An introduction to quantum computing; Advantages and challenges

(6th National Conference on Distributed Computing and Big Data Processing)

Majid Abdolrazzagh-Nezhad[1], Mohammad Amir Jamali[2]

[1] Assistant Professor, Department of Computer Engineering, Technical and Engineering Faculty, Bozormehr Qaenat University, Qaen, Birjand,

abdolrazzagh@buqaen.ac.ir

[2] Master's degree, Department of Computer Engineering, Faculty of Technology and Engineering, Islamic Azad University, Birjand Branch, Birjand

mohammadamirjamali2020@gmail.com

Abstract

The power of quantum computing technologies is based on the fundamentals of quantum mechanics, including quantum superposition, quantum entanglement, or the no-simulation theorem. Since these phenomena do not have a classical analogue, it is not possible to achieve similar results in the framework of traditional calculations. Experimental insights into quantum computing technologies have already been demonstrated and many studies are underway. In this review, we first take a look at the history of quantum computing and then we provide a definition of a qubit or quantum information bits, and then we discuss the differences between quantum computing and classical computing and explain the basic differences, including logic gates that are used to change and manipulate Information or quantum bits in these calculations.

1. Introduction

The field of quantum computing began in the early 1980s. The need for quantum computers (QC) to efficiently simulate quantum physics was first predicted by Richard Feynman [1, 2]. Quantum computing technology offers fundamentally different solutions to computational problems and enables more efficient problem solving than classical computing. Experimental results are promising and quantum computers may be commercially available in a few years [3-7]. One of the most famous algorithms that shows the power of quantum computers is Shor's prime factorization algorithm [8].

Classic computers use conventional and classic bits to perform calculations and processes, which can be 0 or 1. In contrast, quantum computers use quantum bits or qubits, which can be both 0 and 1 at the moment, in fact, this feature makes quantum computers superior. The world's first quantum computer using superconductors was built by a Canadian company called D-Wave Systems. A 28-qubit quantum computer was demonstrated in 2007, followed by 128-qubit in 2010, 512-qubit in 2013, and 2,000-qubit in 2018. However, there is still heavy debate about whether D-Wave computers are truly quantum exists [2].

Recently, IBM announced its company's goal to build universal quantum computing systems available on the commercial market. Now their first IBM-Q system and services are offered through the Cloud. In addition to IBM, other companies such as Google, Intel and Microsoft have also joined the competition for a global quantum computer. All of them have recently made great research efforts in the field of quantum computing, and have spent large amounts of money on their QC labs around the world [2]. Smaller European projects are also underway. For example, MOS-Quito the Quantum Information Technology (MOS)-based project [9], which started after the first qubit-compatible CMOS MOSFETs were built in France.

Many practical researches on other types of quantum bits (qubits) have been proposed and investigated at the laboratory level. Solid-state implementations have gained attention in recent years due to their potential to scale to larger numbers of qubits. It seems that solid state qubits are the main option for industrial and commercial research and development [2]. There are different objects that can be used as qubits to send information: a photon, a nucleus or an electron.

Global and efficient simulation of physical systems is one of the most attractive applications of quantum computing [10]. In particular, quantum simulation of molecular energies [11], which enables accurate numerical prediction of chemical reaction rates, promises significant advances in our understanding of chemistry and can be used in the design of silicon catalysts, pharmaceuticals, and new materials [12].

2- When to use quantum computing

Quantum computers are only faster than conventional computers for certain problems, usually "very hard" problems. A quantum computer works multiple times by executing the same quantum algorithm. The most probable result after these executions is the solution. The time it takes to execute algorithms for hard problems multiple times in a quantum computer is still exponentially faster than a single execution of conventional computers on the same problem [2].

