Lat. Am. J. Sci. Educ. 6, 22030 (2019)

Latin American Journal of Science Education

www.lajse.org

Quantum Computing for Undergraduates: a true STEAM case

César B. Cevallos[a], Manuel Álvarez Alvarado[b], and Celso L. Ladera[c]

[a] Instituto de Ciencias Básicas, Dpto. de Física, Universidad Técnica de Manabí, Portoviejo, Provincia de Manabí, Ecuador

[b] Facultad de Ingeniería en Electricidad y Computación, Escuela Politécnica del Litoral, Guayaquil. Ecuador

[c] Departamento de Física, Universidad Simón Bolívar, Valle de Sartenejas, Caracas 1089, Venezuela

A R T I C L E I N F O

Received: Agosto 15, 2019
Accepted: September 20, 2019
Available on-line: Junio 6, 2019

Keywords: STEAM, quantum computer, engineering curriculum, physics curriculum.

E-mail addresses:
cesarbernace@hotmail.com
manuel.alvarez.alvarado@gmail.com
clladera@usb.ve

ISSN 2007-9842

A B S T R A C T

The first quantum computers of 5-20 superconducting qubits are now available for free through the Cloud for anyone who wants to implement arrays of logical gates, and eventually to program advanced computer algorithms. The latter to be eventually used in solving Combinatorial Optimization problems, in Cryptography or for cracking complex computational chemistry problems, which cannot be either programmed or solved using classical computers based on present semiconductor electronics. Moreover, Quantum Advantage, i.e. computational power beyond that of conventional computers seems to be within our reach in less than one year from now (June 2018). It does seem unlikely that these new fast computers, based on quantum mechanics and superconducting technology, will ever become laptop-like. Yet, within a decade or less, their physics, technology and programming will forcefully become part, of the undergraduate curriculae of Physics, Electronics, Material Science, and Computer Science: it is indeed a subject that embraces Sciences, Advanced Technologies and even Art e.g. quantum music. This paper presents a model of a quantum computer, which describes its actual construction. Moreover, we illustrate with simple exercises and problems, and even application to music, for introducing undergraduates to the present quantum computing revolution. It is assumed that such students will have approved at least a course on Modern Physics and that also are familiar with Linear Algebra, particularly with the algebra of Vector Spaces. Concepts such as quantum entanglement, quantum decoherence, polynomial time, exponential calculation, charge qubit, flux qubit and phase qubit are also introduced.

Las primeras computadoras cuánticas de 5-20 qubits superconductores ahora están disponibles de forma gratuita a través de la *Nube* para cualquiera que quiera implementar matrices de puertas lógicas y, eventualmente, para programar algoritmos informáticos avanzados. Este último se utilizará finalmente para resolver problemas de optimización combinatoria, en criptografía o para descifrar problemas complejos de química computacional, que no pueden ni programarse ni resolverse utilizando computadoras clásicas basadas en la electrónica de semiconductores actual. Además, Quantum Advantage, es decir, el poder computacional más allá de las computadoras convencionales parece estar a nuestro alcance en menos de un año a partir de ahora (junio de 2018). Parece poco probable que estas nuevas computadoras rápidas, basadas en la mecánica cuántica y la tecnología superconductora, se conviertan en computadoras portátiles. Sin embargo, dentro de una década o menos, su física, tecnología y programación forzosamente formarán parte de los planes de estudios de pregrado de Física, Electrónica, Ciencia de Materiales y Ciencias de la Computación: de hecho es una asignatura que abarca Ciencias, Tecnologías Avanzadas e incluso el Arte por ejemplo música cuántica. Este documento presenta un modelo de una computadora cuántica, y describe su construcción real. Además, ilustramos con ejercicios y problemas simples, e incluso aplicaciones a la música, para presentar a los estudiantes de pregrado a la revolución actual de la computación cuántica. Se supone que dichos estudiantes habrán aprobado al menos un curso de Física Moderna y que también están familiarizados con el Algebra Lineal, particularmente con el álgebra de Vector Spaces. También se introducen conceptos como entrlazamiento cuántico, decoherencia cuántica, tiempo polinomial, cálculo exponencial, qubit de carga, qubit de flujo y qubit de fase.

I. INTRODUCTION

The present era is being shaken by an ongoing *quantum computing revolution* initiated a decade ago. Quantum computers in the form of sets of superconductor chips, e.g. the *IBM Quantum Experience* computers are available as a *Cloud* free service for anyone to do complex calculations with very good accuracy in truly short times, or to simulate quantum systems phenomenae. Quantum computing is here to stay, and it has already begun to transform our ways of doing physics, engineering, mathematics and chemistry, and above all to allow us to solve key problems that will never be solved using classical computers. It is foreseeable that within a decade quantum computers will be also impacting basic human activities, and perhaps ordinary life, all across the world. It has thus become of utmost importance that we begin to pay attention to the problem of how to incorporate the know-how, both fundamentals and technology of quantum computing, into our undergraduate curriculae of physics, electronics, computer science, physics engineering and material science. Quantum computing is a challenging and fascinating subject that is now within the reach of our science and technology undergraduate students.

What we need these quantum computers for? After all, in the last thirty years powerful conventional computers, Turing`s machine like, have been built that are capable of crunching sophisticated numerical problems, or solving complex computational problems in tens of hours. Moreover, today modest laptop computers are times faster than the large mainframe computers of 1950-1970. However, there are very important problems that we need to solve which no present conventional computer, built using highly integrated micro-electronics circuits made of doped semi-conductors, will be ever capable to solve. Each *bit* in a conventional computer can assume just two values, usually denoted 0 and 1, which are then cleverly exploited to codify and process information using many of them (e.g. megabits).. In a quantum computer, as will be explained in Section 2, their so-called *qubits* can instead assume not two but a truly large number of basis quantum states, and when properly constructed and programmed a small number of qubits can carry multiple computations simultaneously! *i.e.* massive parallelism, allowing us to design quantum computation algorithms that work at very high speed, or ones that can solve very complex problems with an accuracy never achieved before.

For instance, the search of a particular item in a large and unsorted database, i.e. locating a given phone number in the large telephone directory of a city can take a very long time, even if a very fast conventional computer is used. It is a problem somehow analogous to searching for a needle in a haystack, except that we would be searching for a particular needle (the phone number) in a large pile of needles of different shapes (the massive data in the directory). L. K. Grover showed in 1996 that the time to solve these problems could be drastically reduced using a quantum computer and his now well-known Grover search algorithm (Grover L. K., 1996). Another problem that we could finally solve using a quantum computer is related to our food security. It consists in deciphering the complex quantum electronic structure of the *Nitrogenase* molecule, and its chemical properties. *Nitrogenase* is the enzyme used by bacteria in Nature to reduce nitrogen (N_2) to ammonia (NH_3) to produce fertilizers. The industrial process presently used to reduce N_2 to NH_3 requires high temperature, and consumes 2% of our global energy production! Specialized bacteria use instead the *Nitrogenase* to do the same in Nature *at standard temperature and pressure*! with very low consumption of energy. Let us recall that nitrogen fixation is required for the biosynthesis of essential biomolecules, e.g. aminoacids in plants, from which animals and humans feed. We now understand that only a computer built with quantum objects (e.g. electrons, photons) would be capable of unfolding the chemical bonds of very complex molecules such as that of *Nitrogenase*, allowing us to understand their chemical interactions, and therefore to produce fertilizers more efficiently by using quantum control. Such kind of molecular physics knowledge will lead us to discover new medicines to cure illnesses such as cancer, or even to understand the complex molecules interactions that are the keys to the origin of life.

