



Imagining the Future of Quantum Computing

Erik P. Debenedictis

► To cite this version:

Erik P. Debenedictis. Imagining the Future of Quantum Computing. Leon Strous; Roger Johnson; David Alan Grier; Doron Swade. Unimagined Futures – ICT Opportunities and Challenges :, AICT-555, Springer International Publishing, pp.70-83, 2020, IFIP Advances in Information and Communication Technology, 978-3-030-64245-7. 10.1007/978-3-030-64246-4_6 . hal-03194441

HAL Id: hal-03194441

<https://inria.hal.science/hal-03194441>

Submitted on 9 Apr 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Imagining the Future of Quantum Computing

Erik P. DeBenedictis¹

¹ Zettaflops, LLC, Albuquerque, NM 87112, USA
erikdebenedictis@zettaflops.org

Abstract. The world embarked on an information revolution almost a century ago and has been driven ever since by the well-known evolution of mainframes into Smartphones. While the exponential performance improvements due to Moore’s law have slowed, quantum information concepts will drive the information revolution still further. Although the ultimate structure of a quantum computer is evolving, it seems destined to fit into the familiar program-based paradigm while solving some problems at vastly greater scale and speed than previously possible. Imagine a future “general” computer combining both bits and qubits operating at multiple temperatures yet based on familiar principles in electronics and computer architecture. Also imagine a future where some quantum information principles are taught to children and then become a part of society’s thinking, much as place value numbers became a part of everyday life over the last two millennia.

Keywords: Quantum computer, microprocessor, architecture, software, programming, qubit.

1 From the History of Computing to Its Unimagined Future

Computing changed the world over the last century through a combination of faster hardware, more memory, and software that automated an increasing range of activities. The shrinkage of microelectronics and the accompanying increase in energy efficiency made computers ever more capable. The rate of improvement has slowed as technology approached the physical limits of the underlying computing devices. Yet roadmaps [1] still project improvements going forward of an order of magnitude for hardware and somewhat more for architectures and software.

However, quantum computing offers additional orders of magnitude improvement based on a different technical principle. The original type of computer, henceforth called a classical computer, can answer a question by scanning input data very quickly and then computing an answer. In contrast, a quantum computer can answer some types of questions without looking at every item in the input data, or at least not looking at the input data in the way humans’ non-quantum eyes look at things. This has obvious speed advantages for large data sets, but it comes at the price of the answer being based on probabilities. We will discuss this new type of information later in this chapter through an analogy to Powerball (lottery) tickets.

1.1 A General Computer, Where General Is Classical Plus Quantum

Fig. 1 imagines the future of what I call a general computer, or one with a seamless merging of classical and quantum concepts. I am imagining society will move past the accelerator model in fig. 1a where a traditional computer has sub modules connected to it, such as a floating-point and quantum accelerators, and to the view in fig. 1b where the general computer uses either bits or qubits as needed. While the new quantum technology may be transparent to end users, specialists may spend the next century refining it.

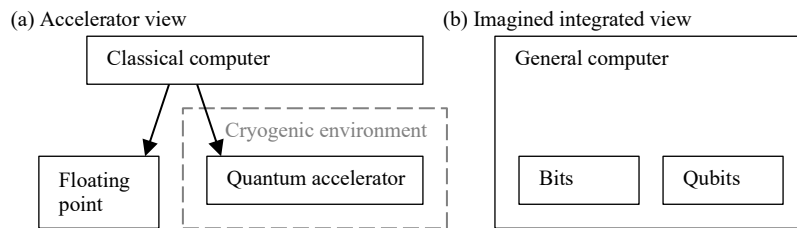


Fig 1. Floating point and quantum (a) as accelerators vs. (b) integrated. Floating point moved from accelerators to a basic data type in microprocessors long ago, just as in the unimaginable future bits and qubits will coexist in the same computer.

A recent quantum computer demonstration [2] pitted what was essentially a one-chip quantum accelerator against the world's largest supercomputer and found a significant speed and energy efficiency advantage for the quantum chip. The demonstration was controversial because the example problem was a natural fit for the quantum computer but was extremely hard for the supercomputer. Yet the demonstration showed the quantum chip had an asymptotic advantage as the problem scaled up, so there is the possibility of the advantage becoming astronomical over time.

Imagining the future of quantum computing will require identifying how far the new methods push out the boundary of what is computable. For example, we have a fairly good idea of the maximum size of a number that can be factored using a classical computer, but we do not know the equivalent size limit for a quantum computer.

