There Is Still Plenty of Room at the Bottom: Feynman's Vision of Quantum Computing 65 Years Later

Alexis Lupo
*The University of Texas at El Paso, alupo@miners.utep.edu*

Vladik Kreinovich
*The University of Texas at El Paso, vladik@utep.edu*

Victor L. Timchenko
*Admiral Makarov National University of Shipbuilding, vl.timchenko58@gmail.com*

Yuriy P. Kondratenko
*Petro Mohyla Black Sea National University, y_kondrat2002@yahoo.com*

Follow this and additional works at: https://scholarworks.utep.edu/cs_techrep

Part of the Computer Sciences Commons, and the Mathematics Commons

Comments:
Technical Report: UTEP-CS-24-04

**Recommended Citation**
[https://scholarworks.utep.edu/cs_techrep/1860](https://scholarworks.utep.edu/cs_techrep/1860)

This Article is brought to you for free and open access by the Computer Science at ScholarWorks@UTEP. It has been accepted for inclusion in Departmental Technical Reports (CS) by an authorized administrator of ScholarWorks@UTEP. For more information, please contact lweber@utep.edu.
Abstract

In 1959, Nobelist Richard Feynman gave a talk titled “There’s plenty of room at the bottom”, in which he emphasized that, to drastically speed up computations, we need to make computer components much smaller – all the way to the size of molecules, atoms, and even elementary particles. At this level, physics is no longer described by deterministic Newton’s mechanics, it is described by probabilistic quantum laws. Because of this, computer designers started thinking how to design a reliable computer based on non-deterministic elements – and this thinking eventually led to the modern ideas and algorithms of quantum computing. So, we have a straight path of speeding up computations: by learning how to use molecules, atoms, and then elementary particles as building blocks of a computational device. But what if we reach the size of an elementary particle? At first glance, it may seem that we will then reach an absolute limit of how fast a computer can be. However, as we show in this paper, we can potentially speed up computations even further – by using the internal structure of elementary particles: e.g., the fact that protons and neutrons consist of quarks. Interestingly, the corresponding mathematics is very similar to what is called color optical computing – the use of light of different colors in computations.

Keywords: faster computers, Feynman, quantum computing, color optical computing, quarks.
1 Introduction

Computers are very fast. Computers now are several orders of magnitude faster than a few decades ago. An average laptop – like the one on which we type this text – performs 4 billion operations per second, much more than record-breaking supercomputers a few decades ago.

A simple calculator has more computing power in it than all the computers that supported the 1960s going-to-the-Moon Apollo program.

But not fast enough. In spite of all these successes, one of the main challenges that computing faces is that for many practical applications, the current computer speed is not sufficient. The media is full of descriptions of how the speed of current high performance computers limits our ability to further improve Large Language Models like the famous ChatGPT.

But to explain the need for faster computing, one does not have to cite such futuristic examples: the need for faster computing is ubiquitous in many real-time control applications, where we need to make control decisions really fast; see, e.g., [1, 4].

How can we further speed up computers? Engineers working on computer design focus on specific technical issues: e.g., how to minimize the undesirable interactions between two neighboring transistors. But, in addition to technical issues, there are also fundamental issues that make speeding up computers difficult. One such issue is the fact that, according to special relativity theory (see, e.g., [3, 6]) information cannot be transferred faster than the speed of light.

It is easy to see that this limitation directly affects the computer speed. For example, the laptop of which we type this text has a diagonal size of about 30 cm; this is the largest distance between the two points from the laptop. If we divide 30 cm by the speed of light – which is approximately 300 000 km/sec – we can conclude that it takes at least 1 nanosecond for a signal to go between these two points. During this time, our 4GHz laptop already performs 4 operations. So, a natural conclusion is that if we want to make computers faster, we need to make computers smaller in size – and this, in turn, means that all the computer components must be made smaller. And this is exactly the tendency that computer designers have been following.

