

## Chapter 7, Director's Cut: It from Qubit

The universe is indistinguishable from a quantum computer. If you look at it closely, you will find that it is made of atoms, electrons, photons, and other elementary particles. Each of these particles registers information, and their interactions process information. The information that these particles register and process is quantum mechanical information. At bottom, the universe can be thought of as performing a quantum computation. In addition, the behavior of elementary particles can be mapped directly onto the behavior of quantum bits that are interacting via quantum logic operations. The resulting quantum computation is indistinguishable from the universe. The conventional view is that the universe is nothing but elementary particles. That is true, but it is equally true that the universe is nothing but bits, or rather, nothing but quantum bits.

Under the dictum that if it walks like a duck and quacks like a duck, it's a duck, from this point on we'll adopt the position that since the universe registers and processes information like a quantum computer, and is observationally indistinguishable from a quantum computer, then it is a quantum computer. But just how does the universe compute and what is it computing? This chapter will describe in detail the microscopic mechanisms by which the universe computes.

To give you this description, I will depart from the ordinary conventions of popular science writing, and present you with simplified versions of the actual scientific papers in which I derived those mechanisms. Normally it is thought that scientific papers in specialized journals are out of the reach of ordinary readers. I do not think that this is or should always be the case. To help you along if you don't like math, I'll let you know what any equations mean. I have been told that every equation in a popular science account reduces the readership by 50%. Don't worry about the equations. Equations in scientific texts remind me of the Peanuts cartoon in which Lucy comes across her younger brother Linus reading *The Brothers Karamazov*: "Don't all those long Russian names bother you?" she asks. Linus replies, "Oh no. When I come to one I can't pronounce, I just *bleep* right over it." Please bleep right over any equations that you would like to in these papers: you will find that most of the technical parts have already been described in words above, or will be explained by my commentary. Anyway, you would be surprised at how often scientists themselves bleep over when they read a technical paper.

My motivation in presenting you with the original works on the computational universe is twofold. First, I'd like you to see what scientific papers are like: they're not so bad,

really. Second, I'd like to emphasize that although the results that this book discusses are certainly new science, they are not a New Kind of Science, to quote the title of Stephen Wolfram's popular book. I have a great deal of respect for Wolfram's book: it is substantial and weighty. In this book, published last year, Wolfram also advocates thinking about the universe in computational terms: he and I are in agreement that one should think about the universe in terms of computation. We differ, however, in that he would like to think of the universe as a classical computer, whereas I have shown that the universe is performing a computation that is intrinsically quantum mechanical.

The idea that the universe might be a classical computer was proposed in the 1960's by Ed Fredkin, one of the founders of physics of information, and independently by Konrad Zuse, the first person to build a modern electronic computer. Fredkin and Zuse suggested that the universe might be a type of computer called a cellular automaton, consisting of a regular array of bits interacting with their neighbors by logical operations. More recently, Stephen Wolfram has extended and elaborated Fredkin and Zuse's ideas. But given how bad classical computers are at reproducing features such as entanglement and quantum dynamics, even for very simple systems, it is hard to see how the universe could be a classical computer such as a cellular automaton. If it is, then the vast majority of its computational apparatus is inaccessible to observation.

Although I disagree with Fredkin, Zuse, and Wolfram on the underlying computational mechanisms of the universe, I still recommend them to the reader for their perspective. They may be barking up the wrong tree, but they're in the right forest.

Wolfram and I also differ in the way in which scientific advances should be presented to the public. His New Kind of Science attempts to present new scientific advances directly in popular form, without providing the detailed technical explanations that allow other scientists to check his results, and to trace the detailed set of ideas that influence them. I although I do new science, I practice an old-fashioned *kind* of science: I believe in publishing articles in journals, where they can be reviewed by other scientists prior to publication. As this book indicates, I also believe in giving popular accounts of that science.

Why do I believe in the system of review for scientific results? Scientific knowledge consists exactly of results that can be reproduced by others. Before being published as science, a result should be reviewed to make sure that it is in principle reproduceable. That is not to say that the reviewer must necessarily reproduce the result before publication: in the case of many experiments, reproducing them will take significant time and future

effort. If those experiments reproduce the result, then the result remains in the body of scientific knowledge.