Many problems have high complexity and high degree of difficulty. These problems cannot be solved in a reasonable amount of time in today's supercomputer clusters. Therefore, we can benefit from quantum computers in problem optimization, machine learning, sampling of large data sets, forecasting, etc. Another example is quantum chemistry (e.g. cloning proteins for a new drug), which is currently running at the limit of classical computers. To simulate reality in a simple molecule, every electron-electron interaction must be considered [2].

Next, Shor's algorithm is the most famous example of a computing problem that requires a quantum computer to solve. To factor an N-digit number into its primes, a classical computer might need more than the age of the world to get the result. Therefore, it is used as a fundamental tool for encryption in information security worldwide. A quantum computer can solve this problem very quickly. Quantum chip manufacturers are making great efforts to avoid quantum effects in order to use them in computing [2].

2-1_ Applications of quantum computing:

Classical computers are limited in the size and complexity of the molecules they can simulate and compare (a fundamental process in early drug development). If we have an input of size N, where N is the number of

atoms of the molecules under investigation, the number of possible interactions between these atoms is exponential (each atom can interact with all atoms). For reading about the recent results of quantum chemistry in quantum computer, we suggest [13]. Also, to learn more about how to implement and estimate the gate count to perform quantum chemistry in small quantum computers, it can be found in [14].

Since quantum artificial intelligence and quantum machine learning are emerging fields, the study of these protocols is of fundamental importance for the experimental processing of quantum information. In [6] the authors studied the implementation of some basic quantum reinforcement learning protocols using superconducting quantum circuits. Superconducting quantum circuits are a viable technique for the practical realization of quantum computing and quantum information processing. In [15], the authors studied a classical quantum deep learning framework for industrial datasets on short-lived devices. Other applications include cyber security, financial modeling, traffic optimization, weather forecasting and climate change, etc.

3-Qubit and quantum logic gates

3-1_ Qubit

A qubit or quantum bit is the main container of information in quantum computing, which replaces the bit in a conventional computer. A qubit can be in two states, neutral and excited (Figure 1). The two logical states of each qubit must be suitable for the systems. The simplest example is spin. The spin of a qubit depends on its electronic or nuclear spin degree of freedom [16], which can hold a bit of quantum information for a long time. Note that there are many examples of qubits: two different polarizations of a photon, two energy states of an electron orbiting an atom, etc. Quantum computers are fundamentally different from classical computers due to the nature and characteristics of qubits. The first feature that makes qubits different from classical bits is "quantum superposition" or the linear combination of possible states. The second case is "quantum entanglement" [2, 16].

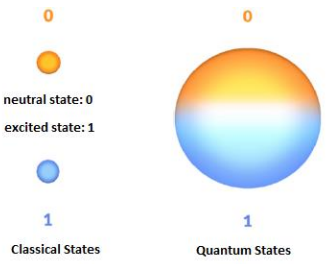

Figure (1): The main unit of information in QC is the quantum bit or qubit, which can be in any linear combination of neutral and excited states [2]

There is a special quantum state that corresponds to the 0 and 1 states of a classical bit. The quantum state corresponding to 0 is usually denoted by |0⟩, the symbol |⟩ is used to denote a qubit and it is called a Ket. This is a fancy symbol representing the following vector [17]:

$$|0\rangle := \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad (1)$$

This special state |0⟩ is called the computational basis state. When we use Ket with something, we mean that it is a vector. So |0⟩ is a computational basis state for a qubit and plays almost the same role as 0 plays for a classical bit. In this way, the other computational mode |1⟩plays the same role as 1. Like |0⟩, |1⟩is just a notation for a two-dimensional vector [17], in this case:

$$|1\rangle := \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (2)$$

3-1-1_ General states of a qubit

Computational states |0⟩ and |1⟩ are only two possible states for a qubit. More modes are possible and these modes are extra features that normal classic bits don't have. In general, a quantum state is a two-dimensional vector. Here's an example, with a graphic that emphasizes the vector nature of state:

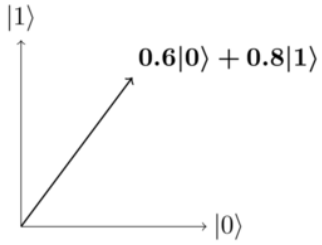

Figure (2): An example of the vector nature of quantum states [17]

A quantum state as a mathematical subject is a two-dimensional vector in a complex vector space. In fact, the idea of describing a simple quantum system using a complex vector in two dimensions summarizes much of what has been learned over the past 25 years.