A third standing problem that justifies our need of a quantum computer is the frequently mentioned problem of factoring a big integer number using a computer, since it is a problem related to the confidentiality, authenticity, and integrity of personal, financial and commercial data. It is indeed a critical issue for the security –through the process of *encryption* – of the digitalized data transported on Internet, or between entities at any institution or .

Government's agencies. To understand this case for quantum computers, let us begin defining what is a *polynomial time* or *P-class* problem. A problem with an input of size n is said to be *solvable in polynomial time* if the time t taken to solve it is given by $t(n,k)= O(n^k)$, k being a non-negative integer. Let us give an example: suppose that you are given an n-digits number, and to solve a problem related to it takes $t(n)=10n^{100}-5\,n^{50}$ steps, therefore yours is a *polynomial time* problem. Finding the first 1000 digits of π is another example of a polynomial time problem, and can be done with a classical computer in less than an hour. Yet, we do know that it took *two years* to a team of scientists, using a *few hundred* computing workstations, to factorize a 232-digit integer in 2009.! It is of utmost importance that encryption of data be based on a problem that cannot be solved in polynomial time, yet a general purpose algorithm is still unknown. On the other it has been shown that factorizing a *composite integer C*, i.e. one that is the product of two prime integers P_1 and P_2, can be done in truly short time on a quantum computer running the Shor algorithm (Shor, 1994), the most recent largest example being 4088459=2017×2027, and was done using an *IBM Quantum Experience* (IBM Quantum Experience, 2017) computer of just a few qubits. The factorization problem is not a polynomial time one, and it takes the exponential time $t{\sim}exp[O(n^{273} \log^{2/3} n)]$ to be solved in ~ 3×10^{22} years with a fast classical computer. In comparison, it can be solved in only ~100 seconds if a few qubits quantum computer and the quantum Shor algorithm are used, since it now becomes a polynomial problem of order $t{\sim}O(n^3)$.

It is currently accepted that the field of quantum computing was born in 1981-82 in the hands of Richard Feynman, in his seminal *Simulating Physics with Computers* paper (Feynman, Int. J. Theor. Phys, 1982). Although Feynman then acknowledged the contribution of some predecessors, there is no doubt he was the first scientist to discuss from a foundational point of view the simulation of a quantum system with a computer. Feynman was first stating that only a computer based on quantum mechanics and built with quantum components could be used to simulate a quantum system, and eventually our physical world. No Turing machine can simulate such quantum systems. He went even further to suggest the possible inter-simulatability of the two quantum systems i.e. the quantum computer and the quantum system being simulated. In his paper and lectures of 1981-82 he already discussed the idea of a quantum universal simulator, one that by using quantum effects could be used to explore complex quantum phenomenae and run simulations of them.

The first quantum computer ever built (1998) was based on magnetic resonance of nuclear spins, and it had only two bits. It is interesting to mention here that by 1970 IBM had already initiated its research program on superconducting **Josephson junctions (JJ´s)** at very low temperatures (see Section 4) having in mind building a quantum computer based on such junctions. Surprisingly, on September 23, 1983 IBM announced the cancellation of that remarkable research program after more than a decade of work by its scientists and engineers, and tens of million dollars spent on it. It is a bit of an irony that thirty two years later (2016) IBM developed its first quantum computers, giving scientists and students at large the great opportunity of learning to use their *IBM Quantum Experience* (IBM Quantum Experience, 2017) quantum computers that happen to be built with precisely superconducting JJ´s !

The main objective of this work is to pave the way for the introduction of the fundamentals, and the most basic blocks, of the quantum computing revolution in the undergraduate curriculae. In Section 2 we shall be presenting the *qubit* concept and the two fundamental principles of quantum mechanics systems that allow quantum computers to be fast, more powerful and more accurate than classical computers, namely the *coherent superposition of quantum states* and the *entanglement of such states*. Section 3 will be devoted to a brief presentation of some of the Boolean gates used in any quantum computer, and to the concept of *quantum circuit* or *quantum gate array*, as well as to examples of actual computing with quantum gates. Section 4 is devoted to a brief presentation of low temperature superconductivity and to the Josephson Effect, and to a description of superconducting qubits, and of how they are actually made. Niobium and aluminium JJ´s qubits will be then described. Section 5 is devoted and to a brief presentation of the Grover, and the Shor algorithms. A short final sub-section of Section 5 is devoted to the application of quantum computing to the art of music composition. Section 6 is devoted to Discussion and Conclusions. The more comprehensive, and probably the best textbook on *Quantum Computing and Quantum Information* is the well-known book written a decade ago by Nielsen and Chuang (Chuang, 2000) which we highly recommend to Instructors and students.

II. CLASSICAL BITS AND QUANTUM BITS (QUBITS)

A conventional or classical computer is built from a myriad of tiny transistors and electronic components integrated in silicon chips (50 million transistors in a Pentium IV processor!) that digitalize and process information in classical bits. They codify and process data in a two-voltage values design, say 0 and 5 V. A *quantum computer* uses instead *quantum objects*, *e.g.* electrons in atoms, atoms nuclei, or even photons, whose states are usually represented in terms of their quantum *spins*, or in the case of photons represented by their polarization states, but never as the convened pair of voltage values handled by the circuits in a conventional computer. The *spin* of a quantum object is an intrinsic magnetic dipole that can assume two preferred directions *up* and *down*, when the object is placed in an external magnetic field **B** (Laloë, 1977), (Griffiths, 2016), Here *up* means that the spin points along **B**, *down* means the spin is anti-parallel to **B**. These two states are called *a pair of eigenstates* or *proper* basis states in quantum mechanics. One may represent these basis states with the vector symbols {|0>, |1>}, and sometimes as {|↑>, |↓>}. The matrix representation of the vectors of the *standard* or quantum *computational* basis is the following:

$$\{|0\rangle, |1\rangle\} = \{\begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}\} \quad (1)$$

Each of these binary quantum objects is known as a *quantum bit*, or a *qubit*, since they resemble the *classical* bits that assume the discrete values (0,1) and are used for codifying and processing information in a classical computer. But there is a fundamental difference between *classical bits* and *quantum bits*: bits can only assume their two discrete values, either 0 or 1, while in the case of an electron, a neutron or a proton, their spin state |s> could be prepared in a linear *coherent superposition* of the two eigenstates (Guilherme Tosi, 2017), namely |s> = a |0> + b |1>, *a*, *b* being *complex numbers* that fulfils the condition $|a|^2 + |b|^2 = 1$. Qubits can therefore assume a multiplicity of states, and a quantum computer may use all those states, not just the two states of a classical bit. In presence of an external magnetic field the magnetic dipole of an electron aligns parallel or anti-parallel to the field, i.e. assume states |0> or |1>, acquiring energies E_2 and E_1 respectively. If in its state |0> of lower energy E_1 the electron is irradiated by an electromagnetic wave of the proper frequency ν given by the well-known Bohr relation $\nu = (E_2 - E_1)/h$, the electron will flip to its other eigenstate |1>. By controlling this interaction, the electron may end up in the balanced coherent superposition state $\frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle)$.

