Quantum computing : a short introduction.

Thomas Cluzel, Claude Mazel, David R.C. Hill

Quantum computing – a short introduction

LIMOS Research Report 3012–2019

Thomas CLUZEL – Claude MAZEL – David R.C. HILL
Université Clermont-Auvergne, CNRS, Mines de Saint-Étienne, Clermont-Auvergne-INP
LIMOS UMR 6158 - 63000 – ISIMA Clermont-Ferrand, France.

ABSTRACT: In this short introduction paper, we first describe the principles of quantum physics required to understand quantum computing, namely, quantum superposition, quantum entanglement and wave-function collapse. These principles define the behavior of a qubit which represents quantum information. Qubits are limited by decoherence, which prevents them to stay in a superposed state for a long period of time, and by the no-cloning theorem, which prevents the copy of the superposed state of a qubit. We finally mention that to process quantum data represented with qubits with quantum algorithms, we need quantum circuits composed of quantum gates.

1. Introduction

In 1843, Ada Lovelace published the first computer program in history. This program was intended to be executed on Charles Babbage's analytical machine. However, this machine was not built yet at that time. Ada Lovelace died before being able to test her program. Quantum computing is now slightly more advanced than analytical computing when Ada Lovelace and Charles Babbage designed such computers. The concepts of quantum computing were devised several decades ago, and quantum algorithms based on these principles were also developed during this period. However, we do not have generic quantum computers to use these algorithms yet. Today's quantum computers are devices limited in terms of computation time and memory but they are growing from year to year. Following Google's announcement of quantum supremacy*[1] [Arute *et al.* 2019], this report aims at providing an overview of the advancement in quantum computing with a short introduction to the concepts needed to understand how we can compute with such machines. This report also presents a state of the art to help understanding what quantum computing is. First, we present the history of Quantum Computing, then we will introduce the core concepts of quantum physics and the basic concepts of Quantum Computing. Finally, we will see an overview of Quantum Algorithms and Technologies.

2. History of Quantum Computing

The idea of using the principles of quantum physics in computer science was first expressed in 1980, when Paul Benioff described the concept of a quantum Turing machine [Benioff 1980]. Five years later, David Deutsch shows that any function computable by a conventional computer can be implemented by a quantum circuit* [Deutsch Penrose 1985].

It was then necessary to wait until 1992 to see the birth of the first quantum algorithm: the Deutsch-Jozsa algorithm [Deutsch Jozsa 1992]. The purpose of this algorithm is to determine whether a Boolean function $f : \{0,1\}^n \rightarrow \{0,1\}$ is constant or balanced. A constant function always returns 0 or always returns 1. A balanced function returns 0 for half of the inputs and 1

[1] Words annotated with an asterisk are defined in the glossary given in Appendix

for the other half. This algorithm is not very useful in practice but has the advantage of being a pioneer in its field. Moreover, it is deterministic, which is rare for a quantum algorithm.

Other algorithms followed the one of Deutsch and Jozsa. In particular Shor's algorithm in 1994 [Shor 1994], which aims at factoring out a number in a product of two prime factors in polynomial time. We can also mention Grover's algorithm, devised in 1996 [Grover 1996], that searches an element satisfying a property in any set, in sublinear time.

The theory of quantum computing was growing but no concrete tests had been held yet and devised algorithms were still theoretical. In 1998, the first quantum computer in history executed the Deutsch-Jozsa algorithm [Jones Mosca 1998]. It was a 2 qubits* quantum computer based on nuclear magnetic resonance. From that moment on, quantum computing was no longer a theory but becomes a reality.

Following this success, more studies were led on quantum computers and quantum processors. Thus, in 2001, IBM announced that they managed to factor the number 15 into 3x5 with a quantum computer processing 7 qubits [Vandersypen *et al.* 2001]. The usefulness of quantum algorithms was now highlighted. The only hurdle that stood in front of them was vertical scaling. The number of available qubits and the computation time were now the two main parameters to improve on quantum computers.