Likewise, a classical computer can simulate quantum chemistry using a variety of heuristic algorithms, the results of which enhance our lives through new or improved products based on chemistry. We are sure quantum computers will be better at simulating chemistry, but we do not know how much better or the nature of the future advances that will result.

2 Quantum Information and Place Value Numbers

A lot of attention has been devoted to making a computer outperform the previous generation of itself on benchmarks, but quantum computers may change society in a more fundamental way.

Computers are essential to today's definition of success both directly and indirectly. Design and simulation of aircraft and motion pictures containing computer graphics are

examples of products that directly use large amounts of floating-point arithmetic. Yet we use spreadsheets to indirectly assess the profitability of businesses.

2.1 Place Value Numbers

Industry was not always assessed this way. Our current place value number system is only two millennia old. Before then, arithmetic was performed on unary numbers comprising piles of stones, written marks, and so forth. A hunter might draw five deer on a cave wall to document the fact that they obtained five deer for food. If we could go back in time and ask hunters if they would prefer to use Excel spreadsheets instead of cave walls because a spreadsheet could represent up to 10^{38} objects, I am sure they would not understand what we are talking about because it was not known at the time that numbers could be that big.

However, the larger range of place value numbers allowed society to understand the relationship between the number of people needing to eat and the total amount of food available, including meat and other foods. This ultimately led the deer hunter to become part of a food industry, which, along with other industries, changed civilization. Yet with only a unary number system, the deer hunter did not have the mathematical background to understand the concept of industry, and without the concept of industry the hunter would not recognize the value of place value numbers.

2.2 Bits to Qubits

Switching from bits to qubits will present similar issues. Qubits have a bit value, making them backwards compatible with bits, but qubits may also exist in probabilistic states such as zero half of the time and one the other half of the time. While these would be considered errors in today's computers, we now know that some problems can be solved much more efficiently by appropriately using the quantum properties of qubits.

Computers perform a lot of stock trades nowadays, making money for the quantitative trader, but also increasing the efficiency by which society allocates capital.

So, imagine that a quantitative trader learns to use a quantum algorithm to predict the future price of a food industry stock with 90% certainty, where the best classical algorithm is only 51% certain. The quantum-enabled trader would then make more money with less working capital than competing traders, possibly starting an "arms race" where all traders would switch to quantum computers. Yet, the fact that quantum computers would be optimizing the distribution of capital to the food industry would make that industry more efficient over time and thus benefit society.

Neither profit and loss spreadsheets nor quantum computers improve upon deer, but the ancient deer hunters' descendants may end up competing based on quantum computers optimizing the way they do business.

2.3 Simulations

People designed just about everything with pencil and paper until computers became powerful enough to make computer-aided design practical. Computers can assist in

designing an aircraft, for example, not only by automating steps previously performed by humans, but by adding new steps, such as putting computer simulation into a loop that optimizes the shape of the wing.

However, many physical simulations in chemistry and biology are too hard for classical computers, ultimately preventing computers from assisting technological development in those fields. This is due to exponential run time for classical algorithms simulating some complex chemicals. There is a theoretical basis for quantum computers having an advantage, but the advantage relies on leaving place value numbers behind and using probabilistic qubits.

Yet unlike the deer hunter, biology provides us with a tantalizing example of what is possible. Biological evolution has explored the possibilities of a carbon-based set of chemical elements, DNA as a storage media, codons as an alphabet, and so forth. We know that chemistry can lead to life and intelligence. Yet due to the limitations of classical computers, we cannot modify life very much or develop alternative approaches. However, quantum computers could be instrumental in creating new products on top of existing life, such as new medicines.

We could also use quantum computers to simulate and design new products from inorganic chemicals, such as better batteries. This would not be building on top of what life has already created but rather using life as inspiration for developing chemistry based on different assumptions.

2.4 New Thinking

While a quantum computer is an improved classical computer, it is also a computer for quantum information. The biggest opportunity is if society starts thinking in new ways due to quantum information. This does not mean people will think like a quantum computer, but rather that people will understand what a quantum computer can do and make sense of its results for other purposes.

Perhaps instead of doing profit and loss statements at the end of a quarter, businesses would also do profit and loss statements for the next quarter based on quantum computer optimizations. Perhaps experimental science will become more like aircraft design, where scientific discoveries will be “designed” on a quantum computer and checked with experiments.

