

Introduction to Quantum Development

Jose Luis Hevia Oliver

¹ Alhambra, Madrid, 28037, Spain²
Jluis.hevia@a-e.es

Abstract. Brief presentation to introduce to the Quantum basics needed to understand the Quantum Computing Principles.

Keywords: Quantum Computer, Quantum Mechanics, Qubits.

1 Quantum Computing introduction

We live an amazing moment of the history, again: we are living the raise of the quantum computers. We have been spectators of the birth and evolution of exciting technologies related to digital information and it's process: from oldest x86 first IBM 16bits to the most powerful i7 x64 multicore processors; from the first gross mobile phone to the most advanced smartphones; from modem and copper communications cable to 5g internet or optical fiber and true high speed; and a lot of discovers we cannot talk about in one paper. All these technologies working together has made possible an incredible world of opportunities, thanks in large measure to the features that bring us the software and the computers.

But in the era in which we live, there are still challenges that not even the most modern computers can help us solve in a reasonable time. All this challenges must be attended in order to solve a lot of uses cases that could make the human raise, move forward a new step in its own evolution: discover news ways to create energy sources that reduces drastically the climate change, discover new materials, discover new molecular combinations to create better drugs or treatments to ill people...and much more. We have a lot of process force to analyze business data... but we don't have enough power to analyze the nature.

In this context is where quantum computers appear to the rescue. Quantum Computers that make possible the usage of the fundamental part of the nature, to know more about the nature; that can give us a power of computation never seen before; and that require us a new way to address the problems.

Copyright © 2020 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

The principles of a Quantum computer are based on Quantum Mechanics! And these principles are like magic. For a long time, brilliant physicist had been trying to explain the marvel behavior of the light particles. And most of the main theories and experiments have these principles in common: Matter at Quantum level can be found simultaneously in multiple and different configurations (states). That is known as *superposition*. Also, these *states* could interfere with others like waves in a pond. And some of these *states* could be connected with others, in such a way that if one changes its state, the other also. Giving to us the *entanglement*'s principle. But not everything is as beautiful as it seems. These *states* can be observed but –and this is a big problem- the observation alters reality (collapse). A good starting point to know more about that is the experiment of the *Schrodinger's cat* and the *Heisenberg's* uncertainty principle. We have all the possible realities in our hands, but in the moment we need to access it, it collapses into a single reality.

Thanks to these principles, quantum computers provides to us the qubits. These “magical artifacts”, makes possible the access a power of computation never seen before. Qubits makes possible the use of a power that *is not simply but exponential* (see Fig. 1). When a classical bit can store one value (0 or 1) at a time, the qubit can handle 0,1 or both at same time. So, a quantum computer with N cubits can process at least 2^n states at the same time.

So, for example –applying a reduce version of the travels man problem-, if we have $N=1,000,000$ possible routes from Bogotá, Colombia to Lima, where Peru is the fastest one. To find the optimum route, a computer must start by digitalizing 1,000,000 options. While Classical computer need to analyze the routes one by one, the quantum computer can consider all the routes at the same time. Classical computer needs $n/2$ steps (500.00 attempts) while Quantum computer can perform entire operation in only \sqrt{N} steps (1.000 attempts).