Thus, since the 1980s, new quantum algorithms have been discovered for many applications. To execute these algorithms, quantum computers are becoming more powerful and affordable. In January 2019, at CES in Las Vegas, IBM presented the "IBM Q System One": the first commercial quantum computer, with 20 qubits[2].

In October 2019, Google claimed they have achieved quantum supremacy [Arute *et al.* 2019]. This means that they built a quantum computer that outperforms any conventional computer [Preskill 2012]. This quantum processor, named Sycamore, has 53 qubits. It solved the quantum circuit sampling problem in 200 seconds. According to Google, it would have taken 10,000 years to Summit, the world's most powerful supercomputer, to solve the same problem. According to IBM, Summit's designer, the same result could be achieved in two and a half days using the entire supercomputer memory[3]. However, it is worth mentioning that IBM also have a 53 qubits quantum computer[4].

Apart from research on quantum algorithms and the race to quantum supremacy, development tools for quantum computing have emerged. The first quantum scripting language, QCL (Quantum Computation Language), appeared in 1998 [Ömer 1998]. It was purely theoretical as it could not be interpreted on any device. Its only goal was to propose a formalism for the design of quantum algorithms. Other types of tools were later created such as GUIs (Graphical User Interface) to drag-and-drop* quantum gates* on partitions* (Quirk for example) or libraries for classic programming languages such as Qiskit or Cirq. There are also quantum simulators in C (e.g. QuEST), in C++ (e.g. Quantum++), in Lisp (e.g. QVM), in Haskell (e.g. Quipper), in Java (e.g. jQuil), in Javascript (e.g. Quirk), in OCaml (e.g. QOCS), in Python (e.g. QuPy). There are also abstractions between high-level languages and quantum computers such as OpenQASM

[2] J. Russell, "IBM Quantum Update: Q System One Launch, New Collaborators, and QC Center Plans" January 10th, 2019. [Online]. Available: https://www.hpcwire.com/2019/01/10/ibm-quantum-update-q-system-one-launch-new-collaborators-and-qc-center-plans/. [Accessed on October 30th, 2019].

[3] E. Pednault, J. Gunnels, D. Maslov and J. Gambetta, "On "Quantum Supremacy" | IBM Research Blog" October 21st, 2019. [Online]. Available: https://www.ibm.com/blogs/research/2019/10/on-quantum-supremacy/. [Accessed on October 30th, 2019].

[4] S. Shankland, "IBM's new 53-qubit quantum computer is its biggest yet - CNET" September 18th, 2019. [Online]. Available: https://www.cnet.com/news/ibm-new-53-qubit-quantum-computer-is-its-biggest-yet/. [Accessed on October 30th, 2019].

[Cross *et al.* 2017] or OpenFermion which are assembly language for quantum processors. Finally, there are even compilers for quantum programming languages like Q# [Prieur 2019].

Today's major players of quantum computing are large IT companies such as IBM, Microsoft, Google, Atos or Intel, but also some start-ups like Rigetti Computing and research centres. They all work on different parts of quantum computing. For example, Microsoft edits the Q# language. Google designed the Sycamore processor. D-Wave is specialized in quantum annealing, a specific application of quantum computing. Researchers like Peter Shor discover quantum algorithms and so on.

3. Core concepts and notations

The classical model of physics enables the explanations of phenomena on a macroscopic scale. However, classical physics cannot be applied at a microscopic level. To explain what happens at a microscopic scale, it is necessary to use quantum physics. Many laws exist in quantum physics such as quantification (an atom can only be excited at specific levels of energy), wave-particle duality (the behaviour of a particle can be described as a wave or as a body) or the Heisenberg uncertainty principle (there is a fundamental limit to the precision with which several properties of a particle can be measured simultaneously). However, not all these laws are necessary to understand quantum computing.

Three principles of quantum physics are fundamental in quantum computing:

- Quantum superposition.
- Quantum entanglement.
- Wave function collapse.