This could change society in unimaginable ways, but our descendants might thank us anyway.

3 Education

We teach our children mathematical concepts that eluded even the greatest adult minds two millennia ago, such as place value number systems. Can we imagine a future where children are taught enough about quantum information to apply it to everyday tasks?

Just as most users of spreadsheets cannot write a spreadsheet application in, say, C++, it will not be necessary for most users of quantum computer applications to know how to build or program a quantum computer.

There are many ways to visualize quantum information, yet children will need a form that does not have extensive mathematical prerequisites. I've used a popular lottery in the United States called Powerball as proof that many people can appreciate correlated probabilities [3], which is all that is necessary to apply the results of quantum computing.

3.1 Powerball and Quantum Computing

The buyer of a Powerball ticket picks five numbers between 1 and 69, as illustrated in fig. 2a. At a designated time, the Powerball operator conducts a random drawing of five numbers between 1 and 69, with no duplicates. Powerball has multiple games rolled into one ticket, but in one game the ticket wins if three numbers are common to both sets, as shown in fig. 2b.

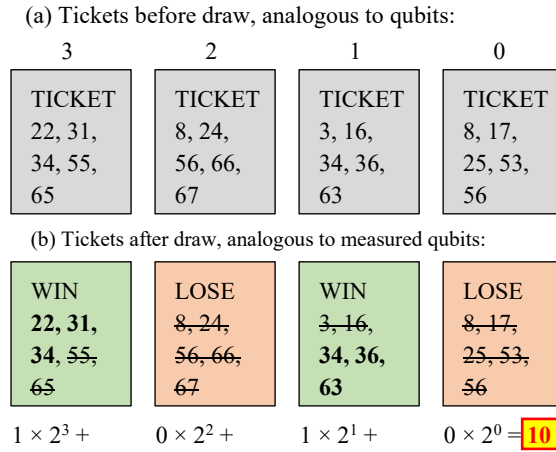


Fig. 2. User view of quantum program computing 10. (a) Qubits are like lottery tickets with pick numbers that are only of interest to programmers. (b) The drawing picks 22, 31, 34, 36, 63 and the pattern of winning tickets reveals the number 10 when viewed as a binary number.

Powerball tickets and qubits both have two phases in their lives. A ticket is characterized by “pick” numbers during the first phase of its life. The second phase starts with the drawing, after which the pick numbers become irrelevant and all that matters is whether a ticket won or lost. Fig. 2b shows the winning/losing tickets with corresponding binary values 1/0 creating the multi-bit binary number 10.

A ticket viewed in isolation either wins or loses at random, yet tickets with identical pick numbers will win or lose at the same time. If two tickets share fewer than five numbers, their probabilities of winning or losing will become correlated.

Qubits have a similar life cycle. Quantum computers create qubits in a standard form, which the physicists call $|0\rangle$. To create a custom ticket, or qubit, single-qubit quantum gates effectively modify the pick numbers. To enable computation, two-qubit quantum

gates can mix up the pick numbers, although some of these gates increase the size of the pick number space multiplicatively, such as a ticket with 4 pick numbers and another with 8 yielding two tickets with 32 each. The drawing, which physicists call qubit measurement, yields a result that has a lot of randomness but also has correlations due to the past application of quantum gates. A quantum computer does not use the specific Powerball rules, such as numbers between 1 and 69, nor can a lottery compute, but Powerball tickets represent the probabilistic nature of quantum information.

It may not have occurred to a deer hunter that putting a deer symbol (digit) to the left of another one would make it represent ten deer, but today's children easily learn that concept.

The Powerball example incidentally addresses a broad concern that place value numbers tend to be interpreted too precisely. Say you fill your car with 38.2 liters of gasoline and note from your odometer that you drove 426 km. After a division, you say your car gets 11.15183246 km/liter, which overstates precision. The Powerball number system analogy has uncertainty built into it.

Can we teach children to be comfortable with quantum computers spitting out random bits or numbers, yet accepting that the computers' output is the solution to an important problem?

4 Unimagined Products from Quantum Linear Algebra

Let me use linear algebra to illustrate the progression of a technology from scientific discovery to products, for both classical and quantum computing. Table 1 (adapted from [4, fig. 3]) illustrates the progression for the purposes of this chapter.