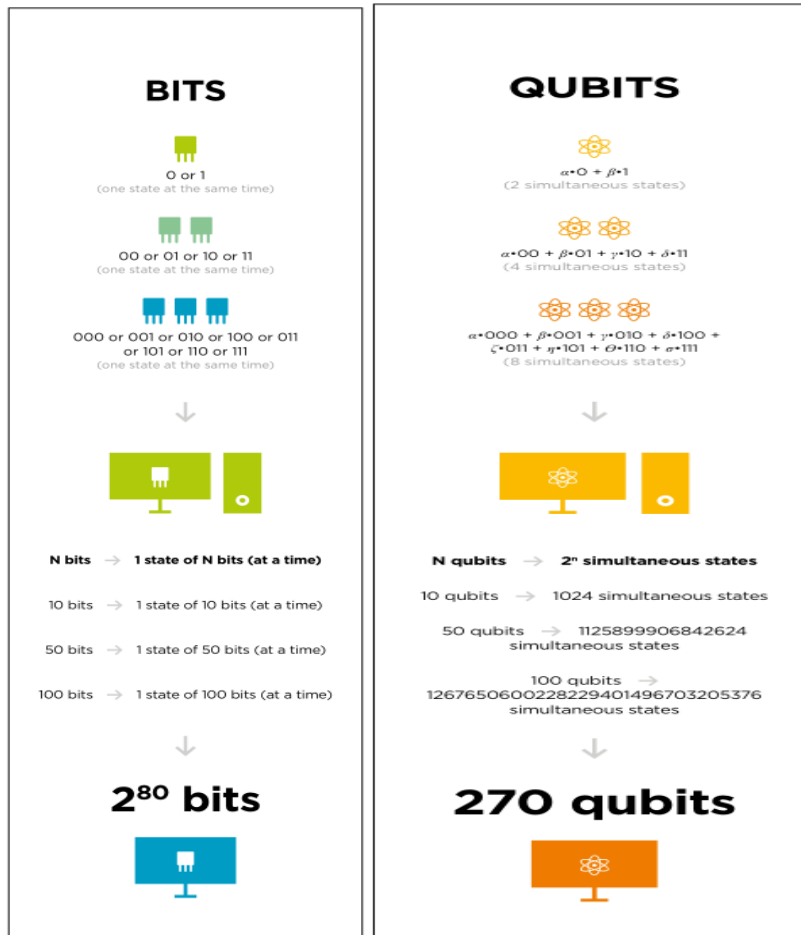


Fig. 1. Bits an qubits comparison

But, in this magic world, there exists challenges that must be solve in order to make real the use of all this power. Not everything is “easy”. Due to the inherent properties of the hardware and the kind of matter that must be controlled, the Quantum Decoherence Obstacle represent the loose of the Quantum properties when an experiment is run in a specific period of time. At the same time, and again due to the properties of the quantum world, we need to fight with Quantum errors. Energy, temperature, noise, gates, magnetic fields...even the inherent entropy of quantum world have a direct impact on Qubits behavior, its quality and its usefulness. The Error rate represents the operating time versus

decoherence time. If it is small enough it's possible to correct the errors. A lot of work must be done in this particular field.

The quantum computers can be implemented in different ways depending on subjacent technologies to obtain the qubits (Superconducting, Trapped ION, Optical Lattices, Majorana Fermion particles), purpose (general purpose, specific solutions) and types (topological, adiabatic, one way...).

The state of the art: quantum race to obtain a topological computer with lots of qubits and, of course, most of them quality qubits.

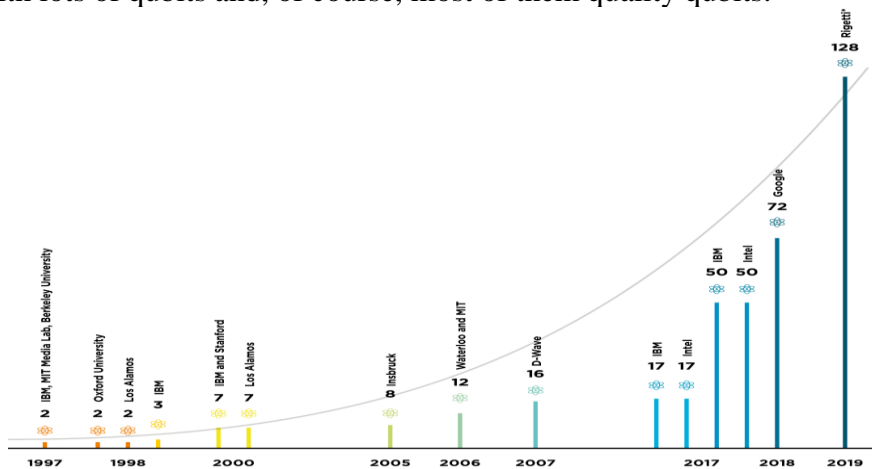
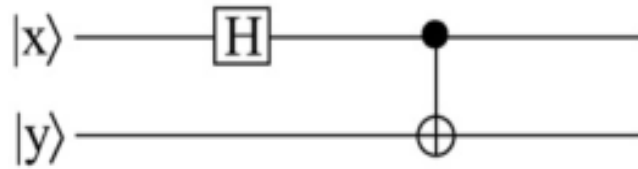


Fig. 2. Evolution of quantum computer power

Complex quantum computers, known as topological quantum computers, uses the concept of Quantum Gates. A collection of basic operations used to interact with the features of the qubits. In this kind of computers, quantum gates make possible the design, process and execution of circuits. These gates are a black box that encapsulate the complexity of the underground mathematics. The names of the gates: Identity, Hadamard (H Gate), Pauli-X, Pauli-Y, Pauli-Z, Square root of NOT (NOT), Phase Shift, SWAP, Square root of SWAP, Controlled (cX, cY, cZ), Toffoli (CCNOT) and CSWAP.

Using combinations of these gates, it's possible to have the quantum principles in our hands, for example:

- Hadamard gate give us the superposition effect
- And hadamard in combination with the CNOT using two qubits, give us the entanglement effect.