In classical physics, a particle is in a given state at a given time. On the contrary, in quantum physics, a particle can be found in a superposition of states thanks to the principle of quantum superposition*.

Let $n \in \mathbb{N}$, $\{|e_1\rangle, \dots, |e_n\rangle\}$ the set of pure states* in which a particle can be and $\{a_1, \dots, a_n\} \in \mathbb{C}^n$. Then the superposition of these pure states is:

$$|\psi\rangle = \sum_{i=0}^{n} a_i |e_i\rangle \,, \forall i \in \{1, \dots, n\}, |a_i|^2 \in [0,1] \,; \sum_{i=0}^{n} |a_i|^2 = 1$$

A quantum state is written $|e\rangle$ and pronounced "ket e". It is a simplified notation of a column matrix. Hence, a superposition is a linear combination of vectors.

$$|e\rangle = \begin{bmatrix} e_{11} \\ e_{21} \\ \vdots \\ e_{m1} \end{bmatrix}$$

The notation $\langle l|$ also exists and is pronounced "bra l". It is the simplified notation of a line matrix.

$$\langle l| = [l_{11}\ l_{12} \dots\ l_{1m}]$$

The names “bra” and “ket” come from a pun with the English term bracket. Furthermore, the notation $\langle l|e\rangle$ refers to the dot product of vectors l and e and is a scalar. These notations were introduced by Paul Dirac in 1939 to simplify the writing of quantum mechanics equations [Dirac 1939] and to emphasize that representing a quantum object requires vectors.

The second important principle in quantum physics to understand quantum computing is quantum entanglement* [Fickler *et al.* 2013]. Saying that two particles are entangled means that two particles are bound. Therefore, if we know the state of the first particle, we can deduce the state of the second particle.

The last but not least principle of quantum physics required to understand quantum computing is wave function collapse* [Heisenberg 1927]. This principle says that when measuring the state of a quantum system, the system is transformed to its measured state. The state of the system “collapses”. The measured state can only be a pure state of the system. It is worth noting the following consequences:

- Measuring the state of a quantum system destroys its superposed state.
- It is impossible to restore the superposed state that a system had before a measurement. As a result, it is not possible to perform several consecutive measurements of the state of a superposed system.
- It is impossible to know the values of the coefficients of the pure states of the superposed state of a system.

However, there is good news. The coefficients of the pure states in the superposed states determine the probability to measure one pure state or another. Hence, the measure can be seen as a stochastic experiment whose possible outcomes are the pure states with, as probabilities, the squares of modules of their coefficients. Let $|\psi\rangle$ be the superposed state of a quantum system.

$$|\psi\rangle = \sum_{i=0}^{n} a_i |e_i\rangle$$

$$\forall i \in \{1, \dots, n\}, P(measure(|\psi\rangle) = |e_i\rangle) = |a_i|^2$$

Now that we have detailed the principles of quantum physics used in quantum computing, we can dive into the core concepts of quantum computing. The first one is the concept of quantum bit.

In classical computing, information is represented with bits* (“Binary digits”). One bit can take its values in $\{0, 1\}$. It can be seen as a Boolean* variable. Bits can be grouped to store larger information. For instance, a group of 8 bits is called a byte*.

In quantum computing, information is represented with quantum bits, also written qubits or qbits. A qubit is in superposition of the pure states $|0\rangle$ and $|1\rangle$ (pronounced “ket zero” and “ket one”). As described earlier a qubit is written like this:

$$|qubit\rangle = a|0\rangle + b|1\rangle, a \in \mathbb{C}, b \in \mathbb{C}, |a|^2 + |b|^2 = 1$$

After the measure, if the qubit is in the state $|0\rangle$, we consider that the result is a bit of value 0. If the measure is $|1\rangle$, we consider that the result is a bit of value 1.

