

In this programme, we learned the basics of quantum computing. We got acquainted with the physical implementation of qubits, with quantum gates, quantum logic and quantum algorithms. Quantum algorithms allows us to show what are the advantages of quantum computers. We also got acquainted with very useful tools for working in the field of quantum computing, such as packages such as libraries SU2, CPX, ROOT. Experiments were carried out on the IBM Q Experience platform that allows to access real quantum computers. I find these skills to be very useful for people not only interested in quantum computing, but also planning to work in this field. I would like to express my gratitude to the professors for such an interesting experience.