To explore the magic and power of the quantum computers, some brilliant companies offers to the engineers, distributed quantum platforms and tools that let them interact with the top quantum computers or powerful simulators available today. Examples of this amazing companies are IBM –with Quantum Experience-, Microsoft –with Quantum SDK and Azure- and Dwave –with ocean SDK-.

These amazing technologies give us the possibility to create quantum circuits or quantum programs -high level code using programming languages- that interacts with real machines or simulators through the software stack. Examples of that:

1) Using the quantum SDK of Microsoft to test the superposition:

```
operation BellTest (count : Int, initial: Result) : (Int, Int)
{
    mutable numOnes = 0;
    using (qubit = Qubit()) {
        for (test in 1..count) {
            Set (initial, qubit);
            H(qubit);
            let res = M (qubit);

            // Count the number of ones we saw:
            if (res == One) {
                set numOnes += 1;
            }
        }
        Set (Zero, qubit);
    }

    // Return number of times we saw a |0> and number of
    times we saw a |1>
    return (count-numOnes, numOnes);
}
```

2) Using the Q Composer of IBM or its Python alternative to test the entanglement:

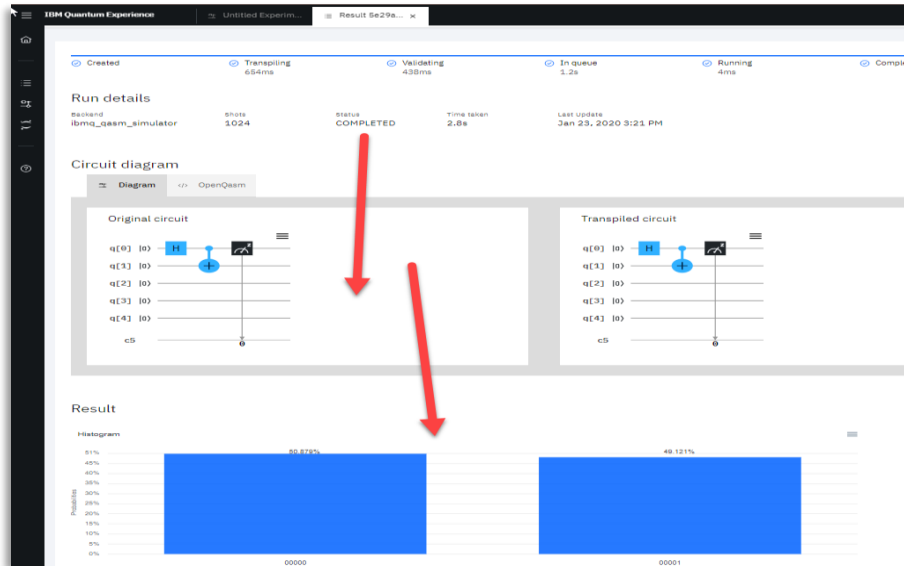


Fig. 3. Example of quantum program execution

```
# Load account data from Opened Session Info
provider = IBMQ.load_account()

# Create a 2 QuBits and 1 Classical Bit -collapse measure-
Quantum Circuit
qcBell = QuantumCircuit(2, 1)
qcBell.h(0)
qcBell.cx(0, 1)
qcBell.measure(0,0)
qcBell.draw()

# Prepare the execution and get the results
job = execute(qcBell,
              backend=Aer.get_backend('qasm_simulator'),
              parameter_binds=[])

counts = [job.result().get_counts(i) for i in range(len(job.re-
result().results))]

print (counts)
```

In both cases, we have the opportunity to create, compile, execute and debug the code. And experiment all the amazing possibilities this incredible technology can bring us in the future... today.
Welcome to the Quantum new era.

References

1. Jose Luis Hevia: Quantum Computing Fundamentals, Course Materials. Alhambra Systems & M2i Formacion. URL: <https://www.m2iformacion.com/Cursos/DetalleCurso.aspx?idcurso=M2iESPQ001>
2. Feynman, R. P. "Simulating Physics with Computers" International Journal of Theoretical Physics, Vol. 21 (1982) pp. 467-488.
3. Nielsen, M. y Chuang, I. "Quantum Computation and Quantum Information" Cambridge University Press (September, 2000), ISBN 0-521-63503-9. Author, F.: Contribution title. In: 9th International Proceedings on Proceedings, pp. 1-2. Publisher, Location (2010).
4. Simon Benjamin: The Dawn of Quantum Technology. Oxford Quantum for Oxford University. URL: <https://www.youtube.com/watch?v=7wThe1Q95Ro> from <http://oxfordquantum.org>.
5. Vladimir-Silva: Practical Quantum Computing for Developers (Apress-2019)
6. Eleanor Rieffel: An Introduction to Quantum Computing for Non-Physicists (FX Palo Alto Labs)
7. O'Reilly: Programming Quantum Computers Essential Algorithms and Code Samples