Measured state of a qubit	Value of the produced bit
$\|\mathbf{0}\rangle$	0
$\|\mathbf{1}\rangle$	1

Table 1: Correspondence between the measured state of a qubit and the value of the bit produced

To craft such qubits, there are some physical phenomenon that can be used like Josephson nanojunctions [Vion 2003] or electron holes [Prechtel *et al.* 2016]. To represent the concept of qubit, the Bloch sphere [Bloch 1946] is often used (see figure 1). It is a geometrical representation of a qubit that translate:

$$|\psi\rangle = a|0\rangle + b|1\rangle, a \in \mathbb{C}, b \in \mathbb{C}, |a|^2 + |b|^2 = 1$$

to:

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\varphi}\sin\left(\frac{\theta}{2}\right)|1\rangle, 0 \leq \theta \leq \pi, 0 \leq \varphi \leq 2\pi$$

With this representation, we assume that a is a real number. We can do so without loss of generality. On this sphere, each point is defined by:

$$\begin{cases} x = \sin\theta \times \cos\varphi \\ y = \sin\theta \times \sin\varphi \\ z = \cos\theta \end{cases}$$

Like bits, qubits can be grouped. One group of n qubits is the superposition of 2^n pure states and has 2^n coefficients. For instance, two qubits are represented as follows:

$$|2\ qubits\rangle = a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle, a \in \mathbb{C}, b \in \mathbb{C}, c \in \mathbb{C}, d \in \mathbb{C}$$

$$|a|^2 + |b|^2 + |c|^2 + |d|^2 = 1$$

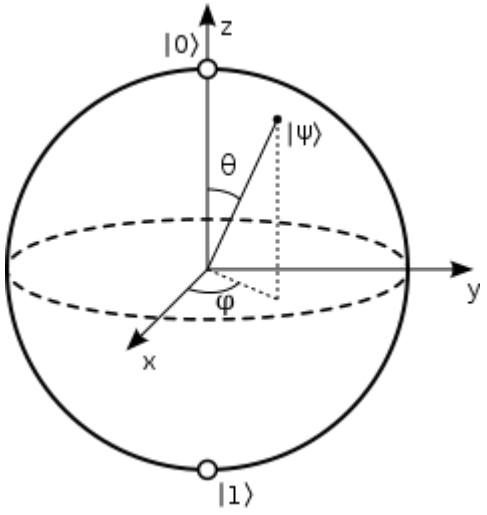

Figure 1 : Bloch sphere[5]

It is possible to entangle two or more qubits. Thus, knowing the state of one of them, we can deduce the state of the other. For instance, with two qubits:

[5] Smite-Meister, "Bloch sphere, a geometrical representation of a two-level quantum system" January 30th 2009. [Online]. Available: https://en.wikipedia.org/wiki/File:Bloch_sphere.svg. [Accessed on November 19th, 2019].

$$|2\ entangled\ qubits\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$$

In this case, if the first qubit we measure is in the state $|0\rangle$, then the other must be in the state $|0\rangle$ because the probability to obtain $|1\rangle$, knowing that the first qubit is $|0\rangle$, is null.

With its ability to be in a superposition of states, the qubit seems to be far more interesting than the bit to process data. However, the qubit has also its own limits. We have already seen one of them: the wave function collapse. We can never have access to the values of the coefficients of the pure states in a superposed state. We can only measure pure states.

A second limit is the quantum decoherence*. This phenomenon occurs when a quantum system loses its quantum properties (superposition) due to interactions with its environment. This means that if we want to use qubits, we need to prevent the external environment to interact with our qubits. The consequence of such a constraint is that the system containing our qubits must be very small to reduce the risk of unwanted interaction. Even if we reduce this risk, decoherence will still happen. In the end, the only thing we can do is to have the longest time possible before decoherence. This time must be long enough to perform operations on qubits and process the data they represent.

We can remark that measuring a quantum system produces decoherence of the system. In this case, it is a desired and controlled decoherence. As mentioned before, decoherence must not occur before the measure.

Another limit induced by the measure is that we only get two different outputs from a qubit. When it is in a superposed state it can be in an infinite number of states, but once measured we can only retrieve 0 or 1 as result.