It is important to stress that in the coherent superposition |s> = *a* |0> + *b* |1> the modulus-squared $|a|^2$, $|b|^2$ represent the *probabilities* of finding the spin in either of the two eigenstates |0> and |1> when the spin of the electron be measured (Guilherme Tosi, 2017). Let us write an example: Let the electron spin be $|s\rangle = \sin 30^0\, |0\rangle + \cos 30^0\, \mathbf{i}\, |1\rangle$. This means that out of a 100 measurements of its spin, it will be found 25 times in the vector state |0>, since $(\sin 30^0)^2 = (½)^2 = 0.25$, and the remaining 75 times in the vector state |1> . Yet, these results represent only one example of the many possible linear combinations that the spin of the electron may assume, as the superposition coefficients *a*, *b* assume different values.

In presence of an external magnetic field the magnetic dipole of an electron aligns parallel (state |0>) or anti-parallel (state or |1>) to the field, acquiring potential energies E_2 and E_1, respectively. If in its state |0> of lower energy E_1, the electron is irradiated by an electromagnetic wave of proper frequency ν given by the well-known Bohr relation $\nu = (E_2 - E_1)/h$, the electron will then flip to upper energy eigenstate |1>. By controlling this interaction, the electron may end up in the balanced coherent superposition state $\frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle)$ of equal probability amplitudes $\left(\frac{1}{\sqrt{2}}\right)^2 = 0.5$.

A classical computer of n=5 bits represents just 2n= 32 (states) for processing. The basis of a quantum computer of 5 qubits is instead a set of 32 different eigenstates, so that each pure quantum state is a vector that belongs to a Vector Space of dimension 2n= 32, i.e. it is a linear combination of the 32 basis vectors, whose 32 coefficients are *assigned by probability*: A register of *n* qubits does provide an exponentially larger processing power than *n* bits. A quantum computer with 32 qubits has over 4×109 basis states available to it. This makes quantum computers to be massively parallel, i.e. for processing information along simultaneous parallel tracks. Undergraduate students that have approved standard Modern Physics courses, and their Instructors, are aware that Quantum Mechanics accounts for the time-space representation of objects such as electrons, atoms or protons, in terms of complex wave-functions whose modulus-squared integral has value 1 (Laloë, 1977), (Griffiths, 2016), (Beiser, 2006), i.e, their mathematical representation belongs to a so-called vector Hilbert Space (Laloë, 1977), (Griffiths, 2016).

The other attribute of quantum qubits that serves to the purpose of constructing a quantum computer is called *entanglement*. It may be interpreted as a kind of intrinsic correlation between quantum objects. It is a pure *quantum effect* that was already known by 1935 to E. Schrödinger, the father of Quantum Mechanics, Entanglement does not occur in our macroscopic classical world. It is a quantum correlation that surprisingly we did not begin to exploit until very recently (since ~1990). *Entanglement* occurs between two or more quantum objects, say two electrons, and it is indeed a remarkable concept of the quantum world. A simple way to understand entanglement is to consider the imaginary case of two *entangled* twins C and M so "intrinsically connected" that when far apart one from the other - say one in Guayaquil and the other in London- and if C begins to rotate clockwise at *t=12.00* h, then its far away twin M will ***instantly*** start rotating anticlockwise by itself, i.e. exactly at *t=12.00* h, *with no communication whatsoever between them*. Of course, this sort of instant correlation without communication *never occurs in our classical macroscopic world*. Yet, in the world of quantum objects (atoms, electrons, molecules, protons) entanglement frequently occurs. There is a single condition for two quantum objects to become *entangled*, say two electrons A and B: *they must have been generated or coupled together at some initial time* t_0. After t_0, and *if not perturbed, they will be entangled forever, no matter how far apart A will be from B*. Therefore, if we measure in our lab a certain property of electron *A*, say its spin state, then we will be able to assert with 100% certainty what the spin state of the B electron is, *without doing any measurement at all* where B is located. Albert Einstein called this bizarre phenomenon *spooky action at a distance*. Two important questions now emerge: How quantum computing exploits quantum entanglement? How important is quantum entanglement for quantum computing?

An entangled qubit quantum state is also represented as a superposition of states of different qubits, but not as a *linear coherent superposition* mentioned a few paragraphs above, but one that **cannot be factored, i.e. written, as a product of individual states.** An example of a factorizable quantum state is the following:

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |11\rangle) = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \times |1\rangle \quad (2)$$

Consider instead the following four examples of entangled quantum states, known as the set of *Bell states*:

$$|\psi_1\rangle = \frac{1}{\sqrt{2}}(|00\rangle \pm |11\rangle), \quad |\psi_2\rangle = \frac{1}{\sqrt{2}}(|01\rangle \pm |10\rangle) \quad (3)$$

These are entangled states of the singlet vectors $|0\rangle$, $|1\rangle$ and cannot be factored as in (2). Tbus, entangling the qubits of a quantum computer will enlarge the number of quantum states available for computer tasks. Quantum entanglement therefore provides further opportunities for the massive parallel processing power that quantum computers offer.

Finally, a couple of critical issues about qubits need to be presented: the extreme fragility of qubits states and their *decoherence*. Present quantum computers, such as those of *IBM* Quantum Experience (IBM Quantum Experience, 2017), or the promising qubit arrays built with phosphor atoms at UNSW (Australia) are extremely fragile physical systems. This means, for instance that their states can be easily altered, say from |*0*> to |*1*>. This is one of the reasons for these qubits to be kept at temperatures that are truly low e.g.. at about 15 mK: And this is not only for maintaining superconductivity, but also to avoid the transfer of heat from the environment, e.g. through the connecting wires of qubits. Qubits states can also be perturbed by weak stray electric and weak magnetic fields, or by the presence of undesirable isotope atoms such as atoms of Si-29 in the Si-wafer platform they are implanted. Because of qubits fragility the building of a quantum computer demands large investment not only of money but also of research time and efforts. We have mentioned above that the qubits in a quantum computer are in coherent superposition states which mean that there exists some constant phase relation between the qubits basis states. Because of undesirable interactions such coherence may degrade in a period of time known as the *decoherence time*. For the present JJ's embedded in silicon wafers, the decoherence time is about 500 μs which is considered an eternity for doing quantum computations. Decoherence means that over time the quantum computer gradually loses its "quantumness" and becomes more like a classical object.

III. QUANTUM LOGIC GATES

Classical computers process code data using Boolean logical gates. e.g. AND, OR gates, which can be *reversible* or not. Quantum computers do the same, using equivalent quantum *logic gates* which act on *input qubits* and *are always reversible*, **both physically and logically** (Chuang, 2000). Ever since the seminal works of Feynman (Feynman, Int. J. Theor. Phys, 1982) *reversibility* of a gate has always been an important issue in quantum computation: it means that the output states are uniquely determined by the input states. The heat dissipated during any physical process, either quantum or classical, is usually taken to be a sign of physical *irreversibility*, and for them the microscopic physical state of the system cannot be restored exactly as it was before the process took place. Reversible gates are represented by *unitary operators* in the conceptual sense defined in elementary Linear Algebra and undergraduate Quantum Mechanics textbooks (Laloë, 1977), (Griffiths, 2016). A simple example of an irreversible gate is the NAND. A given quantum gate acts on its set of input qubits in the sense that they *perturb* or modify their initial states leaving the qubits in other states. The symbol for a quantum gate is a box, input and output qubits are represented by straight lines to the left and right of the box, respectively (Fig. 1): output lines must equal in number the input lines

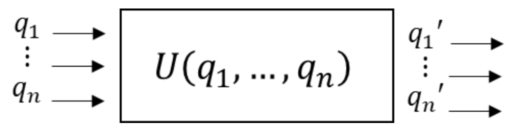


FIGURE. 1 Sketch of a unitary quantum gate operating on *n* qubits of input sates $q_1 \ldots q_n$ and output states $q'_1 \ldots q'_n$. U represents the unitary operation performed by the gate.

