Quantum Error Correction

Stefano Romboni

December 2024

Contents

1 Ideas on the general structure

In my talk I am going to present two papers about Quantum Error Correction implementation using two different platforms, superconducting quantum processor and neutral atom quantum processor. While the Google paper is more focused on the experimental proof of the treshold theorem, and the logical error rate of their processor, the Harvard paper is more focused on the general working scheme of their neutral atom platform with a focus on gate implementation and working principles [This explanation should be improved...].

The talk will start with a small introduction about Quantum Error Correction and the main motivations that lead people to start researching in this area. The main difference between Classical and Quantum computation will be outlined. Going forward, there will be an introduction about topological quantum error correction and the formalism needed to understand the following papers. In particular, there will be a basic introduction on the Stabilizer formalism and the Toric code. Additionally, some information about the Steane code (a particular type of colour code) implemented in the Harvard paper, and the more complex surface code used by Google.

At the end, there will be an outline of the main differences between these two platforms and an outlook on what could be improved in the future.

SUMMMARY TABLE:

1. Introduction on Quantum Error Correction: Small recap from the previous talk and highlights of the most important characteristics (challenges?) of quantum computation
2. Brief hystorical recap and evolution of Quantum Error Correction: For example, SHor algorithm, Kitaev's code, Gottesman PhD thesis...
3. Stabilizer formalism: Main defitions and working principles
4. Toric code A small introduction on the topological meaning of the Toric code
5. Surface code and Steane code Only what is useful to understand the figures in the two papers
6. Google paper The main goal of the paper (what they wanted to proove) and the general direction
7. Harvard paper Working principle, platform explanation, C-Not gate implementation with neutral atom?
8. Conclusion and outlook Small recap and confrontation of the two different platforms and an outlook of the future if possible

2 Introduction about quantum error correction

Quantum computer in order to return a ”coherent” result after performing an algorithm require to be protected by errors. It is not the first time that a computer needs some ”protection” from error. Historically, at the beginning of their development, classical computer were exposed a lot of noise, and people, in order to overcome this obstacle, started developing some algorithm to correct those errors. Unfortunately, while for classical computer there exists only one a single type of error, the bit-flip, this is not the case for quantum computers. In the contest of quantum information, the qubit state can assume a continuum of values, and can be formally visualize in the Bloch sphere representation:

$$|\psi\rangle = \cos(\theta/2)\,|0\rangle + e^{i\phi}\sin(\theta/2)\,|1\rangle\,. \tag{1}$$

An error occurring on this state will lead to a rotated state:

$$|\psi'\rangle = \cos((\theta+\delta\theta)/2)\,|0\rangle + e^{i(\phi+\delta\phi)}\sin((\theta+\delta\theta)/2)\,|1\rangle\,. \tag{2}$$

Qubits are therefore sensitive to a continuum of coherent errors. Another big difference with respect to classical error correction, is given by two major limitations, that initially discouraged people from believing in quantum computers:

1. the no-cloning theorem
2. the wave function collapse

Lucky for us, it turns out that quantum error can be digitised so that the ability to correct for a finite set of errors is sufficient to correct for any error. It is possible to describe coherent noise processes in terms of linear combination of Pauli matrices. [To say this better, all the unitary operation that you can apply to a qubit are unitary operations, and the Pauli group is a generator of all 2×2 unitary matrices. Recall also the algebra of the SU(2) group. This part could be expanded].

$$U(\theta,\phi) = e^{i\theta\hat{n}\cdot\vec{\sigma}} = ... \tag{3}$$

Anyway, protecting a quantum computer from errors turned out to be one of the major challenge. A quantum computer will inevitably interact with the environment, resulting in decoherence and hence the decay o quantum information stored in the ”device”. A possible solution to this problem is to ”perfectly” isolate our our system from any unwanted interaction with the environment, but unfortunately, in order to work, it also needs to perform quantum gates, i.e. unitary operation, chosen from a continuum of possible values. Obviously, quantum gates cannot be implemented with perfect accuracy. Their effects is to accumulate small errors eventually leading to a failure in the output of the computation.

Any effective stratagem to prevent errors in a quantum computer must protect against small unitary (unwanted) evolution in a quantum circuit, as well as against decoherence.

2.1 Digitisation of quantum error

Lucky for us, quantum error can be digitize. In simple words, it is possible to decompose any continuous error into X and Z errors. This is very important because analog errors, even on a classical computer, cannot easily be error-corrected. We have reduced a continuous (infinite) set of errors into only two types of errors: X and Z. Given a general state $|\psi\rangle = a|0\rangle + b|1\rangle$, we have:

$$\text{X-error (or bit-flip):} \quad |\psi\rangle = a|0\rangle + |1\rangle \xrightarrow{X} a|1\rangle + b|0\rangle \tag{4}$$

$$\text{Z-error (or phase-flip):} \quad |\psi\rangle = a|0\rangle + |1\rangle \xrightarrow{Z} a|0\rangle - b|1\rangle \tag{5}$$

2.2 Encoding and decoding

The overall idea of quantum error correction is to encode your quantum information inside a large system (that also means a large Hilbert space), using more qubits than you theoretically need. Then, after the encoding, the state will evolve through a noisy channel, where errors can occur. The larger Hilbert space give us a more freedom (more degree of protection to better say), so that it is possible to perform non-destructive measurements to detect and correct the error, in the decoding process. Depending on the measurement outcome, we perform some operation to restore the information to its original value. The only type of measurement we can perform on the system are non-destructive measurement, also called **stabilizer measurements**, so that the *logical information* is not modified by our actions.