Finally, one last limit is the no-cloning theorem. This theorem tells us that it is impossible to copy the state of a qubit to another qubit. The direct consequence is that we cannot have affectation operator in quantum algorithms.

We have not seen yet what a quantum algorithm is. We have presented, in this part, the core concepts of quantum physics that are important in quantum computing and the concept of qubit at the roots of quantum computing. Let us see, in the next part, how qubits can be used to process data.

4. Toward quantum data processing

Like classical algorithms, quantum algorithms are devised to process data. However, the data they must process is represented as qubits instead of bits. Qubits have their own limitations and require dedicated algorithms to process them.

To process bits, the most basic tools we can use are logical gates. Like classical ones, quantum algorithms have their own gates to manipulate the superposed states of qubits: quantum gates.

Quantum gates can be represented as square matrices of order 2^n to process registers* of n qubits. Some interesting gates are:

- Hadamard: used to superpose the state $|0\rangle$ and $|1\rangle$ of a qubit with coefficients $\frac{1}{\sqrt{2}}$ and $\frac{1}{\sqrt{2}}$;
- X: used to switch the state of a qubit from $|0\rangle$ to $|1\rangle$ or from $|1\rangle$ to $|0\rangle$;
- C_{not}: used to switch the state of a qubit depending on the state of another qubit;
- Toffoli: used to switch the state of a qubit depending on the state of two other qubits.

To apply quantum gates to more than one qubit, they can be expanded with the Kronecker product. Quantum gates can be applied in sequence to qubits in quantum circuits*. At the end of a quantum circuit, there is a measure to retrieve the results of the quantum algorithm.

Several quantum algorithms have already been devised since the 1990s. Some of them are famous like the Deutsch-Jozsa algorithm which was the first quantum algorithm devised. Then, other researchers came up with other algorithms for different purposes. Two famous algorithms are Shor's algorithm and Grover's algorithm. Their goals are the factorization of an integer into two prime numbers for the first one and the research of an element in an unordered set for the second one.

Apart from these two algorithms, other quantum algorithms exist for other domains. Shor's algorithm is a major advancement in cryptanalysis. Consequently, there are also applications of quantum computing in cryptography and post-quantum cryptography. Quantum algorithms exists in many domains like molecular modelling, simulation, logistics optimization, financial modelling or artificial intelligence. These algorithms are promising because the problems they solve are hard for classical computers. Beyond this, there is a theory of complexity for quantum computers, like P and NP for classical computers. Polynomial time problems for quantum computing are in BQP (Bounded error Quantum Polynomial time) [Nielsen and Chuang 2000] while harder ones are in QMA (Quantum Merlin Arthur).

Even though algorithms are interesting, they are not the only point of interest in quantum computing. As mentioned in the first part, there are already several languages to code quantum programs that implement quantum algorithms. The most straightforward ones are graphical tools. Their interfaces offer to drag-and-drop logical gates on a partition representing qubits over time. IBM has made theirs available online through a web browser. Furthermore, they support the OpenQASM language. Another major company in programming languages is Microsoft. Since 2017, they have open sourced their quantum programming language called Q#. It could be interesting to implement the same algorithm with two different languages and compare how different the structures of the codes are. We have mentioned at the beginning that Google had reached Quantum Supremacy in October 2019. Microsoft devised the Q# language but they also launched Azure Quantum following Google's announcement, which is a cloud offer for quantum computing as a service. Some companies work on very specific domain of quantum computing like D-wave which is specialized in quantum annealing.

5. Conclusion

Thus, throughout this report, we first saw that quantum computing was imagined in the early 1980s and the first quantum algorithm was created one decade later. Thenceforth, other algorithms have followed like Shor's one and Grover's one. Not only has the software part of quantum computing evolved, but also the hardware part with the creation of the first quantum computers at the beginning of the millennium. Then, more and more algorithms were discovered, and more and more powerful quantum computers were built. In October 2019, Google reached Quantum Supremacy, meaning that they built a quantum processor that outperforms any classical computer.