The published result need not be correct. If someone tries to reproduce it, and shows that the result is wrong, that's good, too: the fact that the result is wrong is now part of the body of scientific knowledge. For example, a few years ago a rapidly rising young scientist was found to have fabricated results. Many of these fabricated results appeared in major journals. The cry went up that the system of peer review of scientific articles was flawed. In fact, the system worked well, if a little slowly: first, other scientists tried without success to reproduce the results; then, his papers were scrutinized more closely; finally, clear evidence of fraud was uncovered (graphs of data from two entirely different experiments were found to be exactly identical, an experimental impossibility given the presence of random noise in experimental results). The mills of Science grind slowly, but they grind exceedingly fine.

That's not to say that the system of anonymous peer review, in which papers submitted to a journal are reviewed by anonymous scientific peers, is all fine and dandy. The system affords many opportunities for abuse: for example, an unscrupulous reviewer can block publication of a correct paper until the reviewer's own paper on the subject is published (this happens!). Still, the system of peer review resembles democracy: to paraphrase Winston Churchill, peer review is the worst possible system, until one considers the alternatives.

The anonymous nature of the peer review process is important for protecting younger researchers who are commenting on the papers submitted by established scientists. But comments delivered anonymously can lack subtlety. The first paper that I ever submitted, an article on entanglement and black holes, was rejected by Physical Review A after the reviewer (whom I later discovered was a Nobel laureate) submitted the following review: "There is no physics in this paper." Ouch!

In fact, my personal experience of the peer review process is that the more original and imaginative the paper that I submit, the more quickly and harshly it is rejected. One must be willing to stand up and fight for one's ideas. For example, I once came up with a simple and novel idea for controlling quantum systems. Feedback is a common method of control in which the controller gets information about the system to be controlled, processes the information, and applies a control on the basis of the results of information processing, i.e., the controller 'feeds back' the processed information to the system. A thermostat is a

simple example of feedback: a thermometer monitors the temperature, and if it is too hot, the thermostat turns off the heat; if it is too cold, it turns it back on. In the conventional picture of feedback control of quantum systems, the controller makes a measurement on the system, processes the results of the measurement, and then feeds back the processed information. This type of quantum control has two drawbacks. First of all, because it involves quantum measurement, this type of quantum feedback is inherently chancy (God does play dice). In addition, the measurement intrinsically disturbs the system while getting information about it: quantum measurement destroys the original state of the system and replaces it with a different state.

My idea was a simple one. Suppose that the quantum feedback controller never performs a measurement, but instead interacts with the quantum system to obtain quantum information. It then processes that quantum information using a quantum computer, which could be quite simple. Finally, it feeds the quantum information back by interacting with the system in a way that preserves quantum coherence. In other words, such a coherent quantum feedback controller gets, processes, and feeds back qubits rather than bits. It's not hard to show that its use of quantum information allows such a coherent quantum feedback controller to do things that can't be performed by a feedback controller that gets classical information via quantum measurement. (I did not know it at the time, but this idea of coherent quantum feedback had previously been discovered by Harold Wiseman and Gerard Milburn in Australia.)

Since the idea was a simple and novel one, I immediately wrote up a letter and submitted it to *Science* magazine. Several months later, the referee report returned: the idea was clearly novel and interesting, but because it relied on the speculative technology of quantum computation, it could not be implemented. Ha! I knew the referee was wrong. In fact, David Cory's simple NMR quantum computers were perfectly capable of implementing coherent quantum feedback. Within a few weeks, my graduate students Richard Nelson, Yaakov Weinstein, together with David and me, had implemented coherent quantum feedback control of a nuclear spin. The quantum feedback controller consisted of two other nuclear spins in the same molecule, and quantum information was shunted around the molecule and processed coherently using nuclear magnetic resonance. And they said it couldn't be done!