Feynman’s 1959 talk. Nobel-winning physicist Richard Feynman was the first who, in his famous 1959 talk titled “There’s plenty of room at the bottom” [2], explained natural consequences of this need for miniaturization. According to Feynman, since we need to have smaller and smaller components to design faster and faster computers, eventually, we will have to decrease the size of these components to the sizes comparable with the sizes of individual molecules.

And here comes an important challenges. At the usual macrolevel – in particular, at component components sizes that were prevalent at that time – the vast majority of events in the computing process can be described by Newton’s mechanics. One of the important features of Newton’s mechanics is that it is deterministic: once we know the initial state, we can uniquely predict
all future states. A good illustration to this determinism is celestial mechanics, the first area to which Newton applied his mechanics: we can predict Solar eclipses hundreds years ahead, and what we observe now, in the 21 century, is in perfect accordance with predictions made several centuries ago.

This determinism is a perfect fit for what we want for computer components: we want computers to perform the same sequence of computations every time we need it, we do not want to have different answers to the same computational problem. And here lies a challenge. Already molecules – and even more atoms and elementary particles – cannot be fully described by Newtonian physics. To adequately describe the behavior of molecules, atoms, and elementary particles, we need to use quantum physics; see, e.g., [3, 6]. And one of the features of quantum physics is that it is non-deterministic: we cannot predict what exactly will happen in the future, we can only predict the probabilities of different futures. For example, if we take an atom of a radioactive material such as Uranium (U), we cannot predict when exactly it will decay – we can only compute the probability that it will decay before a given moment of time.

So Feynman formulated a challenge to computer designers: once components reach the size at which quantum effects need to be taken into account, how can we design a deterministic reliable computer out such non-deterministic unreliable elements?

Feynman’s idea eventually led to modern quantum computing. When a Nobel prize winner talks, people listen. So researchers started thinking about how a reliable computer can be built from quantum components. And they succeeded, even beyond the original Feynman’s expectations: not only quantum algorithms lead to deterministic answer, some of these algorithms are actually much faster than the corresponding deterministic ones – so, in addition to a speedup causes by a decrease in size, we have an additional speedup caused by using quantum physics. Here are the two classical examples (for details, see, e.g., [5]).

The first example is searching for an element with the desired property in an unsorted list. If we have a list with $n$ elements then, in a non-quantum world, the only thing you can do is search the elements one by one. If you are in luck, you can find the desired element at the very first attempt. However, no matter in what order you test these elements, in the worst case, the last element you search is the desired one. So, in the worst case, the time needed for this search is proportional to the number of elements $n$. Interestingly, by using quantum effects, Lov Grover came up with an algorithm that requires, in the worst case, time proportional to the square root of $n$ – which is, of course, much faster. For example, in case we have a million records, we have $\sqrt{n} = 1000$, so the search time decreases by a factor of thousand.

The second example is a fast quantum algorithm developed by Peter Shor for factoring large integers. This may sound like a not-very-useful theoretical problem, but it is actually very practically important, since most current encryption algorithms for computer-based communications are based on the understanding that factoring large integers is very complex. For example, the most widely used
RSA algorithm uses a product \( n = p_1 \cdot p_2 \) of two large prime numbers \( p_1 \) and \( p_2 \) for encoding.

- If one knows the values \( p_1 \) and \( p_2 \), then decoding is easy.
- However, in situations when we do not know the factors, when we know only their product, all known non-quantum algorithms require unrealistic astronomical time to decode.

In contrast, Shor’s quantum algorithm finds the factors in feasible time – provided, of course, that we have a universal quantum computer with a sufficient number of quantum bits (also known as qubits). This may not be the most interesting application of quantum computing, but the possibility to be able to read all secret messages is the main reason why many governments invest billions of dollars in quantum computing research.

**Limits to growth.** This all sounds good: we can go from using molecules as computational elements to using atoms to using elementary particles – and each time we drastically decrease the size of computing elements, we drastically speed up computations.