Then, we dived into the core concepts of quantum computing. Quantum computing relies on three important quantum physics concepts: quantum superposition, quantum entanglement and wave function collapse. These three principles are used to describe the concept of qubit. A qubit represents quantum information with two pure states: $|0\rangle$ and $|1\rangle$. These two states can be superposed. It is possible to entangle several qubits. This way, we are able to deduce the state of one qubit if we know the states of the others. Finally, even if a qubit can be in an infinite number of superposed states, we do not have access to its superposed state. We need to measure, but due to wave function collapse, we can only measure a pure state. Moreover, the measure reduces the state of the qubit to its measured state. The measure corresponds to a stochastic experiment that has the pure states has possible outcome and the square of the module of the complex coefficient of each pure state as probabilities. This is how we can get 0 and 1 from $|0\rangle$

and $|1\rangle$. However, quantum superposition has its own limits. The first one is the decoherence. If the interaction with the external environment is too high, a quantum system sees its superposed state collapse. A limit of the qubit is that it cannot be cloned because of the non-cloning theorem.

To perform computations with qubits, one needs quantum gates. Quantum gates are like logical gates for qubits. They modify the state of one or more qubits. By assembling gates, we can create quantum circuits. Quantum algorithms can be implemented with quantum circuits. To devise such circuits, tools like GUIs, programming languages with quantum libraries or quantum programming languages can be used. Companies like Microsoft or IBM are competing to propose the best tools and offers.

Quantum computing is an active field of research. There is a lot more to know about this field, we have only addressed the basics. How many quantum algorithms have already been discovered? What research domain are they used in? Which problems can they solve efficiently? How can we implement these algorithms? Which development tools exist? Which quantum computers are available today on the market? How powerful are they? Is quantum computing reliable and reproducible? Broadly speaking, what can we do today with quantum computing?

References

[Arute *et al.* 2019] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon and John M. Martinis, "Quantum supremacy using a programmable superconducting processor", Nature, Vol. 574, October 24th 2019, pp. 505-510.

[Benioff 1980] Paul Benioff. 1980, "The Computer as a Physical System: A Microscopic Quantum Mechanical Hamiltonian Model of Computers as Represented by Turing Machines", Journal of Statistical Physics, Vol. 22, No. 5.

[Bloch 1946] Félix Bloch, 1946. "Nuclear Induction", Phys. Rev., Vol. 70, Issue 7-8, pp. 460-474.

[Cross *et al.* 2017] Andrew W. Cross, Lev S. Bishop, John A. Smolin, Jay M. Gambetta, 2017. "Open Quantum Assembly Language", 24 pages, [arXiv:1707.03429].

[Deutsch Jozsa 1992] David Deutsch, Richard Jozsa, 1992. "Rapid solutions of problems by quantum computation", Proceedings of the Royal Society of London A, Vol. 439, pp. 553-558.

[Deutsch Penrose 1985] David Deutsch, Roger Penrose, 1985. "Quantum theory, the Church–Turing principle and the universal quantum computer", Proceedings of the Royal Society of London A, Vol. 400, Issue 1818.

[Dirac 1939] Paul A. M. Dirac, 1939. "A new notation for quantum mechanics", Mathematical Proceedings of the Cambridge Philosophical Society, Vol. 35, Issue 3, pp. 416-418.

[Fickler *et al.* 2013] Robert Fickler, Mario Krenn, Radek Lapkiewicz, Sven Ramelow, Anton Zeilinger, 2013. "Real-Time Imaging of Quantum Entanglement", Scientific Reports Vol. 3, Article 1914, 5 pages.

[Grover 1996] Lov K. Grover, 1996. "A fast quantum mechanical algorithm for database search", Proceedings, 28th Annual ACM Symposium on the Theory of Computing, pp. 212-219.

