Classical and quantum computing theory: a comparison

Erik Deumens*

March 22, 2023

Abstract

The computer science theory of quantum computing has been developed in close parallel with the classical theory. We take a closer look to preent a framework for discussing how the computer science links to the underlying statistics for management of exprimental data and underlying phsyics theory, which is quantum mechanics in both cases.

This is note # 4 in a series of notes to untangle quantum mechanics for a general audience and experts alike.

URL https://people.clas.ufl.edu/deumens/files/pap-computing.pdf

The efforts to build a functional quantum computer have led to the identification of error correction as essential: Without error correction it will be impossible to scale to the long processing times with the number of qubits that is considered necessary to perform computations that are meaningfully better than can be executed on classical computers. Computer scientists have designed [5] quantum error correction (QEC) codes that rely on Born's rule and need to run at that rate for every logical qubit, which is the composite of the several physical qubits to allow error detection and correction.

Error correction for quantum computers has been developed colsely parallel to error correction for classical computers. We briefly sketch the basics of classical bits and quantum bits (qubits) as part of their treatment in computer science as well as in the relevant theory in physics, namely QM, for their implementation. A thorough discussion can be found in many textbooks, such as Nielsen and Chuang [5].

At a conceptual level, computers have three levels of description. For *classical* computers we have:

Top - classical computer science Computers are structures that have bits of information that take on values of 0 or 1; information is encoded in words of a number of

*Quantum Theory Project, University of Florida, ORCID 0000-0002-7398-3090

bits and are stored in registers in the *central processing unit* (CPU) where they are operated on by *gates*.[1] Words can be transmitted over communication lines to memory banks and to other devices. Algorithms are written in programming languages and compilers transform these into streams of gates that are then executed by the computer to produce the desired output.

Bottom - quantum mechanics The computer science model of the classical computer is implemented in an intricate assembly of devices that are governed by the laws of QM. The relevant capabilities of registers for storing bits and of gates for operating on them are provided by atoms and electrons in materials subject to atomic-level processes. The capabilities are implemented in Silicon-based "chips" with billions of elementary gates. The mathematical description at this level includes Hilbert spaces and wavefunctions governed by the SE.

Middle - statistics There is a layer between the abstract computer science and the implementation in physical devices that can be characterized by the fact that it introduces probability and statistics. The physical implementation of bits and gates involves large numbers of electrons and atoms. It is not necessary, nor practical or even possible, to describe the individual dynamics of the electrons and atoms involved in the phsyics of the materials used. Rather a statistical treatment suffices. The devices used to implement classical computers operate in what is known in physics as the *classical limit* of QM. That means that all the weirdness of QM has been washed out and is no longer visible. In turn, that allows for a clean implementation of the classical computer science model of a computer.

In addition, there are a range of internal and external influences acting on the physical computer devices that cause deviations from the design: these show up as device errors. Built into the devices are capabilities to stabilize the voltages of the devices that hold the "bits," with 0 represented by some positive voltage and 1 by a negative voltage. The devices also can detect single-bit errors and correct them and the can detect double bit errors and signal an alert that the error happened.

Because of the capabilities built into the bottom layer and the statistical properties and actions in the middle layer, the people working with classical computers can essentially focus on the computer science model of the computer and ignore the layers below that.

The description levels of *quantum* computers follow, by design, the same three-level pattern. There are some differences, some subtle that have been easily overcome, others are challenging.

[1] Modern CPUs have multiple parts that independently can execute gates; these parts are called "cores." More complex combinations of cores also exist: *graphical processor unit* (GPU) and *Field Programmable Gate Array* (FPGA).

Top - quantum computer science The *quantum processor unit* (QPU) operates on *quantum bits* (qubits). The generic implementation of a qubit is a quantum system with two states, denoted $|0\rangle$ and $|1\rangle$ to distinguish them from the classical bit that can be 0 or 1. The superpower of the qubit comes from the fact that it can not only be in one of these two states $|0\rangle$ or $|1\rangle$, but also in any linear combination like $a|0\rangle + b|1\rangle$, where a and b are complex numbers; the squares of these numbers need to add up to 1, in formula $|a|^2 + |b|^2 = 1$. For simplicity, you can think of real numbers between -1 and 1; you will not miss any of the essential features of quantum computing. The *quantum gates* operate on one qubit at a time or on a pair of qubits. All computing operations can be composed of such gates, just as in classical computer science.

Bottom - quantum mechanics The implementation of the quantum computer science model of a quantum computer relies on the same physical laws of QM as the classical computer. But there are some differences.