Summary scheme:

$$|\psi\rangle \xrightarrow{\text{encoding}} |\psi\rangle_E \xrightarrow{\text{noise}} |\tilde{\psi}\rangle \xrightarrow{\text{decoding}} |\psi\rangle_E \tag{6}$$

One of the most popular code which can detect both bit-flip and phase-flip is the surface code (also called toric code), which belongs to the more general class of topological quantum codes.

2.3 Hystorically evolution of Quantum Error Correction

One of the main steps of quantum corrections:

1. Shor algorithm (1995) 2. Kitaev's code (1997) 3. Daniel Gottesman PhD thesis (1997)

2.4 OLD: Introduction on Quantum Error Correction

Quantum Computers work with quantum bits, i.e. qubits, while classical computers work with bits. Efficient quantum computing algorithms have been developed with applications such as integer factorisation, search, optimisation and quantum chemistry.

Nowadays there is no preferred platform to run this algorithms; a great variatey of different hardware is been used to create qubits. The main platforms are superconducting qubits, trapped ions, neutral atoms etc. A deficiency shared between all these platforms is the difficulty to sufficiently isolate the system from the external noise, that leads to errors during the computation and measurement steps.

For this reasons, any quantum computer platform require a sort of error correction scheme to rely on. Due to some limitations, intrinsic of quantum mechanics, is not possible to apply classical error correction scheme on a quantum platform (no cloning theorem, different types of errors, wavefunction collapse).

Initially, people thought these limitations where to strict to allow the possibility of any development of error correction algorithm, so quantum computing was not considered much. This was until 1995 when Peter Shor published a paper proposing the first quantum error correction scheme. Shor's method demonstrated how quantum information can be redundantly encoded by entangling it across an expanded system of qubits.

3 The stabilizer formalism

The stabilizer formalism is a powerful set of techniques to define and study quantum error correcting codes in terms of Pauli operators, rather than directly with the state vector kets of the code-words. It has many advantages, and is the most widely used formalism used to describe topological codes. Its advantages include:

- An efficient description of many-qubit states
- A straightforward analysis of symmetries of codes, error detection measurements and corrections
- Clear rules to derive encoded logical operators

3.1 n-qubit Pauli group

3.2 The stabilizer group

3.3 Error correction with stabilizer

4 Introduction to topological quantum error correction

Toric code

Surface code

Staene code

5 The Toric Code

The Toric Code is the simplest example of a *topological code*, or topological *surface code*, introduced by Alexei Kitaev in 1996.

5.1 Defining the code

The toric code is defined in terms of a square lattice with periodic boundary conditions. The lattice consists of*edges, vertices* and *plaquettes*. We associate every qubit with every edge on the lattice. On a $L \times L$ grid with periodic boundary conditions there are $2L^2$ edges, where the edges on the boundary are not counted twice. A toric code has therefore $n = 2L^2$ physical qubit.

5.2 Stabilizer generators

An n-qubit stabilizer code encoding k qubits is defined by specifing $m = n - k$ independent stabilizer generators, together with encoded logical $\bar{X}$ and $\bar{Z}$ qubits. On this toric code, there are two types of stabilizer generators associated with *vertices* and *plaquettes* of the lattice.

5.3 Multiplication and independence of plaquette operators

Any two plaquettes will have a single common boundary edge or no common boundary edge. Thus multiplying together any set of plaquettes will lead to cancellations of Zs at the shared boundaries. The result will be that the product of the operators will consist of the overall boundary of the plaquettes in the product.

Consider now the product of all plaquettes in the entire lattice. What is its boundary? The plaquettes now cover the entire surface of the torus and have no boundary (see figure 2.5). Thus, the product of all plaquette operators is the identity. This means that the full set of plaquettes is not a set of independent operators. Recall, we would like our generating set to be independent, as defined above in equation (1.20). This means if should be impossible to construct the identity from any product of the generators. In order to create an independent set of plaquettes, the resolution is simple. We remove any single plaquette from the generator set. There is then no longer product of such plaquettes with zero boundary, and the operators are thus independent. We therefore count $L^2 - 1$ independent plaquette operators.

5.4 Dual lattice

[...]

5.5 Encoded qubits

[...]

5.6 Equivalence of logical operators under stabilizer multiplication

[...]