[Heisenberg 1927] Werner K. Heisenberg, 1927. "Über den anschaulichen Inhalt der quantentheoretischen Kinematik und Mechanik", Z. Phys. 43, pp. 172–198.

[Jones Mosca 1998] J. A. Jones, M. Mosca, 1998. “Implementation of a quantum algorithm on a nuclear magnetic resonance quantum computer”, The Journal of Chemical Physics, Vol. 109, Issue 5.

[Nielsen and Chuang 2000] M. Nielsen and I. Chuang, 2000, “Quantum Computation and Quantum Information”, Cambridge University Press, (ISBN 0-521-63503-9).

[Ömer 1998] Bernhard Ömer, 1998. “A procedural formalism for quantum computing”, Department of theoretical physics, technical university of Vienna.

[Prechtel *et al.* 2016] Jonathan H. Prechtel, Andreas V. Kuhlmann, Julien Houel, Arne Ludwig, Sascha R. Valentin, Andreas D. Wieck, Richard J. Warburton, 2016. “Decoupling a hole spin qubit from the nuclear spins”, Nature Materials Vol. 15, pp. 981–986.

[Preskill 2012] John Preskill, 2012. “Quantum computing and the entanglement frontier”, Proceedings of the 25th Solvay Conference on Physics, 18 pages.

[Prieur 2019] Benoît Prieur, 2019. “Informatique quantique : de la physique quantique à la programmation quantique en Q#”, ISBN 978-2-409-01741-4, ENI editions, 244 pages.

[Shor 1994] Peter W. Shor, 1994. “Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer”, Proceedings of the 35th Annual Symposium on Foundations of Computer Science, IEEE Computer Society, pp. 124-134.

[Vandersypen *et al.* 2001] Lieven M.K. Vandersypen, Matthias Steffen, Gregory Breyta, Costantino S. Yannoni, Mark H. Sherwood, Isaac L. Chuang, 2001. “Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance”, Nature, Vol. 414, pp. 883-887.

[Vion 2003] Denis Vion, 2003. “Josephson quantum bits based on a cooper pair box”, Quantum entanglement and information processing, session LXXIX (Proceedings of the Les Houches Summer School), 41 pages.

Credit author statement:

Thomas Cluzel: Writing – Original Draft, Conceptualization, Methodology, Software, Investigation. David Hill: Writing – Review & Editing, Conceptualization, Methodology, Validation, Investigation, Supervision, Project administration. Claude Mazel: Writing – Review & Editing, Conceptualization, Methodology, Validation, Investigation, Supervision, Project administration.

APPENDIX

Glossary

Bit: “Binary digit”, can hold a Boolean value 0 or 1.

Boolean: variable that can take two different values “true” or “false” often represented as 0 and 1.

Byte: group of 8 bits.

Drag-and-drop: press the button of the mouse when hover an element to “catch” it, then move (“drag”) the mouse and finally release the button to “drop” the element where the cursor is.

Partition: graphical temporal representation of qubits and gates of a quantum circuit.

Pure quantum state: a base state of the quantum superposition; also, a possible state for a quantum particle before or after measurement.

Quantum circuit: sequence of quantum gates.

Quantum decoherence: when a quantum system loses its quantum properties (superposition) due to interactions with its environment.

Quantum entanglement: link between two quantum particles; knowing the state of one, we can deduce the state of the other.

Quantum gate: base element of a quantum circuit manipulating qubits.

Quantum register: group of qubits on which quantum operation can be applied.

Quantum superposition: combination of the possible states, associated with a probability, in which a quantum particle may be.

Quantum supremacy: limit from which a quantum computer becomes more powerful than any conventional computer (another definition: threshold from which a conventional computer can no longer simulate a quantum computer).

Qubit or qbit (quantum bit): basic unit of quantum information; can be a superposition of pure quantum states $|0\rangle$ and $|1\rangle$.

Wave function collapse: measure of the state of a quantum system (the measure destroys the superposed quantum state of the system collapsing its state to its measured state).