But there seems to be a limit: once we reach the level of elementary particles, there is no way to go: by definition, elementary particles are the end in itself, they cannot be divided into smaller parts. In Feynman’s words, once we reach the level of elementary particles, there seems to be no remaining room at the bottom.

So what can we do?

**There is still plenty of room at the bottom: what we do in this paper.** In this paper, we show that, contrary to the above first impression, it is potentially possible to further speed up computations. In other words, we will show that there is still plenty of room at the bottom.

## 2 Our Idea

**Let us recall some physics: the basic facts about quarks.** To explain our idea, we need to recall basic facts about such elementary particles as protons and neutrons; see, e.g., [3, 6].

Yes, as we have mentioned earlier, these are elementary particles:

- we can divide an atom into electrons, protons, and neutrons,
- but we cannot divide a proton or a neutron into smaller particles.

On the other hand, we all have read in popular articles that protons and neutrons consist of special particles called *quarks*. It is important to understand that this “consists of” has a different meaning that a similar statement that an atom consists of elementary particles:

- an atom *can* be divided into elementary particles, but
• a proton or neutron cannot be divided into quarks.

We can actually observe quarks by experiment similar to what Rutherford did at the dawn of atomic physics to show that an atom consists of a nucleus and other particles floating in empty space. When he bombarded a layer of material with energetic alpha-particles, in most cases, they went through or deviated a little bit, but in some cases, they were reflected almost 180 degree and moved back. This showed that most of the space in a solid body is practically empty, but there are few areas with high concentration of matter than lead to a strong reflection.

Similarly, if we bombard protons with high-energy particles, there are a few cases when we get a very strong reaction. This shows, kind of, that inside a proton there are dense areas – known as partons – surrounded by (almost) empty space. These partons is what physicists call quarks. A proton or a neutron consists of three quarks. (It should be mentioned that there are other particles – e.g., pions, responsible for strong interactions – that consist of two quarks.)

Just like a nucleus forms a very small part of the corresponding atom, partons occupy a very small part of the particle. In this sense, the partons (quarks) are very small, much smaller than elementary particles themselves.

In contrast to particles in an atom – that can be separated – quarks cannot be separated from each other. The possibility to separate elementary particles from each other is based on the fact that for interactions between these particles – be they electromagnetic interactions or “strong” interactions between protons and neutrons – the interaction force decreases with distance. So, once we apply some energy and move the particles sufficiently far away from each other, it is sufficient to apply a relatively small force and separate them completely.

In contrast, as experiments showed, if we try to increase the distance between quarks, the force pushing them back does not decrease at all. So, no matter how far away we move them apart, they will still be connected as forcefully as before.

**Quark colors.** The observed interaction between quarks seems to imply that each flavor of quarks can be of three different types (and an anti-quark can be of one of the opposite types), so that an elementary particle:

• either has three quarks of different types,

• or has three anti-quarks of different anti-types,

• or has a quark or some type and an anti-quark of the corresponding antitype.

Physicists observed that this is somewhat similar to three basic colors: red (R), green (G), and blue (B), with anti-colors interpreted as filters stopping the corresponding color. In this interpretation, the only way to avoid colors – and to get while or black – is:

• either to combine all three colors,
• or to apply all three filters,
• or to combine a single color with the corresponding filter.

So, there is a natural analogy between quark types and colors, in which elementary particles – formed by quarks – cannot have color, they have to be black or white.

Because of this analogy, quark types are usually called \textit{colors} – but it is important remember that beyond this simple analogy, quark types have nothing to do with the actual basic colors – which correspond to electromagnetic waves of certain frequencies.

\textbf{So how can we potentially use quarks for computations?} Quarks are smaller than elementary particles. So if we can use quarks for computations, we can definitely speed up computations. But how can we do it?

How can we use any device for computations? Performing computations means changing some states. So, to be useful for computing, an object has to have at least two different states – otherwise, if the object has only one possible state, we cannot do anything with it. The simplest case is when we have exactly two different states. This way, we can use this object to represent a bit: we associate one of its states with 0, and another states with 1. If an object has more than 2 states, we may be able to use to represent 2 or more bits.