Table 1. Levels of Classical and Quantum Technology Development

Job title	Function
A. Hardware engineer	Creates quantum devices and a classical control system
B. Physicist	Designs pulse sequences to drive a quantum circuit
C. Information scientist	Creates a quantum circuit to create desired output
D. Software developer	Uses a quantum system to run an application and find a solution
E. Domain expert	Uses applications to create value in areas outside of computation

The ability of a computer to solve matrix equations, called linear algebra, is an enabler for many higher-level applications, so classical and quantum linear algebra should be of interest to IFIP readership. I will explain the progression based on my personal experience with a microwave simulator called the High Frequency Structure Simulator (HFSS) [5].

4.1 HHL Quantum Algorithm

Harrow, Hassidim, and Lloyd (HHL) [6] discovered a quantum algorithm, or circuit, in 2009 that gives exponential speedup on linear algebra problems. The HHL algorithm

solves $Ax = b$, where A is an $N \times N$ matrix, b is an N -element input vector, and x is an N -element output vector. The algorithm runs in $O(\log N)$ time.

I owe the reader an explanation because the algorithm counterintuitively runs in less time than it takes to write down the answer. The algorithm computes x as a quantum superposition of all the elements in the output vector x . The superposition can be thought of as a group of $\lceil \log_2 N \rceil$ Powerball tickets, as illustrated in fig. 2. The win/lose pattern after the drawing reveals an index i . If the algorithm is run many times, the probability of i appearing is the vector value x_i .

Skipping forward to 2020 when this chapter was written, the HHL algorithm is mentioned in textbooks and source code is available [7] for both classical simulators and quantum hardware. Students can run HHL on quantum computers available on the Internet up to size $N=2$.

While quantum simulators should be able to solve larger problems, one textbook included the explanation “[i]n fact, the overheads required by [the eigenvalue inversion] step are a contributing factor to why the full code sample of even the simplest HHL implementation currently falls outside the scope (and capabilities!) of what we can reasonably present here” [7-8].

While students can experiment with HHL, current quantum code has many limitations, which ref. [8] describes as “the fine print.” This phrase refers to the fine print in contracts, where contracts that look like a good deal on the surface have fine print at the bottom of the page that makes it difficult for a person to actually get the value they expect from them.

An example of the fine print for the HHL algorithm is the fact that it does not compute the entire vector x , but instead requires the user to embed HHL into a larger problem that can make use of probabilistic samples of index i . If expanded to 53 qubits, this sampling is exactly what happened in the quantum supremacy experiment [2].

4.2 Sparse Matrices

I recall taking a class on sparse matrix algorithms as a master’s student in 1980. The sparse matrix algorithm for solving $Ax = b$ was about as mature as HHL is now, although there are significant differences. The sparse matrix methods in that era involved storing matrix elements efficiently, such as linked lists of rows. The sparse matrix algorithm suppressed multiplications by zero but otherwise performed the same arithmetic operations as dense matrix algebra. Students could experiment with sparse matrix algorithms in homework assignments about as easily as they can with HHL today.

After improvements to HHL in 2010 [9] and 2013 [10], in 2016 the U. S. government funded an assessment of the resources required for a quantum linear algebra solution to a radar scattering problem [11]. The assessment counted the number of quantum gates as a function of the size of the problem N , concluding that the number of quantum gates would be less than the number of classical gates for problems above the size $N = 332,020,680$ (which is huge). At that size, the circuit would include 3.34×10^{25} gates applied to 340 qubits (exclusive of Oracles, which are beyond the scope of this chapter).

Setting aside parallelism, an Exascale supercomputer executes 10^{18} floating point operations per second and each floating-point operation involves about 10^5 gate

operations, so an Exascale supercomputer performs 3.34×10^{25} classical gate operations in about 5 minutes.

The assessment pointed out that an algorithm developed in 2015 [12], which was apparently not ready to be used for resource estimation, should allow a “reduction of circuit depth and overall gate count by order of magnitude $\sim 10^5$ ”—or potentially cutting run time to about 3 ms based on our loose analogy to an Exascale supercomputer.

So, the limitations of HHL, or the fine print [8], made radar scattering computations on a quantum computer impractical. Yet, the degree of impracticality dropped by algorithmic improvements in 2010 [9], 2013 [10], 2015 [12], and others not mentioned. If improvements continue to reduce gate count in steps of 10^5 , quantum linear algebra will be practical before long.

