1 The Splendors and Miseries of Quantum Computers

Moore's Law states that the number of transistors in computers we can build doubles every two years. This progress is only possible if we make transistors ever smaller. In 2017, the width of a transistor is at the scale of 10 nanometers, which corresponds to a layer of only 50 atoms in depth. Already at this scale, quantum effects, such as quantum tunnelling, become significant. Clearly with the trajectory of Moore's Law, our present paradigm for computer architecture will soon hit the wall, since the size of a transistor cannot possibly be smaller than the distance between atoms in a crystal. Moreover as we approach this barrier, quantum effects will become more prominent.

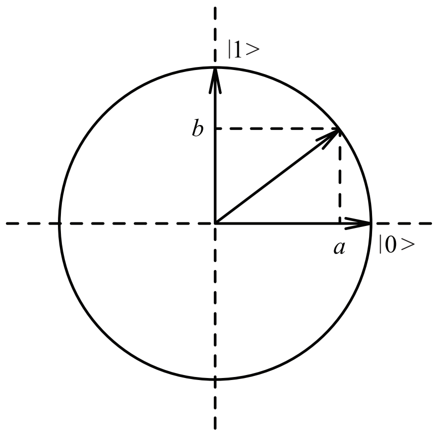
In our daily life, we deal with the objects that consist of many atoms (their number in a grain of sand is 10^{20}). In large collections of atoms, quantum effects get averaged out, and as a result we do not experience quantum mechanics with macroscopic objects. Yet quantum mechanics is increasingly present in our technology – such an ordinary thing like an LED flashlight, operates on quantum principles.

The idea of quantum computing is to embrace the bizarre quantum world, instead of fighting its influence. This is not easy, but there is a lot to gain. Quantum computers are devices that use quantum systems as processors.

Quantum computers have sound theoretical foundations in both physics and mathematics. However technological obstacles remain very serious, and a significant breakthrough is required. A lot of progress is also needed in developing quantum algorithms. In order to work on algorithms, one does not need access to a quantum computer, but only pen and paper, empowered with the knowledge of the theory of quantum computing.

Let us try to understand the difference between classical and quantum computers. In a classical computer data is stored in the memory as sequences of 0's and 1's. The unit of memory is called the bit, and it can store either 0 or 1. For the purpose of this discussion, it is useful to view 0 and 1 purely as symbols.

The unit of memory of a quantum computer is called the *qubit*, and it can store 0 and 1 simultaneously. More precisely, the value of a qubit is a vector with length 1 on a plane:



In order to make a connection with a classical bit, we label one coordinate axis with symbol 0 and the other axis with symbol 1. Accordingly, for the unit vectors on the coordinate axes we use notations which are traditional in quantum mechanics:

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

A qubit may be then written as a vector

$$\begin{pmatrix} a \\ b \end{pmatrix} = a\,|0\rangle + b\,|1\rangle\,,$$

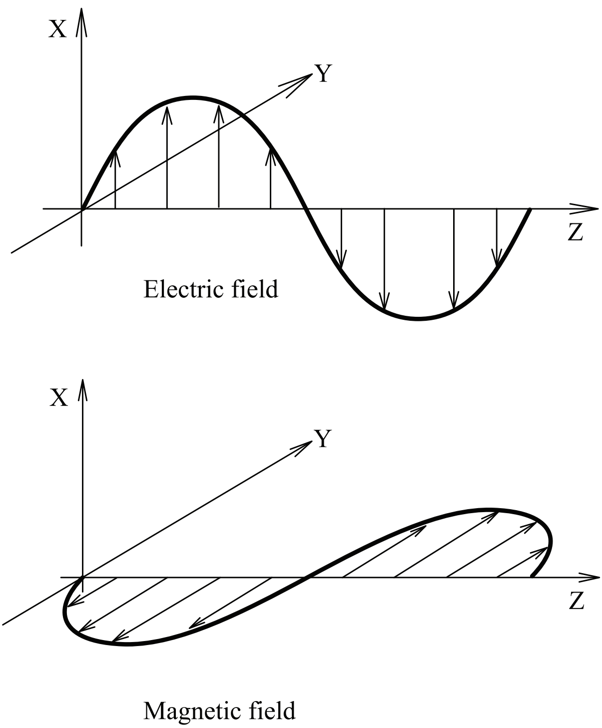
which is interpreted as a superposition of two classical bit values 0 and 1 with the weights a and b, where $a^2 + b^2 = 1$. We emphasize that $|0\rangle$ is not a zero length vector, but rather a unit length vector on an axis that is labelled with symbol "0".

When we build a computer as a physical machine, we need to use physical objects which can implement our abstract constructions of a bit and a qubit. A capacitor (an electronic device that can hold an electric charge) may serve as a unit of memory of a classical computer. A charged state of a capacitor represents 1, while discharged state represents 0.

A photon may serve as a physical realization of a qubit. A photon is a quantum of an electromagnetic wave. Imagine a photon flying in a 3-dimensional space along the Z-axis. As it propagates, electric field and magnetic field oscillate in mutually perpendicular directions in XY-plane.

The specific way how this oscillation occurs, is called the *polarization* of a photon. There are two kinds of polarization – circular and linear. In circular polarization, the electric field spirals around the Z-axis as the photon propagates. In this book we will only consider a simpler case of a linear polarization, when the electric field oscillates in a fixed direction perpendicular to the Z-axis. Linearly polarized photons may be obtained by passing a beam of light through a polarizing filter.