.

From the mathematical point of view gates are represented by *unitary operators* written as square matrices. The identity gate I produces output quantum states identical to its input states, and it is represented by the Identity square matrix. Thus, in the 2-dim standard computational basis the identity is represented by the 2×2 matrix operator:

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{4}$$

III.1 One Qubit Gates: the X, Pauli-Z and Hadamard gates

The most common quantum gates act either on one qubit (*unary gates*) or two qubits (*binary gates*). One can define infinite one-qubit gates, but the most used in quantum computing include: the *Pauli-Z*, the *T*, the NOT or X, and the Hadamard gates. The symbols for the last two are shown in Fig. 2,

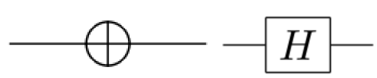


Fig. 2 Symbols used for the quantum NOT and Hadamard gates.

One-qubit gates operate on single 2-dim qubits, hence these gates are represented by 2×2 matrices. For instance the NOT or X gate, that changes the input state |0> into |1>, and the input|1> into |0> are represented by:

$$X \equiv NOT = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \Longrightarrow X \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}; X \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}; \tag{5}$$

:

The *Pauli-Z gate* gate, sometimes called the *phase flip* gate, is represented by the unitary operator:

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \Longrightarrow Z \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} a \\ -b \end{pmatrix} \tag{6}$$

i.e. it transform a superposition input qubit state $|s\rangle = a\,|0\rangle + b\,|1\rangle$ into the output state $|s'\rangle = a\,|0\rangle - b\,|1\rangle$.

Undoubtedly, the most important *one-qubit* quantum gate is the ***Hadamard***, represented with a capital *H* in its box. It is mathematically represented by the following unitary operator *H*:

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{7}$$

When this gate operates on either of the qubit of the computational basis {|0>, |1>} the result is a balanced coherent superposition of the two basis state. This is shown below for the input basis state $|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$:

$$H|0\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \tag{8}$$

This is a good opportune to check the *reversibility* of the Hadamard quantum gate. In effect, applying its matrix H to the output state $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ obtained above one recovers the state $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ as it is shown below:

$$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\left\{\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right\} = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} = |0\rangle \tag{9}$$

Another important one qubit gate is the ***Phase Gate.*** It is again a *one-qubit gate* that introduces a phase shift $e^{i\varphi}$ between the two matrix elements of the qubit. We may represent it with the symbol P_φ and its matrix is:

$$P_\varphi = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\varphi} \end{bmatrix} \tag{10}$$

which applied to a qubit in the general state $|\psi\rangle = \begin{bmatrix} a \\ b \end{bmatrix}$ gives the expected result:

$$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\varphi} \end{bmatrix}\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} a \\ e^{i\varphi} b \end{bmatrix} \tag{11}$$

Thus

$$P_\varphi|0\rangle = P_\varphi\begin{bmatrix} 1 \\ 0 \end{bmatrix} = |0\rangle, \quad \text{and} \quad P_\varphi|1\rangle = P_\varphi\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ e^{i\varphi} \end{bmatrix} = e^{i\varphi}|1\rangle \tag{12}$$

Yet, another important one-qubit gate is the ***T-gate*** which is the particular phase gate case for ***$\varphi=\pi/4$*** (also known as "π/8" gate, which no doubt is a bit confusing, so better call it ***T***)

$$T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix} \tag{13}$$

III.2 Control-Not Gate (or XOR Gate)

The *Control-Not*, or XOR gate (Fig. 3) is a very important two-qubit gate frequently used in quantum algorithms. In the figure the upper qubit |*A*> is called the *control qubit*, while the lower one |*B*> is the *target qubit*. The action of C-NOT is defined as follows: if the *control qubit* is |*0*> then the lower qubit is not changed, and the output from the gate is then equal to the input pair |*A*>, |*B*>; but if the *control qubit* is |*1*> then the NOT matrix is applied to the target qubit. The symbol used for this two-qubit gate is therefore:

Operating on two qubits the operator representing a C-NOT is of course a 4 ×4 matrix. Moreover, since it operates on two qubits its standard input states are products of the two single entries of the computational basis: {$|00\rangle$, $|01\rangle$, $|10\rangle$, $|11\rangle$}, and the respective output states are according to the definition of this gate given: $|00\rangle \rightarrow 00\rangle$,$|01\rangle \rightarrow |01\rangle$, $|10\rangle \rightarrow |11\rangle$, and $|11\rangle \rightarrow |10\rangle$. Therefore the 4×4 matrix of C–NOT (or XOR) is:

$$C - NOT \equiv XOR = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{15}$$

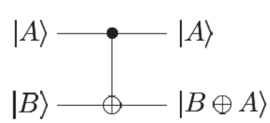


Fig. 3 Symbol and scheme of input and output qubits of the C–NOT gate.

One of the striking applications of the C-NOT gate is to generate useful entangled states, as shown in the following entangling example:

$$C-NOT\,[\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)\times|0\rangle]=C-NOT\,[\frac{1}{\sqrt{2}}(|00\rangle+|10\rangle)]=\frac{1}{\sqrt{2}}(|00\rangle+|11\rangle)=|\beta_{00}\rangle \quad (16)$$

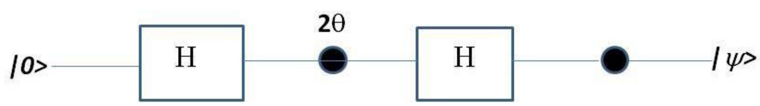
whose output state is $|\beta_{00}\rangle$ is in effect one of the four *entangled Bell states*, used in Quantum Information (Chuang, 2000).

III.3 Coherence time and *gate length*

The length of the time interval required for a gate operation is known as *gate length* and it is *~10-100 ns* (nanoseconds). Since the present coherence time of supercomputing qubits is *~500 μs* the number of possible computing operations of such qubits is of the order of 10000, which is sufficient for building a successful quantum computer.

III.4 Quantum circuits (quantum gate arrays)

The quantum circuit in Fig. 4 is a sequence of quantum transformations performed on a quantum register (*i.e.* on an array of coupled qubits or *quantum register*), and represents a sequence of unitary operations on the quantum register. Lines in it represent qubits in evolution and not physical wires. Quantum circuit diagrams are drawn with *time going from left to right*, with the quantum gates across one or more "wires" (qubits). Figure 4 [after 9] is a sequence of four gates H – $P_{2\theta}$ –H–$P_{(\varphi+\pi/2)}$, where the left Hadamard gate is applied first to the input qubit initiated in the state *|0>*. The output state of this circuit, denoted **|ψ>** in the figure, is the most general superposition of the basis states |*0*>,|*1*>.



$$[P_{\varphi+2\pi}HP_{2\theta}H]|0\rangle=|\psi\rangle=\cos\theta\,|0\rangle+e^{i\varphi}\sin\theta|1\rangle$$

Fig. 4 Example of quantum circuit and its operational representation (at the bottom).