4.3 Finite Elements and Solid Modelers

Getting back to my personal experience, I learned about the finite element method in my class in 1980, but I could not use it because it was too complicated for a student to code in a homework assignment. Yet over several decades, finite elements went from 2D to 3D and became adaptive, meaning that the elements were dynamically resized for higher accuracy near features where a lot of physics was in play. Solid modeling front ends were developed once graphical user interfaces became effective on personal workstations around 1990. This led to the HFSS product where a student or an engineer could design a 3D structure on a workstation and simulate its microwave response. HFSS is in use today for many things, including simulating transmon quantum computer systems.

However, finite elements and solid modelers are not linear algebra algorithms; they are technologies further down table 1. Classical and quantum linear algebra algorithms correspond to row C of table 1. Once the technical community was confident with sparse matrix algorithms, new research went to finite element methods, at row D of table 1, which solves differential equations, and so forth.

Many quantum algorithms have limitations like HHL [8], but classical algorithms have limitations as well. Even though bits are different than qubits, I’m suggesting that the process of computer technology maturing is the same in both cases.

I cannot imagine what specific quantum applications will be discovered, but quantum machine learning looks like a promising candidate. Instead of the finite elements in row C of table 1, there is a framework for a neural network.

The process of working down the list at the top of the section applies to other quantum algorithm classes as well, such as period finding (Shor’s algorithm for factoring), simulation of quantum physics, and optimization.

Readers seeking more insight may find [8] helpful. I have tried to illustrate solutions to the issues Scott Aaronson found in the fine print.

5 Integrating Bits and Qubits

This section assumes the physics of quantum computing will be further refined and then engineered into computers. The physicists' view of quantum computing has always presumed a close coupling of the quantum hardware with a classical computer, yet this section views qubits and quantum gates as second computational technology that will be integrated with Complementary Metal Oxide Semiconductor (CMOS) gates to create a "general" computer as shown in fig. 1b.

This section treats qubits like Dynamic Random-Access Memory (DRAM) cells. DRAM cells are distinct from CMOS gates in function, structure, and electrical interfaces, but are often collocated on the same chip.

5.1 Qubits vs. Transistors

Qubits are different than transistors but are on a similar evolutionary trajectory.

The transmon superconducting qubit [13] is perhaps the quantum analogy to Transistor-Transistor Logic (TTL), a semiconductor logic family from the 1970s.

Over time, the TTL logic family spawned variants such as low power Schottky TTL (LS TTL), which could be analogous to superconducting qubits with a name other than transmon, like capacitively shunted flux qubits [13]. TTL's silicon bipolar transistor was ultimately replaced by a different type, although it was not known in advance whether the successor would be Gallium Arsenide or CMOS.

Superconducting and spin qubits must be cooled to 10-20 mK for operation today whereas ion traps work best when cooled to around 4 K. There are ideas for room temperature qubits, but qubits seem more sensitive to thermal noise than transistors, so we cannot count on room temperature qubits just because they would be convenient.

It seems inevitable that transmons will be improved and the improvement will have a different name. However, trapped ions, spin qubits, or a room-temperature qubit could replace superconducting qubits in general, yet nobody knows which will be analogous to Gallium Arsenide (which was never successful in computers) or CMOS (which is preeminent).

5.2 The Chandelier and Physical Structure

The physical structure of a superconducting quantum computing accelerator is very different from standard computer packaging. The quantum computing structure is called a chandelier due to physical appearance and is shown diagrammatically in fig. 3.