3-2_ Quantum superposition:

Generally speaking, superposition is the linear combination [17]. Consider a system with two ground states, call them |0⟩ and |1⟩. A classical bit of data can be represented by an atom that is in one of two states denoted by |0⟩ and |1⟩ (left side of Figure 1). As mentioned earlier, in contrast, a quantum state of a qubit is continuously between "0" and "1" until the qubit is measured, but the result can only be "0" or "1" [2]. Therefore, the qubit is a continuous object and its quantum state is given by:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \quad (3)$$

where α and β are complex domains. If we measure this based on calculations, the state |0⟩ with probability |α2| or state |1⟩ with probability |β2| We get as follows [2]:

$$|\alpha^2| + |\beta^2| = 1 \quad (4)$$

The restriction is that the sum of the squares of the domains is 1, this action is called the Normalization Restriction. If one qubit can be in a superposition of two classical states, two qubits can be in a superposition of four states, and n qubit can be in a superposition of 2^n states $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle + \ldots + \mu\,|2^n\text{-}1\rangle$ therefore:

$$\sum_{j=0}^{2^n-1} |a_j|^2 = 1 \quad (5)$$

Figure 3 shows the Bloch sphere. The Bloch sphere provides a useful tool for visualizing the state of a single qubit.

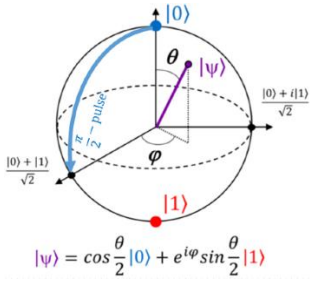

Figure 3: The Bloch sphere provides a useful tool for visualizing the state of a single qubit and operations on it, each point of this sphere represents a linear combination of 0 and 1 states with complex coefficients, a ∏ / 2 pulse of a qubit It rotates from state 0 to super state [2].

3-3 _ Quantum entanglement

The second feature that distinguishes quantum computers from conventional computers is entanglement. If two qubits are "entangled", there is correlation between the two qubits. If one qubit is in a certain state, the other must be in another state. If two electrons are entwined, their spin state is correlated so that if one of the electrons has spin-up (can be considered state 0) or high spin, the other after measurement will be spin-down (can be considered state 1) [2]. Albert Einstein pointed out this feature in 1935[19]. Creating and manipulating entangled states plays a fundamental role in quantum information processing.

There is no clear interpretation of this state as a classical state, perhaps a computational ground state such as $|00\rangle$. In fact, entangled states can be used to perform a variety of interesting information processing tasks, including quantum transport and fast quantum algorithms [17].

3-4_ How the qubit works

For example, an electron in an atom can be in a neutral state or an excited state. By shining light on the atom, with the right energy and for the right time, the electron can be transferred from the neutral to the excited state or vice versa. More interestingly, by reducing the time of light irradiation, an electron can be moved from the $|0\rangle$ state to the halfway between $|0\rangle$ and $|1\rangle$.

Similar to the previous example, first the qubits of a quantum computer, using a large static magnetic field, about one tesla, are first placed in high spin (zero state on the Bloch sphere in Figure 3). Then an electromagnetic pulse is applied to each qubit separately to bring the qubit to any possible state in the Bloch sphere. The actual qubit state depends on the pulse amplitude and time.