The first is that the devices to implement qubits and gates involves smaller numbers of atoms and electrons, or photons. The means that the classical limit of QM does not come into play. As noted above, that is important for quantum computers, but it makes making them much more delicate and subtle.

The second difference is that different qubits must interact in a subtle way, they must get "entangled" by the action of the quantum gates to get to the power of quantum computers. This is also different from classical computers where each bit is completely independent of all other bits at all times and during all gate operations.

Middle - statistics Because quantum computers rely on some of the features of quantum mechanics, the middle layer does not have the great capacity to buffer the computer science from the underlying phsyical laws. Some features of the physics must be dealt with in the quantum computer science model.

The external influences that cause errors in the states of qubits and the operations of quantum gates exist also in quantum computers. If anything quantum computers are a lot more seneitive to such disruptions, interferences, and noise.

The differences between classical and quantum computers stem from the difference in the gap, the middle, between the computer science model at the top and the quantum mechanics at the bottom: For classical computers this gap is big, it is wide enough to cross the classical limit of QM; for quantum computers there is no real gap because some of the features of quantum mechanics, like "entanglement," are essential to the functioning of a quantum computer and to its ability to significantly outperform classical computers. Because the operation of a quantum computer depends in an essential way on the phsyics features of QM, the middle layer overlaps the top and bottom layers, and the designers of quantum computers must deal with all layers at once. This is an important diference with classical computers, where the implementation in physics is cleanly separated from the computer science.

Historical precedent: space flight

The history of space flight provides a similar situation where initial, unmanned space flights were well-served by the phenomenological Kepler Laws of Planetary Motion, but manned space flight required solving Newton's dynamical equation.
Americans and Russians launched satellites in space in the late 1950's. The trajectories were determined using Kepler's Laws of Planetary Motion by joining parabolic and elliptic trajectories together at points where rocket motors made course corrections. The burns were short compared to the time it took to traverse the full trajectory, so the errors were acceptably small.
In 1961, Americans wanted to launch Alan Shepard in Mercury-Redstone 3 spacecraft and bring him back. It turns out that the error of where he would return in the ocean was about 100 miles, which would take the aircraft carrier several hours to cover. On an early American flight, the chimpanzee Ham almost drowned during recovery, because the craft landed too far from the recovery ship, which was at the predicted landing site. Proceeding as before, would put the floating astronaut at unacceptable risk because the error on the predicted trajectory, and landing site, were too big.
The needed accuracy for computing the landing location was beyond what could be accomplished by splicing Kepler trajectories. Newton's equation of force had to be solved numerically to get adequate accuracy. Katherine Johnson, the NASA "computer" (computers were people at that time) proposed to solve Newton's equations numerically with Euler's method to determine the trajectory with acceptable accuracy. This story is told in the 2016 film "Hidden Figures."

We will take a closer look at error correction as a didactic illustration of the difference between building a classical computer and a quantum computer. The example will also show the crucial role of the measurement process, which is the phenomenological part in the physical theory of QM, the part that is less understood than the dynamical law of the SE. The effort to build quantum computers started in earnest about 20 years ago. Only in the last couple of years have there been experiments that realize the implementation of encoding and correcting qubits [6]: Takada *et al.* on silicon spin qubits [7], Egan *et al.* on trapped-ion qubits [2], and finally Krinner *et al.*, Zhao *et al.*, and Livingston *et al.* on superconducting-circuit qubits [3, 8, 4].

Classical error correction Various influences, such as noise, cosmic radiation, and many more, can cause the physical device that hold a bit, at the value of 0 or 1, to flip that bit to the other possible value. To remidy that, the bit can be encoded in multiple physical bits with some property that can be used to check whether an error has occurred. Multiple ways to do this encoding exist, butthe simplest is to make three (3) copies. Then if one of them flips because of something affecting the computer, we can check all three and see that they are no longer the same. Then we use majority voting to dtermine the correct value and we set the bitthat was different back to the same as the other two. That is single-bit error detection and correction.

For this to work, we need to make sure that the probability that two bits erroneaously get flipped is much smaller than the probability only one gets flipped. That has been accomplished with modern clasical computer devices.

Quantum error correction The idea of error correction on qubits is essentially the same. But there is a difference that comes from the process of measurement in QM: If one measures the state of something in QM, one does not get the full picture: The state of a qubit, as we have seen is given by $a|0\rangle + b|1\rangle$. If we measure it we do not get these two numbers a and b, we happens is that randomly the qubit changes to either $|0\rangle$ with probability $|a|^2$ or $|1\rangle$ with probability $|b|^2$. That means the qubit state is changed, which we do not want to do.