Back we went to *Science* with our experiment. A month or two later, we got back the referee report, which this time clearly came from an NMR experimentalist. "I do not

understand the intention of this experiment,” the report read, “but given how trivial it was to implement, the result cannot possibly be important.” Rejected. We had passed from impossible to trivial without ever passing through publishable. (A year and a half later, after another set of clashes with referees, we finally managed to get this work published in *Physical Review*.)

One must expect rejection. In some rare cases, if one is sufficiently masochistic, one can even savor it. Before returning to the Computational Universe, let me present one final referee report, rejecting a further paper on the role of information in control. Though it was painful to have months of work rejected in such an extreme way, it was almost worth it given the remarkable humiliation inflicted by the referee (I have omitted purely technical criticisms from the report to focus on the most embarrassing parts):

“It is a pity that such a well-written and erudite paper, that is not devoid of substance, offers so little content of interest. The authors attempt to characterize well-known control-theoretic concepts (such as controllability, observability, and stability) in terms of information theoretic entropies and mutual informations. Like many others before them, the authors fall victims to the allure and beauty of Information Theory concepts. By failing to realize that, unless operational sense is made out of these quantities, the discussion stays in the sphere of philosophical discourse, they engage in correct, but meaningless, observations and in correct, but useless, mathematical inferences.

Although they illustrate their (correct) concepts by means of simple illustrative examples, there is simply no useful insight, guideline, or application that emerges from this essay. The parallels between error control coding and control or between rate distortion and control are superficial and sterile (though correct!).

In fact, the only statement of some consequence to control theory (and by that I mean the only derivation that follows from the authors’ information-theoretic modeling, that says something not obvious or trivial about controller structure, even though it is not translated into any concrete conclusions) is theorem 10. Unfortunately, that is already published by the authors and the only new thing here is an alternative proof and its imbedding in a more general discussion framework.

Another debilitating feature of the paper is the extreme didactic and pedantic style. The erudite language is used unproductively to magnify trivial or obvious statements and to make them appear as monumental and profound observations. The mere length of these discussions is overbearing. Coupled with the shallowness of meaning it becomes

excruciating. Yet, as I said earlier, there is nothing incorrect in the paper and in fact the viewpoint of the authors regarding the control system structure is of some interest.

So, what to do? I suggest that the authors scale back significantly their own view of their work. It may sound patronizing and harsh but it is sound advice. If they choose to do so, they can rewrite their paper as a short technical note (no more than five-six pages) in which, without much ado, they introduce their control system model and quickly state and prove their most significant theorems on stability, controllability, and observability. No philosophical digressions, no speculative cross-disciplinary pronouncements, and no connection to continuous variables, thermodynamics, and the like. Perhaps, before engaging in a revision, the authors may want to consult Shannon's short but famous "Bandwagon" article (IT Transactions, 1955). "

It hurts! It hurts! What would you have done if you received such an erudite rejection? My co-author and I lightly revised the paper to make it less didactic and pedantic (we definitely did not cut it down from its original twenty-five pages to five). Then we submitted it to another, equally prestigious journal, where it was immediately accepted. Whatever the complaints about its style, the paper was, after all, correct.

Sometimes peer review is not anonymous. On occasion, the referee will reveal his or her identity and correspond directly with the author to explain the referee report in greater detail. When I submitted my 1993 paper to Science showing for the first time how a quantum computer could actually be constructed, I received a response from Rolf Landauer directly. As noted above, Landauer was one of the founders of the field of physics of information processing. He and I had been friends since he had invited me to come to IBM Watson Laboratory to talk about Maxwell's Demons. He was the first person ever to offer me a job in science (and as noted above, also one of the last). Rolf was a delightfully gruff, not to say grumpy person. Once, when he was having dinner at our Santa Fe, his chair was placed directly in front of an obtrusive table leg. When Eve tried to make him move, he barked, "I prefer to be uncomfortable!"