III.5 Universal Set of Gates

It is clear that one can define any number of reversible quantum gates by defining their operators using unitary matrices, and to implement them with advance integrated circuit technology, plus a bit of luck. Fortunately quantum computer algorithms and processes can be implemented with just a few quantum gates. A minimum set satisfying such condition is called a ***Universal Set***. Probably the best example being the set of four gates: ***{Hadamard, CNOT, S phase gate, T or π/8 gate}***. It may be shown that any unitary operation can be efficiently approximated to arbitrary accuracy using the above set of gates (Chuang, 2000).

IV. SUPERCONDUCTING QUBITS AT VERY LOW TEMPERATURES

To build a quantum computer include four technological feats: (i) Fabricating the qubits; (ii) Capacity to isolate them so that the environment will not perturb their fragile quantum states i.e. avoid decoherence; (iii) The technology to initialize the qubits, to address them electro-magnetically to implement the gate operations, and to couple qubits to be to other qubits; (iv) The technology to measure the output signals i.e. "read-out" the qubit output states; Fortunately, these technologies have already been developed in the last 30 years. This section shall motivate Instructors and students to pay attention to the different quantum computing hardware technologies, ones that students may eventually consider as subjects for their future graduate studies.

IV.1 Superconductivity and quantum tunneling

Superconductivity of metals has been known for about 100 years. When cooled below certain very low critical temperatures, say 10 K, most metals lose their resistance to electron transport and become superconductors i.e. can sustain a constant electric current for practically infinite time. The required low temperatures were initially reached using liquid Helium, but nowadays cryogenic techniques exist that allow superconducting experiments at milli-Kelvin (mK) temperatures, as in the case of the *dilution tanks* at 15 mK (Fig.11) of *IBM Quantum Experience* computers (IBM Quantum Experience, 2017)

Quantum tunneling is a striking phenomenon familiar to undergraduate students that have taken *Modern Physics* courses [see Ch.5, Ref. 10]: *"Quantum objects, e.g. electrons or neutrons, without sufficient energy to pass over a potential energy barrier may still tunnel through it and appear to the other side of the barrier."* In 1962 B. D. Josephson discovered theoretically that it would be possible to observe the *quantum tunneling* phenomenon in a system of two superconducting metallic pieces separated by a very thin layer of an insulator such as an oxide of the metal e.g. $Al{:}Al_2O_3{:}Al$. Short time after Josephson prediction its tunneling effect was experimentally confirmed. A scheme of a *Josehpson junction* is shown in Fig. 5 (a), where A, B are the metallic superconductors and C the thin insulating layer. The circuit symbol for such junctions is shown in Fig. 5 (b). An actual scanning electron microscope picture of a small JJ made of $Al{:}Al_2O_3{:}Al$ with present technology is shown in Fig.6; its dimensions are of the order of a few μm.

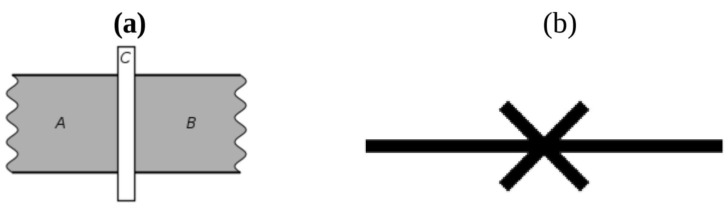


FIGURE. 5 (a) Diagram of a Josephson junction: A, B are the superconductors and C the insulating layer **(b)** Circuit symbol of Josephson junctions.

An excellent pedagogic presentation of superconductivity and Josephson junctions can be found in Vol. III (Ch. 21) of the well-known *Feynman Lectures on Physics* of Richard Feynman (Feynman, The Feynman Lectures on Physics III, 1963), which we strongly recommend to read. Superconductivity at very low temperatures *($T<1$ K)* was first successfully explained by Bardeen, Cooper and Schreiffer in their well-known *BCS Theory* (1957), in which they introduced the idea of *electrical conduction by pair of electrons*, instead of currents of free electrons.

IV.2 Cooper pairs of electrons and super currents in a metal

At temperatures near to 0 K the thermal oscillations of the atoms in a metal lattice practically cease (oscillations of very low energy $E=kT$, k= Boltzmann constant $\Rightarrow E\sim 10^{-25}$ Joule), and then free electrons in a metal are forced to couple in pairs called *Cooper electron pairs*. Super-electric currents represent flows of such Cooper Pairs of electrons travelling in the metal with zero resistance (the pair of electron behaves like two professional bicyclers riding as a team in a bike race: one behind the other alternatively, to reduce air drag effects).

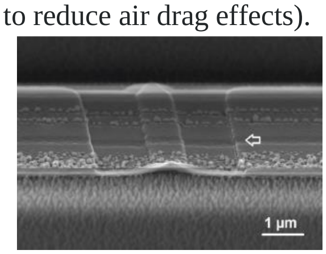
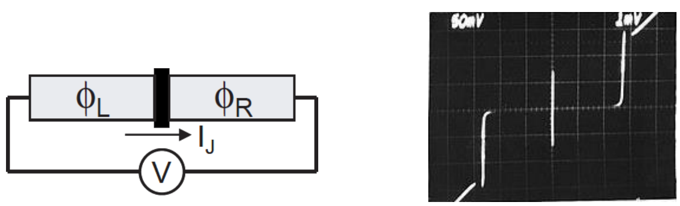


FIGURE. 6 A recent picture of the actual Josephson junction in a qubit taken with a Scanning Electron Microscope: the arrow points to a *trench* where the thin insulating layer of Al_2O_3 was deposited, the layers of *Al* are at the left and right of the trench (Phys. Proc. **36** 21 – 216 (2012).

Note that: (i) Zero resistance means *no dissipation* of heat energy in the cooled superconductor, (ii) On the other hand, should the temperature of the metal increase by just a small amount the increased thermal agitation of the metallic lattice uncoupled the Cooper pairs and superconductivity ceases. Superconductivity is a quantum phenomenon, and must be accounted for using Schrödinger´s equation (Feynman, The Feynman Lectures on Physics III, 1963). Moreover, currents of Cooper pairs or electrons in a superconductor must be represented by probability currents of **quantum wave functions** (Laloë, 1977), (Griffiths, 2016), (Feynman, The Feynman Lectures on Physics III, 1963) : *they are not like the currents in conventional circuits*.

Free electrons are *spin-1* objects and are therefore *fermions* (quantum objects of odd spin) and the well-known *Pauli Principle of Exclusion* (Laloë, 1977), (Griffiths, 2016), (Beiser, 2006), (Feynman, The Feynman Lectures on Physics III, 1963) dictates that no more than two of them (fermions) can coexists in the same quantum state. This means that *the two bound electrons in a Cooper pair should have opposite spins*, henceforth a total spin equal to zero. Cooper pairs are therefore *bosons*, and *there is no limit to the number of possible bosons in the same quantum state*. At near zero-Kelvin temperatures all the Cooper pairs in a superconductor metal are in the same ground state i.e. they form a Bose condensate. In its bound state Cooper pairs have a small energy gap but not have sufficient energy to interact with the metal lattice as they travel thru the superconductor without energy dissipation.