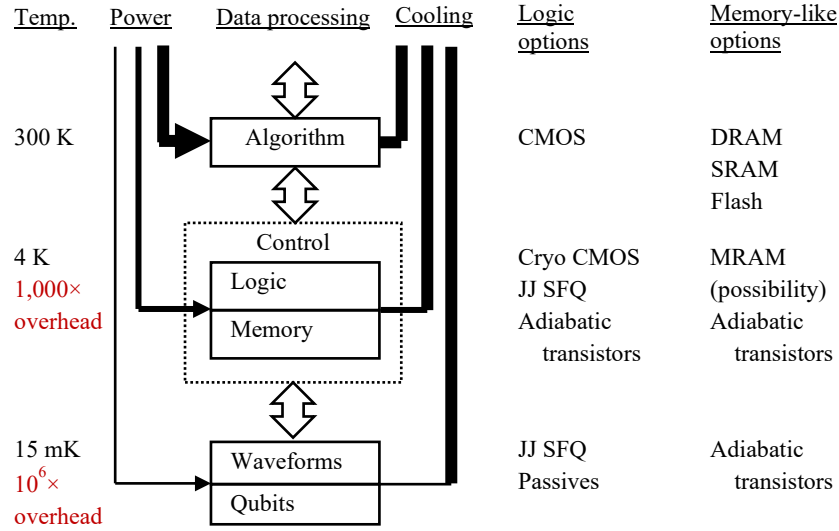


Fig. 3. Chandelier structure. (vertical) Classical and quantum processing is distributed across temperature stages, exposing issues of (left) cooling overhead, (center) power generated by control electronics and leaked on cables crossing temperature gradients. (right) Options for logic and memory vary by temperature. Acronyms: JJ = Josephson junction, SFQ = Single Flux Quantum, SRAM = Static Random Access Memory.

Qubits are in a 10-20 mK temperature stage due to physical requirements. The classical control system is distributed across progressively warmer stages, such as 4 K, until reaching room temperature (300 K) where there is a computer of conventional design. Design issues unique to qubits are listed in fig. 3, such as cooling overhead, heat and noise flow between stages, and the device options available at each temperature stage.

The current structure evolved cryogenic apparatus for physics experiments yet is moving towards a scalable computer architecture. For example, a recent paper adapts Rent's rule to quantum computers [14].

5.3 Minimum Dissipation of Classical Computation

People who are familiar with microprocessors will recall that their clock rate and performance rose over time until the early 2000s, at which point top-of-the-line microprocessors had a 4 GHz clock and were dissipating ~200 W per chip. While laptop and supercomputer system performance continues to rise over time, a close look reveals that the additional performance is from the graphical processing units (GPUs). Since the microprocessor is essentially mature, it will be used as the base classical technology for the imagined future computer.

The physics of computation includes concepts and tools for understanding classical computers. At the low level, Landauer [15] made a physics-based argument that the heat generated by AND or OR gates must be on the order of kT per (irreversible) operation. Engineers know that the practical minimum is hundreds of times larger due to

wire losses, variances in device manufacturing, noise, and so forth. Yet it is a good approximation to say that a microprocessor's heat dissipation is the product of the dissipation of a single gate operation multiplied by the number of gate operations used by an algorithm.

5.4 Mixed Classical-Quantum Circuitry

Future computer engineering will need to extend the reasoning in the last paragraph to include qubits. The straightforward extension of Landauer's minimum energy to quantum computers would conclude that qubits do not create any heat. Without debating the previous point, we will see below that many qubits will need to be partnered with classical logic, leading to much the same effect as if there were a minimum energy per qubit, or qubit gate. There are several cases:

1. Data must be converted between quantum and classical forms at the input and output of quantum algorithms. Translating bits to qubits requires applying electrical waveforms (or sometimes laser signals) to the qubits. The reverse translation is called quantum measurement. Both involve classical electronics on one side.

2. Large-scale quantum computers will require continuous quantum error correction of all qubits, not just the ones used for I/O. Errors in qubits are revealed by a quantum measurement that must change the state of a classical state machine so that it knows to correct the error.

3. Some quantum algorithms can be performed more efficiently if they include a measurement step and possibly a loop, which will be explained below.

All the cases above lead to a mixed classical and quantum sub circuit, where the classical portion is governed by Landauer's minimum dissipation. For example, fig. 4 illustrates a circuit to rotate a qubit $|\Psi\rangle$ by a small angle of about θ^2 . The method is to execute the quantum circuit in fig. 4a repetitively until the measurement returns a 1 [16].

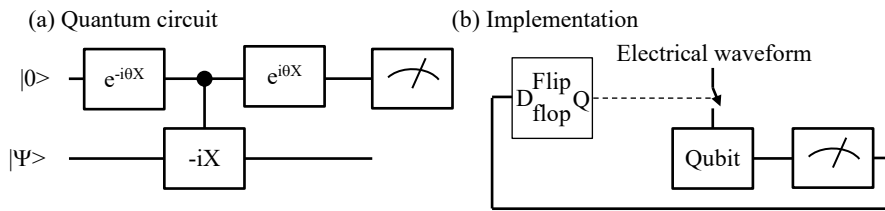


Fig. 4. (a) If 1 is measured, the circuit will have rotated $|\Psi\rangle$ by a small angle of about θ^2 , otherwise a correction must be applied to $|\Psi\rangle$ and the process repeated. (b) The implementation requires a flip flop, qubit, an electrical switch, and measurement circuit in the cold environment. See [16] or <https://www.youtube.com/watch?v=zwBdwCoVmSw> @5:56 for more information.