From this viewpoint, let us consider two states of each of the three quarks that form a proton or a neutron: a usual state and an excited state. Then, depending on which of three quarks are excited, we get the following 8 different options – which we will describe by the combination of colors of excited quarks:

• it may be that none of the quarks is excited; in terms of the color analogy, this corresponds to black; we will denote this proton state by B;
• it may be that only the red quark is excited; we will denote this proton state by R;
• it may be that only the green quark is excited; we will denote this proton state by G;
• it may be that only the blue quark is excited; we will denote this proton state by B;
• it may be that only red and green quarks are excited; we will denote this proton state by \(R + G\);
• it may be that only red and blue quarks are excited; we will denote this proton state by \(R + B\);
• it may be that only blue and green quarks are excited; we will denote this proton state by \(G + B\);
• it may be that all three quarks are excited; in terms of the color analogy, this corresponds to white; we will denote this proton state by W.
In our analogy, states of anti-proton correspond to similar filters. Mixing of the two colors means that the two protons interact, so that one of them stops being excited and the excitation energy moves to the other proton.

This becomes very similar to color optical computing. At first glance, the above scheme sounds very complex, and it is not immediately clear what we can do with these 8 states. Good news is that this has already been studied for actual colors: namely, several schemes have been proposed to use combinations of basic colors for computations (see, e.g., [7, 8, 9, 10, 11, 12]) – and some of these schemes turned to be useful for such practical tasks as providing safety of ship navigation [8].

So, our idea is to apply this computational scheme to quark colors.

3 Conclusion

No matter how fast our computers have become, there is still a practical need for faster computations. To speed up computations, we need to make computer components smaller in size. In 1959, exactly 65 years ago, Nobel-winning physicist Richard Feynman presented a talk titled “There’s plenty of room at the bottom” (published in 1960), in which he explored this idea and predicted that eventually, we will have to bring the size of our components to the size of molecules – and then atoms and elementary particles. To do that, said Feynman, we need to take into account that at these sizes, Nature follows not deterministic Newton’s laws, but probabilistic laws of quantum physics. So, we need to think how to perform deterministic computations on a computer consisting of such non-deterministic components. Computer researchers solved this challenge – and Feynman’s 1960 paper is now considered one of the foundational papers of the field of quantum computing, with all its spectacular theoretical (and potentially practical) successes.

This is all good, but as we progress in this direction, we seem to hit a limit to the possible computer speed up: we will get a drastic speedup when we get to the level of atoms and then elementary particles, but after that, there seems to be nowhere to go, there seems to be no room left at the bottom.

In this paper, we show that we can go further, we can potentially achieve further speedup, that there is still plenty of room at the bottom. Specifically, we show that:

- while we cannot divide an elementary particle like proton or neuron into smaller parts,
- we can use the fact that, in some reasonable sense, a proton and a neutron consist of three “sub-particles” – quarks.

While these sub-particles cannot be separated, they have a certain autonomy within a proton or a neutron.
We propose to use this autonomy and thus, to use quarks for computations – since quarks are smaller than the elementary particles formed from them, this will potentially lead to an additional computational speedup.

Interestingly, because of the known analogy between quark types and basic colors – because of which quark types are called quark colors – the quark-based resulting computational environment is similar to the computational environment of color optical computing, when we use light of different colors for computations. We can therefore use computational techniques from color optical computing for quark computations as well.

Acknowledgments

This work was supported in part by the National Science Foundation grants 1623190 (A Model of Change for Preparing a New Generation for Professional Practice in Computer Science), HRD-1834620 and HRD-2034030 (CAHSI Includes), EAR-2225395 (Center for Collective Impact in Earthquake Science C-CIES), and by the AT&T Fellowship in Information Technology.

It was also supported by a grant from the Hungarian National Research, Development and Innovation Office (NRDI).

References