Figure 7(a) shows a Josephson junction driven by a voltage source V, and Fig. 7(b) shows an actual oscilloscope trace of its typical characteristic *I-vs-V* curve at low temperature. Note that **even at zero voltage there is a DC current thru the junction** (vertical line at the origin *V=0*). This current at zero voltage represents the tunneling of the Cooper pairs thru the insulating layer of the junction, and it is known as the ***DC Josephson Effect***, and given by the $\boldsymbol{I_c = I_0 \sin\phi}$, where $\boldsymbol{\phi = \phi_L - \phi_R}$ is the phase difference between the quantum wave-functions of the Cooper Pairs at the left and right sides of the junction in Fig. 7 (a). The constant $\boldsymbol{I_c}$ is the *critical-current parameter*, a small electrical current value that depends on the numbers of carriers at the sides of the union, and upon the thickness of the insulating layer. The tunneling of electrons across the junction gives the two current branches that appear to the left and right in the trace shown in Fig.7(b). When a **constant voltage V** is applied across a JJ an oscillating (AC) current appears in the junction, an effect which is called the ***AC Josephson Effect***. It is represented as $\boldsymbol{I(t) = I_C \sin(2\pi f_J\, t)}$, where the frequency $\boldsymbol{f_J}$ is proportional to V, and it is given by $f_J = \frac{2e}{h}$V. A typical value of $\boldsymbol{f_J}$ being 10^{15} Hz (for V~2 Volt)..



(a) **(b)**

Fig. 7 (a) A Josephson junction driven by a voltage source **(b)** Oscilloscope trace of the characteristic I-V curve of a Josephson junction. Even at zero voltage a DC current is measured which appears as the vertical line at the centre of the oscilloscope trace.

When a constant voltage V applied a Josephson junction behaves as a non-linear circuit component. In fact it may be shown that it can be modeled as a non-linear inductance of value $L_J = {\Phi_0}/{(2\pi I_c \cos\phi)}$ where $\boldsymbol{\Phi_0}$**=483.6 THz/V** is the ***Flux Quantum,*** an important superconductivity constant. A simple inspection of Fig. 7 (a) reveals that a Josephson junction also has a capacitance *C* associated to it. Therefore, this junction is a quantized non-linear LC-oscillator (or *resonator*) of multiple energy levels, of which the two lowest ones are used as a two-level system, analogous to a spin system: *this being the explanation of why a Josephson junction can be used as a qubit.*

IV.3 Charge, Flux and Current Josephson qubits. The transmon qubit

Depending on the form of biasing of a superconducting Josephson junction one may obtain three types of qubits*: Cooper Pair* or *Charge qubit*, the *Flux qubit*, and the *Current qubit* shown in Figs. 8 (a), (b) and (c), respectively. In the *charge qubit* a voltage source U_g is applied to the Josephson junction via a gate capacitor C_g, providing energy to the qubit. It's states are quantum charge states, and the two non-degenerate lowest energy states are used to define the computational basis of the qubit. This qubit configuration is sensitive to charge noise effects. In the flux qubit a superconducting transformer is used to provide energy instead of a capacitor. Figure 8(b) shows that an electrode of the junction of capacitance C is connected to the secondary of the transformer creating a loop of inductance L, through which magnetic flux is applied by the primary coil. Figure 8(c) shows the *phase* or DC-current qubit, the junction biased with a DC-current source, which provides a current value close to the critical current I_c. This biasing represents a large loop inductance, the circuit behaving as an anharmonic one. The mathematical models of these three superconducting anharmonic oscillators, based on Schrödinger equations, are out of the scope of this work (see Ref. 6).

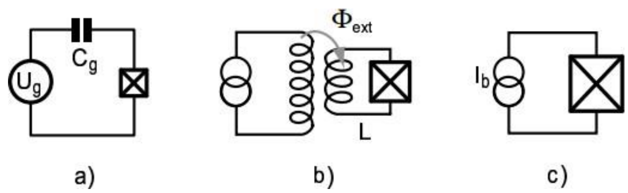


FIGURE. 8 (a) Charge qubit, **(b)** Flux qubit, **(c)** Phase qubit.

The *transmon qubit*, developed in 2007, is an improved version of the charge qubit. Its most simple sketch is shown in the Fig. 9, and can be compared to the sketch of the charge qubit in Fig. 9 (a). Note that apart from the gate capacitor there is a second shunting capacitor C_s connected to the junction electrodes. This simple addition diminishes the sensitivity to charge noise to very low values. Its name is an abbreviation of the term *transmission line shunted plasma oscillation qubit*. Its typical transition-frequency values are 5-5.4 GHz, with an anharmonicity of ~346 MHz, so that its charge dispersion is less than 30 KHz. It is the qubit circuit now being successfully used in IBM quantum computers freely available through the *Cloud* [3].

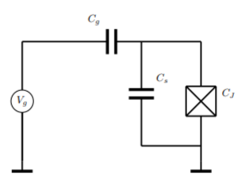


FIGURE 9. Circuit sketch of a *transmon qubit*. It is similar to the circuit of the charge qubit but with a shunting capacitor C_s connected to the junction.

IV.4 Quantum computers through the *Cloud*

At present (2018) there are several versions of quantum computers available for free through the *Cloud*. For the sake of space and simplicity we only consider here the simpler computers of the *IBM Quantum Experience* (IBM Quantum Experience, 2017). Figure 10 (left) shows the register of IBM `s four JJ-qubits clearly visible at the centre of the picture. The four radio-frequency control and readout lines of the four qubits can be distinguished. The wiggly coupling coupling resonators can also be distinghished. A scanning electron microscope of a JJ qubit is shown on the right of Fig. 10. These supercomputing qubits are quite large ~200 μm long, and are fabricated using niobium or alluminum (with thin insulating layers of their oxides).

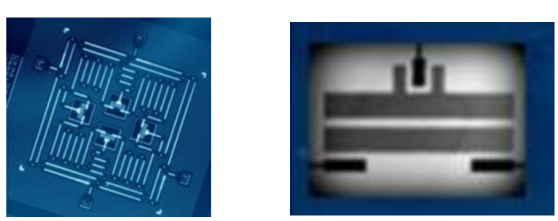


Fig. 10 Network of four qubits (left) of IBM Quantum Experience. An electron microscope picture of a JJ qubit (right).

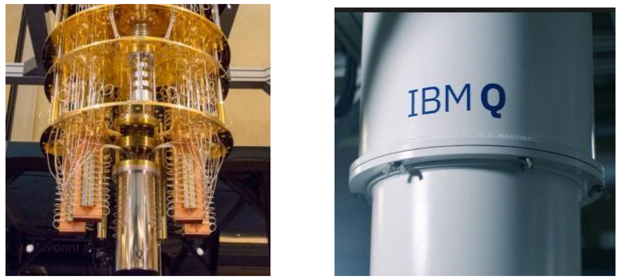


Fig. 11 A 50-qubits quantum *IBM Quantum Experience* [3] computer (left) and its cryogenic dilution tank (right).

Figure 11 is an impressive picture of an actual *IBM* Quantum Experience (IBM Quantum Experience, 2017) computer of 50 qubits located inside the golden canister at the bottom of the picture.The large set of coaxial cables that carry the radio-frequency pulses to drive the qubits (gate pulses) and carry out the readout signals are clearly visible. In operation this computer is placed inside the *cryogenic dilution* IBM-Q tank shown on the right of Fig. 11. The temperature at the top of the tank is ~4 K, and gradually goes down so that it is only ~10 mK inside the golden canister. The high quality coaxial cables used, and the cryogenic dilution tank warrants excellent isolation i.e. no decoherence from interaction with the environment. The readout signals are processed by electronic circuits based on FPGA (field programmable gate array) boards located outside the tank. As mentioned above classical computers are used to control and processing signals of the quantum computer.