Imagine applying the rotation in fig. 4b to many qubits at once. The electrical waveform can be created at room temperature and routed to many qubits in parallel with near zero heat dissipation. Yet each instance of the circuit performs a measurement that is latched into a per-qubit classical flip flop. When the measurement returns a 1, the flip flop

blocks the electrical waveform for that qubit only. The flip flop is governed by Landauer’s minimum heat dissipation, and that heat will be multiplied by the overhead factor of the cooling system.

If the classical electronics were not collocated with the qubit, a separate microwave cable to room temperature would be needed for each qubit, leading to heat backflow and other issues that would limit scalability.

While there is an argument that qubits do not create any heat, many qubits will be associated with a classical circuit that must dissipate heat. This leads to a minimum dissipation per quantum operation that is somewhat analogous to Landauer’s minimum dissipation per classical operation.

The thermodynamics of the “general” computer in fig. 1b has not been fully developed, but we can imagine a time when we will be able to say how many joules will be required to solve a problem, such as solving a linear algebra problem or performing a quantum simulation.

6 New Issues in Computer Architecture

While CMOS is being proposed for classical control systems, its requirements are different from a microprocessor and should lead to different tradeoffs in device optimization and architecture. Table 2 contains simple models of classical and quantum computer throughput and cost.

Table 2: Energy Efficiency and Throughput Tradeoffs

	Classical	Classical control system	Quantum
Throughput	$N_G f_{\text{clk}}$	$N_q S(N_q) f_{\text{clk}}$	
Cost of Ownership	$N_G (\$_G + \$_e)$	$N_q k (\$_G + \$_e 300 \text{ K}/T_q)$	$+ N_q \$_q$
N_G, N_q	= number of gates, qubits		$S(N_q)$ = quantum speedup
$\$_G, \$_q$	= cost of gate, qubit		T_q = qubit temp
$\$_e, \$_R$	= lifetime cost of energy, refrigerator		f_{clk} = clock rate

Let us first look at the throughput row in table 2. Classical computer throughput, such as a microprocessor, is proportional to the number of gates N_G times the clock rate f_{clk} . This motivates engineers to clock microprocessors as fast as possible.

However, a quantum computer’s throughput $N_q S(N_q) f_{\text{clk}}$ also depends on the quantum speedup, $S(N_q)$, which can vary between 1 and 2^{N_q} . With a little thought it’s clear that the engineer has a lot more to gain by raising speedup. Doubling a quantum computer’s clock rate doubles the throughput but doubling the number of qubits or quantum gates can create as much as an exponential increase in throughput. Of course, the quantum speedup is dependent on the algorithm.

Now let us look at the cost of ownership row in table 2. The cost of ownership for a classical computer contains the term $(\$_G + \$_e)$, the cost of buying the computer in the first place plus the lifetime cost of energy. While it is socially appropriate to save energy, the cost of energy is well under the original purchase price of most computers. This creates an economic disincentive to employ energy savings technologies

particularly if throughput decreases. So, auto-sleep mode on laptops is acceptable but not reversible logic, because the latter requires reducing the clock rate during computation.

However, a cryogenic classical control system contains the term $\sum (\$_G + \$_e 300 \text{ K}/T_q)$. This multiplies the effective cost of energy by $300 \text{ K}/T_q$, a factor in the range $10^3 \dots 10^9$. This makes energy saving technologies that were uneconomical at room temperatures into huge winners at cryogenic temperatures. Reversible logic is such a technology.

The microprocessor has been one of the biggest economic drivers of all time, essentially imposing its requirements on both the computer and software industries. Some computer design principles carry over from classical to quantum computing, yet design principles related to energy efficiency and throughput will change. These decisions are fundamental in classical computer design, so changing the decisions for quantum computer design will require rethinking many aspects of computer engineering.

6.1 Quantum Error Correction and Computer Aided Quantum Computer Design

Current quantum computer technology is called Noisy Intermediate Scale Quantum (NISQ) [17], which is defined as raw physical qubits without quantum error correction. Raw qubits are analog devices that naturally accumulate error over time. While decoherence time, essentially the average time before an error, varies between qubit types, the leading quantum computer demonstration to date [2] can only perform about 40 gate operations before the result becomes meaningless due to noise. Quantum error correction will create logical qubits from multiple physical qubits, allowing an essentially unlimited number of gate operations.