3-5_ Quantum logic gates

A quantum logic gate is a method for manipulating and changing quantum information, that is, changing the quantum state of a qubit or a set of qubits. They are similar to the classic logic gates used in everyday computers such as AND, OR and NOT gates. Just like the role of classical gates in conventional computers, quantum gates are the main building blocks of quantum computing. They are also a suitable way to describe many other tasks in quantum information processing, such as quantum transmission [17]. To build a universal quantum computer, a set of quantum logic gates, similar to those found in classical computers, is needed [20]. A single operator that operates on a small number of qubits is often called a "gate".

3-5-1_ NOT quantum gate: X

Let's take a look at the first quantum logic gate, the quantum NOT gate. As you can no doubt guess, the quantum NOT gate is an extension of the classical NOT gate, which is also called the X gate. On a computational basis, the quantum NOT gate does what you would expect, mimicking the classical NOT gate. That is, it brings the state $|0\rangle$ to $|1\rangle$ and vice versa [2, 17].

$$\begin{aligned} \mathrm{NOT}|0\rangle &= |1\rangle \\ \mathrm{NOT}|1\rangle &= |0\rangle \end{aligned} \tag{5}$$

But computational ground states are not the only possible states for a qubit. What happens when we apply the NOT gate to the general superposition state α |0〉 + β |1〉 ? In fact, it does almost the simplest thing possible: it acts linearly on the quantum state, swapping the roles of |0〉 and |1〉 :

$$\text{NOT}(\alpha|0\rangle+\beta|1\rangle) = \alpha|1\rangle+\beta|0\rangle \quad (7)$$

We used the NOT symbol for the quantum NOT gate. But for historical reasons people working on quantum computing usually use a different notation, the symbol X, and so the above formula is rewritten as:

$$X(\alpha|0\rangle+\beta|1\rangle) = \alpha|1\rangle+\beta|0\rangle \quad (8)$$

In a quantum circuit, we represent the gate X as follows:

Figure (4): NOT gate in quantum circuit

3-5-2_ Hadamard quantum gate: H

Another important quantum gate is the Hadamard gate, which is used to create quantum superposition (π/2 pulse). As with the X gate, we will begin by explaining how the Hadamard gate works in computational modes. This gate is denoted by H and its function is as follows:

$$\text{H} | 0\rangle = \frac{| 0\rangle+| 1\rangle}{\sqrt{2}} \quad (9)$$

$$\text{H} | 1\rangle = \frac{| 0\rangle-| 1\rangle}{\sqrt{2}} \quad (10)$$

Specifically, the Hadamard gate transforms a superposition α |0〉 + β |1〉 into the following output:

$$\text{H}(\alpha|0\beta|1\rangle) = \alpha(\frac{|0\rangle+|1\rangle}{\rangle+\sqrt{2}}) + \beta(\frac{|0\rangle-|1\rangle}{\sqrt{2}}) \quad (11)$$

The Hadamard gate is shown in the circuit as shown below. Just like the X gate, the H gate has a matrix representation as shown in equation (12).

Figure (5): H gate in quantum circuit

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (12)$$

3-5-3_CNOT quantum gate:

The gates that we introduced so far were applicable for one qubit alone. To compute, we need a way for the qubits to interact with each other. That is, we need quantum gates that contain two (or more) qubits. An example of such gates is the CNOT gate. In the quantum circuit language, we have two wires that represent two qubits, and the following symbol is used to represent the CNOT gate:

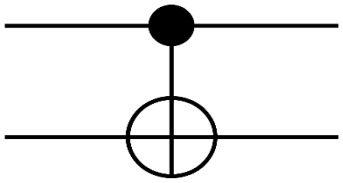

Figure (6): CNOT gate in quantum circuit

This gate actually negates the second bit of its input if the first bit is $|1\rangle$ and does nothing if the first bit is $|0\rangle$. Therefore, CNOT is a quantum gate which is different from NOT and the output depends on the first input. The first qubit is called the control qubit, and the second qubit is called the target qubit.