Polarized light is used in 3D movies. To create a 3D effect, left and right eyes should see slightly different pictures. The movie is shot with two cameras that are slightly apart. The images from both cameras are simultaneously projected on the movie screen, but the light from

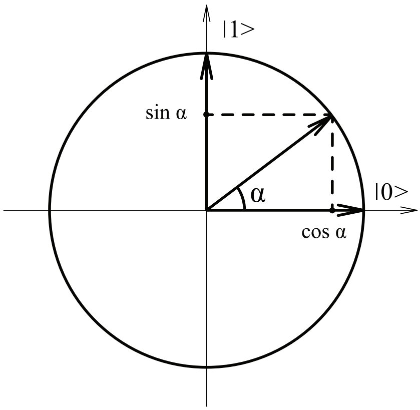


the two projectors are polarized in two different ways. The glasses have polarizing filters, each passing light only from one projector. As a result, two eyes receive distinct pictures, creating a 3D effect.

Imagine a photon with a linear polarization at an angle α in XY-plane. This photon can be used as a physical implementation of a qubit with value

$$\cos(\alpha)\left|0\right\rangle + \sin(\alpha)\left|1\right\rangle .$$

A small technical point about the photon states $\left|0\right\rangle$ and $-\left|0\right\rangle$. Both states correspond to photons with the same axis of polarization, however the oscillation of the electric field for $-\left|0\right\rangle$ occurs in *antiphase* relative to $\left|0\right\rangle$. Individually, these photons are essentially indistinguishable, however given a pair of such photons, we can detect the difference in phases, and view their states as two distinct qubit



values.

Now let us progress towards multi-qubits. A 2-qubit is a vector with 4 components of the form:

$$a_0 |00\rangle + a_1 |01\rangle + a_2 |10\rangle + a_3 |11\rangle ,$$

where $a_0^2 + a_1^2 + a_2^2 + a_3^2 = 1$. The basis vectors in the space of 2-qubits, $|00\rangle$, $|01\rangle$, $|10\rangle$, $|11\rangle$, also called *pure states*, correspond to the 4 possible classical values of 2-bit expressions. A general 2-qubit stores a combination (also called *superposition*) of the 4 classical 2-bit values simultaneously, with weights.

For example, the 2-qubit $0.3 |00\rangle + 0.1 |01\rangle + 0.9 |10\rangle + 0.3 |11\rangle$ is a superposition of all 4 classical values, but the 2-bit value "10" has a heavier weight in this 2-qubit.

As you might now guess, a 3-qubit is a vector with 8 components:

$$\begin{aligned} a_0 |000\rangle + a_1 |001\rangle + a_2 |010\rangle + a_3 |011\rangle \\ + a_4 |100\rangle + a_5 |101\rangle + a_6 |110\rangle + a_7 |111\rangle . \end{aligned}$$

Notice the pattern in our notations between the index of the coefficient "a" and the label of the corresponding pure state. Take the term $a_6 |110\rangle$, for example. Here "110" is the binary expression for the integer 6.

We can see that the number of terms in these expressions doubles with each additional qubit. Thus for an n-qubit the number of terms will be 2^n. An 8-qubit involves 256 terms:

$$a_0 |00000000\rangle + a_1 |00000001\rangle + a_2 |00000010\rangle + \ldots + a_{255} |11111111\rangle .$$

As an exercise, let us determine the index of "a" for the term with $|11010011\rangle$ in this expansion. The digits in a binary expansion correspond to powers of 2 (as opposed to powers of 10 in the decimal form). For an 8-bit expression, the leftmost digit corresponds to 2^7, while the rightmost digit corresponds to 2^0. We read the binary expression "11010011" as an integer

$$\begin{aligned} 1 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \\ = 128 + 64 + 16 + 2 + 1 = 211 \text{ (decimal).} \end{aligned}$$

In order to use more compact notations, we shall sometimes write an 8-qubit using a decimal form:

$$a_0 |0\rangle + a_1 |1\rangle + a_2 |2\rangle + \ldots + a_{255} |255\rangle .$$

Here we understand that all decimal integers appearing in the notations of basis vectors need to be converted to the 8-bit binary form.

As the number of bits increases, such expressions will become very long. An efficient mathematical way of writing such sums is to use the Σ notation. With this notation an 8-qubit is written compactly as

$$\sum_{k=0}^{255} a_k |k\rangle .$$

Here k is the index of summation and runs from 0 to 255, so the sum has 256 terms. When $k = 0$, it produces the summand $a_0 |0\rangle$, $k = 1$ yields $a_1 |1\rangle$, and so on. For each basis vector $|k\rangle$, the integer k is understood to be in an 8-bit binary form.

As we shall see in the next chapter, the joint polarization state of n interacting photons is described as an n-qubit. The amount of memory required to record such a state on a classical computer grows exponentially in n. If we allocate 1 byte to record the value of each "a" coefficient, then we need 2 bytes to store a 1-qubit, one kilobyte to store a 10-qubit, one megabyte to store a 20-qubit, one gigabyte to store a 30-qubit, one terabyte to store a 40-qubit, one petabyte to store a 50-qubit. If we take all the matter in the visible Universe, and make a giant memory chip based on today's approach to computer memory, we will not be able to store a 100-qubit on that device. At the same time, a collection of 100 interacting photons is something that may be everywhere around us. This realization led to inception of quantum computing.