Luckily there is a way around this: We can measure whether two qubits are the same without changing the state! The classical procedure measures all three physical bits 1, 2, and 3; the quantum procedure measures whether physical qubit 1 is equal to 2, and whether 2 is equal to 3. From that it can be determined which physical qubit has flipped: 1, 2, or 3. There is a way, that I will not go into, to flip back that changed physical qubit without changing the logical qubit state.

A quantum computer is, by its design specifications, very different from and much more complex than any experiment done in the first century of exploring quantum phenomena, including some of the most complex experiments like the Large Hadron Collider (LHC) and Laser Interferometer Gravitational-Wave Observatory (LIGO). All experiments carried out thus far have a clear stage where the outcome is measured once, which can be described by applying Born's rule to the final wavefunction, which is projected onto a subspace as part of the process. There is the worldwide effort to design, build, and test a working quantum computer, which has billion-dollar investments from governments and private industry. This effort is providing exciting opportunities for physicists and mathematicians to enter the field of quantum computing and it has engaged a new cohort of smart people, namely computer scientists. The design of quantum computers requires the engineering of systems that can, when full maturity is reached, to run quantum codes with billions of instructions on millions of logical qubits for hours at GigaHertz rates, which is the standard set by classical computers.

The Livingston group [4] introduces an interesting twist that is relevant to this note on measurement. They implement a process of coninuous measurement instead of the more traditional instantaneous projection-type of measurement to improve the stability and sustainability of maintaining the error correction-process longer, as will be needed for fully operational quantum computers.

The work is just starting: These acomplishments take care of error correction for the states of individual qubits. As Sean Carroll points out [1], the Born rule does not have the same status as the SE and is more like a phenomenological shortcut, albeit a very effective one. To guide the engineering and design of a quantum computer, the community in this massive effort will need a much more accurate and reliable handle on what happens during a measurement process than what Born's rule can provide. Useful computations on quantum computers require that many qubits, thousands and millions, are brought into a coherent state of entanglement and that that state is maintained while thousands and millions of gates operate on the qubits involved. The error correction of such a collection of qubits has not been addressed in the quantum computer science model and is far from being realized in an experiment.

References

[1] Carroll, S.: Addressing the quantum measurement problem. Physics Today **75**(7), 62–63 (2022). DOI 10.1063/PT.3.5046

[2] Egan, L., Debroy, D.M., Noel, C., Risinger, A., Zhu, D., Biswas, D., Newman, M., Li, M., Brown, K.R., Cetina, M., Monroe, C.: Fault-tolerant control of an error-corrected qubit. Nature **598**, 281–286 (2021). DOI 10.1038/s41586-021-03928-y

[3] Krinner, S., Lacroix, N., Remm, A., Di Paolo, A., Genois, E., Leroux, C., Hellings, C., Lazar, S., Swiadek, F., Herrmann, J., Norris, G.J., Andersen, C.K., Müller, M., Blais, A., Eichler, C., Wallraff, A.: Realizing repeated quantum error correction in a distance-three surface code. Nature **605**, 669–674 (2022). DOI 10.1038/s41586-022-04566-8

[4] Livingston, W.P., Blok, M.S., Flurin, E., Dressel, J., Jordan, A.N., Siddiqi, I.: Experimental demonstration of continuous quantum error correction. Nature Communications **13**(1), 2307 (7 pages) (2022). DOI 10.1038/s41467-022-29906-0

[5] Nielsen, M.A., Chuang, I.I.: Quantum Computation and Quantum Information, 10th anniversary edn. Cambridge University Press (2010)

[6] Saraiva, S.: The dawn of error correction with spin codes. Nature Materials (2022). DOI 10.1038/s41563-022-01415-x

[7] Takeda, K., Noiri, A., Nakajima, T., Kobayashi, T., Tarucha, S.: Quantum error correction with silicon spin qubits. Nature **608**, 682–688 (2022). DOI 10.1038/s41586-022-04986-6

[8] Zhao, Y., Ye, Y., Huang, H.L., Zhang, Y., Wu, D., Guan, H., Zhu, Q., Wei, Z., He, T., Cao, S., Chen, F., Chung, T.H., Deng, H., Fan, D., Gong, M., Guo, C., Guo,

S., Han, L., Li, N., Li, S., Li, Y., Liang, F., Lin, J., Qian, H., Rong, H., Su, H., Sun, L., Wang, S., Wu, Y., Xu, Y., Ying, C., Yu, J., Zha, C., Zhang, K., Huo, Y.H., Lu, C.Y., Peng, C.Z., Zhu, X., Pan, J.W.: Realization of an error-correcting surface code with superconducting qubits. Phys. rev. Lett. **129**(3), 030,501 (7 pages) (2022). DOI 10.1103/PhysRevLett.129.030501