$$\begin{aligned} |00\rangle &\text{ --CNOT}\rightarrow |00\rangle \\ |01\rangle &\text{ --CNOT}\rightarrow |01\rangle \\ |10\rangle &\text{ --CNOT}\rightarrow |11\rangle \\ |11\rangle &\text{ --CNOT}\rightarrow |10\rangle \end{aligned} \quad (13)$$

And its matrix is as follows:

$$\text{CNOT} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (14)$$

4 - Conclusion:

In this review, we described quantum computing with a focus on logic gates and its differences with the classical system. This is more of a brief overview with short descriptions of various aspects of quantum computing. Although the field of quantum computing and its infrastructure is still in its beginning steps, today quantum computing, quantum optical transmission, and quantum cryptography are commercially available. As we said, the experimental realization of Shor's quantum factorization algorithm is possible through quantum computing. Considering that we are still in the early stages of this technology, various fields can be pursued. Among them: pattern recognition problem in quantum computer, quantum machine learning, quantum deep learning, study of laboratory implementation of quantum support

vector machine, quantum artificial intelligence, quantum point neural networks, quantum inference in Bayesian networks, etc.

References

.1 Feynman, R.P., *Simulating physics with computers.* Int. J. Theor. Phys, 1982. **21**(6/7).

.2 Jazaeri, F., et al. *A review on quantum computing: From qubits to front-end electronics and cryogenic MOSFET physics*. in *2019 MIXDES-26th International Conference" Mixed Design of Integrated Circuits and Systems"*. 2019. IEEE.

.3 Barends, R., et al., *Superconducting quantum circuits at the surface code threshold for fault tolerance.* Nature, 2014. **508**(7497): p. 500-503.

.4 Biamonte, J., et al., *Quantum machine learning.* Nature, 2017. **549**(7671): p. 195-202.

.5 Debnath, S., et al., *Demonstration of a small programmable quantum computer with atomic qubits.* Nature, 2016. **536**(7614): p. 63-66.

.6 DiCarlo, L., et al., *Demonstration of two-qubit algorithms with a superconducting quantum processor.* Nature, 2009. **460**(7252): p. 240-244.

.7 Gyongyosi, L. and S. Imre, *A survey on quantum computing technology.* Computer Science Review, 2019. **31**: p. 51-71.

.8 Shor, P.W., *Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer.* SIAM review, 1999. **41**(2): p. 303-332.

.9 *https://www.mos-quito.eu/*.

.10 Lloyd, S., *Universal quantum simulators.* Science, 1996: p. 1073-1078.

.11 Aspuru-Guzik, A., et al., *Simulated quantum computation of molecular energies.* Science, 2005. **309**(5741): p. 1704-1707.

.12 O'Malley, P.J., et al., *Scalable quantum simulation of molecular energies.* Physical Review X, 2016. **6**(3): p. 031007.

.13 Lanyon, B.P., et al., *Towards quantum chemistry on a quantum computer.* Nature chemistry, 2010. **2**(2): p. 106-111.

.14 Wecker, D., et al., *Gate-count estimates for performing quantum chemistry on small quantum computers.* Physical Review A, 2014. **90**(2): p. 022305.

.15 Benedetti, M., J. Realpe-Gómez, and A. Perdomo-Ortiz, *Quantum-assisted Helmholtz machines: A quantum–classical deep learning framework for industrial datasets in near-term devices.* Quantum Science and Technology, 2018. **3**(3): p. 034007.

.16 *https://quantumai.google/education*

.17 *https://quantum.country/*.

.18 Nielsen, M.A. and I. Chuang, *Quantum computation and quantum information*. 2002, American Association of Physics Teachers.

.19 Einstein, A., B. Podolsky, and N. Rosen, *Can quantum-mechanical description of physical reality be considered complete?* Physical review, 1935. **47**(1 :(0p. 777.

.20 Deutsch, D., *Quantum theory, the Church–Turing principle and the universal quantum computer.* Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences, 1985. **400**(1818): p. 97-117.