We have discussed qubits as though they were a non-CMOS classical device, such as DRAM bit or a sensor element. The engineering processes for generating control signals for DRAM have been built into tools like CACTI [18], allowing a DRAM block to be integrated into a higher-level design through the same interfaces as a digital circuit.

It seems inevitable that somebody will write a CACTI-like program for mixed classical-quantum circuits, ultimately becoming a computer aided design tool for what I call “general” computers. Writing such a tool will be quite a challenge—particularly considering the need to cope with multiple temperatures—but this section should at least give an idea of where to start.

7 Conclusions

Until a year ago, skeptics postulated that progress in quantum computing might be blocked by unanticipated challenges in the physics. The recent demonstration by Google [2] pitting a single quantum computer chip against the world’s largest supercomputer alleviated this concern significantly.

With well-studied potential quantum chemistry simulations, cryptographic, and some other applications, it is likely that multiple parties worldwide will make the investment to at least evaluate the engineering challenges in building a large-scale quantum computer.

I suggest that such a system should be called a “general” computer, as it must contain novel classical, quantum, and integrated classical-quantum technology.

Qubits may evolve much as transistors evolved in the history of computing. However, the physical architecture would have some novel properties, including spanning a temperature gradient and two technologies for computation (classical gates and qubits).

If such a computer can be built, its application to physics simulation is likely to lead to new materials, biotechnology advances, as well as advances in other areas. The potential advantage of quantum computers in optimization and machine learning is tantalizing and could develop into other important application areas.

The result could change the way society thinks more broadly. Society already knows how to think precisely using place value numbers, but place value numbers may be augmented by the probabilistic data that emerges from a quantum computer.

References

1. International Roadmap for Devices and Systems, <https://irds.ieee.org>, last accessed July 2020.
2. Arute, Frank, et al. Quantum supremacy using a programmable superconducting processor. *Nature* 574.7779 (2019): 505-510.
3. DeBenedictis, Erik P. Powerball and quantum supremacy. *Computer* 52.10 (2019): 110-112.
4. Córcoles, Antonio D., et al. Challenges and Opportunities of Near-Term Quantum Computing Systems. *Proceedings of the IEEE* (2019).
5. Ansys HFSS Homepage, <http://www.ansys.com>, last access July 2020.
6. Harrow, Aram W., Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical review letters* 103.15 (2009): 150502.
7. Johnson, Eric R., Nic Harrigan, and Mercedes Gimeno-Segovia. *Programming Quantum Computers: Essential Algorithms and Code Samples*. O'Reilly Media, Incorporated, 2019.
8. Aaronson, Scott. "Read the fine print." *Nature Physics* 11.4 (2015): 291-293.
9. Ambainis, Andris. Variable time amplitude amplification and a faster quantum algorithm for solving systems of linear equations. *arXiv preprint arXiv:1010.4458* (2010).
10. Clader, B. David, Bryan C. Jacobs, and Chad R. Sprouse. "Preconditioned quantum linear system algorithm." *Physical review letters* 110.25 (2013): 250504.
11. Scherer, Artur, et al. Concrete resource analysis of the quantum linear-system algorithm used to compute the electromagnetic scattering cross section of a 2D target. *Quantum Information Processing* 16.3 (2017): 60.
12. Berry, Dominic W., Andrew M. Childs, and Robin Kothari. Hamiltonian simulation with nearly optimal dependence on all parameters. 2015 IEEE 56th Annual Symposium on Foundations of Computer Science. IEEE, 2015.
13. Kjaergaard, Morten, et al. Superconducting qubits: Current state of play. *Annual Review of Condensed Matter Physics* 11 (2020): 369-395.
14. Franke, David P., et al. Rent's rule and extensibility in quantum computing. *Microprocessors and Microsystems* 67 (2019): 1-7.

15. Landauer, Rolf. Irreversibility and heat generation in the computing process. *IBM journal of research and development* 5.3 (1961): 183-191.
16. Wiebe, Nathan, and Vadym Kliuchnikov. Floating point representations in quantum circuit synthesis. *New Journal of Physics* 15.9 (2013): 093041.
17. Preskill, John. Quantum Computing in the NISQ era and beyond. *Quantum* 2 (2018): 79.
18. Wilton, Steven JE, and Norman P. Jouppi. CACTI: An enhanced cache access and cycle time model. *IEEE Journal of Solid-State Circuits* 31.5 (1996): 677-688.