V. QUANTUM ALGORITHMS

Oddly enough several of the most important quantum algorithms were invented decades before the present quantum computer technology, between 1980 and year 2010. A good description of the available algorithms can be read in (Knight, 2005). The best known are the Deutsch-Josza (1992), Shor (1995), and Grover (1996) algorithms. Another obviously important algorithm is the Quantum Fourier Transform algorithm, actually a quantum version of the discrete Fourier transform of a function. An up-to-date account of quantum algorithms can be found in (P. J. Coles, 2018).

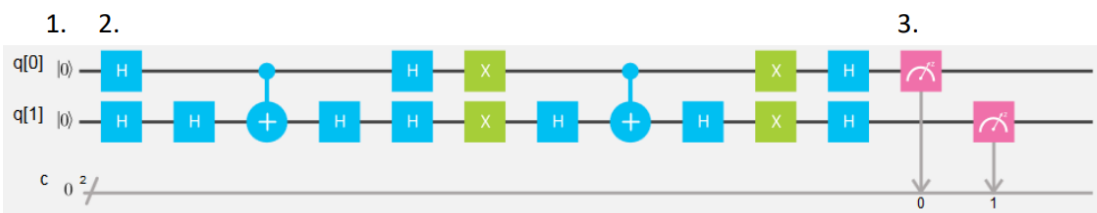


Figure 12: Stages of a typical quantum algorithm run on qubits q(0) and q(1): **(1)** Initialization of the two qubits to state *|0>*. **(2)** Sequence of *unitary operations* or *gates* applied to one qubit (Hadamard and X gates) or two qubits (two C-NOT gates). **(3)** Measurement (read-out) of the final states of the qubits. Repetitions of these three stages for statistical claims are necessary.

Most quantum algorithms are assembled in three stages: (i) Encoding of the input data, either classically or quantum mechanically, to define the input states of the qubits, (ii) Preparation of the sequence of quantum gates (unitary operations) to be applied to the input states, (iii) Measurements of one or more of the output states of the qubits to obtain a classically interpretable result, and multiple repetitions of the algorithm for statistical claims. Figure 12 shows a scheme of a quantum algorithm implementation. There are several issues to consider when implementing an algorithm on a quantum computer, namely: **1.** What is the set of available gates for the user to to implement its algorithm?; **2**. What physical gates are actually implemented?; **3.** What is the qubit connectivity or coupling (e.g. which are the pairs of qubits that two-qubit gates can be applied to?); **4**. What are the sources of noise (i.e. errors)? The two main sources of noise are typically *gate infidelity* and decoherence. Gate infidelity refers to the fact that the user-specified gate does not precisely correspond to the physically implemented gate. Perfect *fidelity* of a gate means that its output state is exactly equal to the programmer expected state. Gate infidelity is usually worse for multi-qubit gates than for one-qubit gates, so typically one opts for minimizing the number of multi-qubit gates in one's algorithm.

V.1 Grover's Algorithm

As written in the Introduction Grover's algorithm is a database searching algorithm. Following Grover himself (Grover L. K., 2001): "it is a technique for searching *N* possibilities in only $(\sqrt{N})$ steps". For instance, you know a phone number and wish to find the name of the person to which that number belongs in an unsorted telephone directory of *N* clients. Unless you are very lucky you had to look at least to an average of *N/2* entries before succeeding. Using Grover`s algorithm the number of times you had to consult the directory is reduced (Chuang, 2000) to the smaller number $(\pi/4\sqrt{N})$. Thus, if $N=1000000$, instead of searching the database *~500000* times you only have to search *785* times, which is great!. Grover's algorithm is based in the evaluation of a Boolean function *f(x)*, and in the application of Schrödinger equation (Grover L. K., 2001) to superposition quantum states, and it is out of the scope of this work. References (Chuang, 2000), (Grover L. K., 2001), (Candela, 2015) can be consulted for detailed explanations and applications. The number *n* of qubits required must be such that the database entries are $N=2^n$. After initializing the qubits in state |0> the algorithm uses Hadamard gates to generate superposition states. The main part of the algorithm is devoted to apply the product of two operations *R* and *D* to each state, *D* being known as the *Diffusion Transform*. A good account of these transformations can be found in Grover`s pedagogical paper (Grover L. K., 2001).

V.2 Factoring algorithm of Shor

As already explained in the Introduction Shor`s quantum factoring algorithm is applied to factorize a large composite number *C*, a number that is the product of two unknown prime numbers P_1 and P_2. Good enough only one of the steps of this important algorithm requires the use of a quantum computer; the initial and final steps of the algorithm

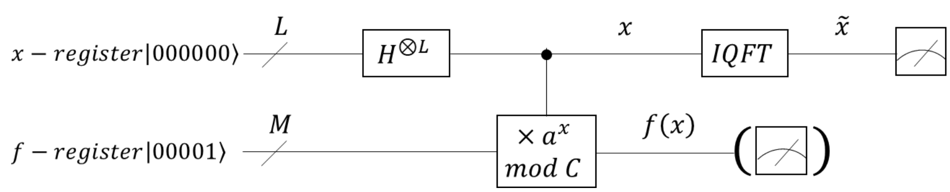


FIGURE. 13 A quantum circuit for the Period Finding algorithm that lies at the hearth of Shor's algorithm. Note that two sets of qubits are required, *L* qubits at the top line for the *x* values, and *M* at the bottom line for the *f(x)* values (after (Candela, 2015))

can be done using a classical computer (Candela, 2015). It is a factorization algorithm with quantum polynomial complexity, and it would not very hard to learn by undergraduate students (Candela, 2015) (Styles, 2015). When using the best classical algorithm, the so-called *number field sieve*, the prime factorization of an *n*-bit integer requires $exp(O(n^{1/3} log^{2/3} n))$ operations. Using the Shor quantum algorithm the order of operations is exponentially smaller: $O(n^2 \log n \log \log n)$.

As explained by Candela (2015) two well-known elementary-school arithmetic concepts are required for Shor´s algorithm: **(i)** the concept of *greatest common divisor* of two integers g and p, i.e. *gcd(p,q)*, and **(ii)** the concept of the integer rest *q* obtained when *an integer p is divided by another integer C*. The latter is known as *modular congruence*, denoted $p\equiv q mod(C)$, which means: $(p-q)$ is a multiple of *C*. A sequence of steps for Shor's factorization algorithm of an integer *C* is: **(i)** First check that *C* is odd and not a power of some smaller integer; **(ii)** Choose any integer *a* in the open interval of integers *(1,C)*; **(iii)** Find the *gcd(a,C)*. If this *gcd* is greater than 1, then you have found a factor of *C* (precisely the *gcd*) and the factorization problem is solved; **(iv)** Find the smallest integer $p>1$ such that $a^p \equiv 1 mod(C)$. **this being the only step that demands using a quantum computer**: It is called the ***Period finding step;*** **(v)** if p is odd, or if p is even and $a^{p/2} \equiv -1 mod(C)$, go back to step (ii) and choose a different integer *a;* **(vi)** *t*he numbers $P_{\pm}=gcd(a^{p/2} \pm 1, C)$ are the sought nontrivial factors of *C*. *Period finding* is also a simple concept but its calculation is not an easy task. We are familiar with periodic functions *e.g*. $\cos x$, a periodic function of period 2π. Yet, finding the period *p* of an arbitrary periodic function *f*, i.e. $f(x+p)=f(x)$, is not easy at all. A possible solution is to evaluate *f(x)* at a large number of *x* values, hoping to discover what *p* is. However, if you use a quantum computer you can exploit its quantum parallelism to plot the function *f* for all those *x's* in a single operation. A complete account of Shor's algorithm, including *Period Finding*, is out of the scope of this work. It demands the application of the *quantum inverse Fourier transform* algorithm (Grover L. K., 2001), (Candela, 2015) in one of its steps. Figure 13 shows an example (Candela, 2015)of one of several available quantum circuits for *period finding*.Shor`s algorithm is not infallible (Styles, 2015) . Fortunately, there are ways for correcting the possible errors that may be successfully applied e.g. using the Continued Fractions Algorithm and the so-called Phase Estimation, as explained at the end of Ch. 5 of (Chuang, 2000).

V.3 Quantum music

Recently Putz and Svozil have suggested (Svozil, 2015) the quantum codification of just an octave of piano music in terms of quantum states, actually the eight tones c, d, e ,f, g, a and b and c´ that corresponds to eight consecutive white keys of the piano (C-major scale). Their simple idea was to assign to this list of musical tones the vectors of a Hilbert space of seven or eight dimension. With this idea they have opened a door for using a quantum computer to tackle the quantization of a complex score of a musical composition, say of Mozart or Prokoviev, a codification impossible to be treated with a classical computer, and more interestingly to a new and fantastic mode of writing and hearing the art of music: e.g. imagine yourself hearing an entangled melody of musical states!

VI. DISCUSSION AND CONCLUSIONS

In this work we have presented solid arguments for introducing quantum computing in our Science and Technology undergraduate curriculae e.g. electronics, electrical engineering, physics, physics engineering, computer science, chemistry. We have even mentioned that quantum computing is offering us a new and fantastic mode for music composition. Quantum computing is indeed within the reach of undergraduate students that have approved present Modern Physics and Linear Algebra courses; particularly if they have mastered Vector Spaces and the application of Schrödinger Equation to solve standard one-dimensional problems of introductory quantum mechanics. Quantum computers are here to stay, in spite of the modest number of qubits in present quantum registers, and even in spite of the lack of quantum memory chips. However, as it has been explained above, the latest 20-50 superconducting-qubits quantum computers of *IBM Quantum Experience* (IBM Quantum Experience, 2017) provide us with a formidable power of massive computing parallelism. Of course, there is still the requirement of keeping the quantum chips at the really extreme low working temperatures of *~10 mK*, and again present quantum computers need to be controlled, and their output processed, with classical computers. Since their birth, in the 1930´s to 1950´s, the achieved computer technologies advancements outspoke experts in their estimation of advancement (recall the case of *IBM* itself cancelling their 13 year of research on superconducting JJ´s in 1983); thus it is not unthinkable that a quantum computer based on high-temperature superconductors (say at ~77 K) may be developed in the future based on new quantum models (Ladera, 2017). We have made frequent references to the *IBM Quantum Experience* (IBM Quantum Experience, 2017) quantum computers available as a free *Cloud* service, that include well prepared User Guides. Undergraduate students and their Instructors are expected to be familiar with programming languages such as Python to exploit these quantum computers. Useful pedagogical papers such as (Candela, 2015) can be used for an undergraduate course on simulating a quantum computer, including examples and exercises. We have also presented brief introductions to the most important quantum algorithms, namely Grover´s and Shor´s. Prospective instructors may also consider teaching the important quantum Fourier Transform algorithm (Chuang, 2000) which is used as a step in the Grover´s one. An important subject in quantum computing is *Quantum Error-Correction* devoted to the ways of protecting information from noise (we recommend reading Ch.10 of the book by Nielsen and Chuang).

REFERENCES

Beiser, A. (2006). *Concepts of Modern Physics.* New York: McGraw-Hill. Retrieved from http://phy240.ahepl.org/Concepts_of_Modern_Physics_by_Beiser.pdf

Candela, D. (2015). Undergraduate computational physics projects on quantum computing. *American Journal of Physics*, 688-702. Retrieved from https://www.researchgate.net/publication/281979524_Undergraduate_computational_physics_projects_on_quantum_computing

Chuang, M. A. (2000). *Chuang Quantum Computation and Quantum Information.* Cambridge, United Kingdom : Cambridge U. P. Retrieved from http://www.michaelnielsen.org/qcqi/QINFO-book-nielsen-and-chuang-toc-and-chapter1-nov00.pdf

Feynman, R. P. (1963). *The Feynman Lectures on Physics III* (Vol. III). New York, New York , USA: Addison-Wesley. Retrieved from http://www.engineersbench.com/phil/docs/books/The%20Feynman%20Lectures%20on%20Physics%20-%20Book%201.pdf

Feynman, R. P. (1982). Simulating Physics with Computers. *International Journal of Theoretical Physics,, 21*, 467-488. Retrieved from https://people.eecs.berkeley.edu/~christos/classics/Feynman.pdf

Griffiths, D. J. (2016). *Introduction to Quantum Mechanics.* Cambridge: Cambridge U. P.

Grover, L. K. (1996). A fast quantum mechanical algorithm for database search. *STOC*, 212-219.

Grover, L. K. (2001). From Schrödinger's Equation to the Quantum Search Algorithm. *arXiv:quant-ph/0109116 22 Sep 2001*, 5-6. Retrieved from https://arxiv.org/pdf/quant-ph/0109116.pdf

Guilherme Tosi, F. A. (2017). Silicon quantum processor with robust long distance coupling. *Nature Communications 8, 8*, 450 . Retrieved from https://arxiv.org/abs/1509.08538

IBM Quantum Experience. (2017). *IBM Quantum Experience*, WEB. Retrieved from https://www.research.ibm.com; https://github.com/ Qiskit/qiskit-api-py/ibm.qx-user-guides

Knight, C. G. (2005). *Introductory Quantum Optics.* Cambridge, United Kingdom: Cambridge U. P. Retrieved from http://assets.cambridge.org/97805218/20356/frontmatter/9780521820356_frontmatter.pdf#page=1&zoom=auto,-135,969

Ladera, M. M. (2017, Dec 4). Andreev levels as a first approach to quantum computing with high-Tc superconductors. *arXiv.org > quant-ph > arXiv:1712.01233*, 1-3. Retrieved from https://arxiv.org/abs/1712.01233 Laloë, C. C.-T. (1977). *Quantum Mechanics 1 Chs. 2, 3.* New York: Wiley. Retrieved from http://fulviofrisone.com/attachments/article/480/Cohen-Tannoudji%20C.,%20Diu%20B.,%20Laloe%20F.%20Quantum%20mechanics,%20vol.%201(T)(887s).pdf

P. J. Coles, S. E. (2018, April). Quantum Algorithm Implementations for Beginners. *arXiv 1804.03719v1 [cs.ET] 10 April 2018*. Retrieved from https://docopdf.com/queue/arxiv180403719v1-cset-10-apr-2018.html

Shor, P. W. (1994). Algorithms for Quantum Computation: Discrete Logarithms and Factoring. *IEEE Computer Soc*, 124-134.

Styles, I. B. (2015). Shor's Factorisation Algorithm. *www.cs.bam.ac.uk*, 1-3. Retrieved from http://www.cs.bham.ac.uk/internal/courses/intro-mqc/current/lecture07_handout.pdf

Svozil, V. P. (2015). Quantum music. *arXiv:1503.09045v2 [quant-ph] 17 Aug 2015*, 2-3. Retrieved from https://arxiv.org/pdf/1503.09045.pdf