Rolf could be uncomfortable in a very productive way. He had decided ideas on what was and what was not possible. He claimed that noise and imperfections made quantum computation impossible in practice. I claimed that although noise certainly caused problems for quantum computation, its effects could be counteracted and overcome to some degree. We had a heated dispute. When the time came to turn in the final version of the paper, I asked Rolf how he would like me to reference his papers. He replied, irascibly,

that since I had paid no attention to his arguments he didn't see why I wanted to reference them at all. So I gave the final version of the paper to Science without any references to Landauer's work. When the paper appeared, Landauer exploded: it seems that his testy comment had not meant that he wanted none of his papers to be referenced, but rather that he wanted *all* of his papers to be referenced. I hastily composed an addendum to the paper in which I referenced every paper in which Landauer had ever mentioned quantum computation.

The first paper that will be presented below, 'Ultimate physical limits to computation,' benefited from a referee who disclosed his identity. The referee was Lev Levitin, of Boston University, one of the world's experts on physical limits to communication. Together with Norm Margolus, one of the founders of the physics of computation, Levitin was the author of the Margolus-Levitin theorem, which plays a central role in limiting the speed with which a computer can compute. After giving a positive referee report which suggested revisions, Levitin dogged each revision with further suggestions and comments, all of which improved the final paper.

In the following paper, which appeared in the journal *Nature* in the the year 2000, I'll insert comments in italics. I have simplified the paper in order to make it more accessible. I'll let you know when you might want to bleep over something. When you do, just bleep to the next section or to the next italicized comment. If you don't want to read the paper, then just read the abstract and bleep over the rest. I hope that you find, however, that the what you have learned so far in this book should allow you to understand most if not all of the contents.

## **Ultimate physical limits to computation**

*A paper should have a stirring title.*

Seth Lloyd

d'Arbeloff Laboratory for Information Systems and Technology

*I am a member of many laboratories. I chose to list this one because Alex d'Arbeloff is a*

*delightful person to have lunch with.*

MIT Department of Mechanical Engineering  
MIT 3-160, Cambridge, Mass. 02139  
slloyd@mit.edu

*The next paragraph is called the Abstract: it says briefly what the paper is about.*

**Computers are physical systems: what they can and cannot do is dictated by the laws of physics. In particular, the speed with which a physical device can process information is limited by its energy and the amount of information that it can process is limited by the number of degrees of freedom it possesses. Here I explore the physical limits of computation as determined by the speed of light  $c$ , the quantum scale  $\hbar$  and the gravitational constant  $G$ . As an example, quantitative bounds are put to the computational power of an ‘ultimate laptop’ with a mass of one kilogram confined to a volume of one liter.**

*Now we come to the introduction: it says more about what the paper is about.*

Over the past half century, the amount of information that computers are capable of processing and the rate at which they process it has doubled every two years, a phenomenon known as Moore’s law. A variety of technologies — most recently, integrated circuits — have enabled this exponential increase in information processing power. There is no particular reason why Moore’s law should continue to hold: it is a law of human ingenuity, not of nature. At some point, Moore’s law will break down. The question is, When?

The answer to this question will be found by applying the laws of physics to the process of computation<sup>1–86</sup>. *This paper has 99 references: typing them all in took a long time.* Extrapolation of current exponential improvements over two more decades would result in computers that process information at the scale of individual atoms. Although an Avogadro scale computer that can act on  $10^{23}$  bits might seem implausible, prototype quantum computers that store and process information on individual atoms have already been demonstrated<sup>64–65,76–80</sup>. Existing quantum computers may be small and simple, and able to perform a few hundred operations on fewer than ten quantum bits or ‘qubits,’ but the fact that they work at all indicates that there is nothing in the laws of physics that forbids the construction of an Avogadro-scale computer.

The purpose of this article is to determine just what limits the laws of physics place on the power of computers. At first, this might seem a futile task: because we don't know the technologies by which computers one thousand, one hundred, or even ten years in the future will be constructed, how can we determine the physical limits of those technologies?

*Bleep over the numbers in the next two sentences if you like. The important point is merely that the fundamental constants of Nature determine the ultimate physical limits to computation.*

In fact, as will now be shown, a great deal can be determined concerning the ultimate physical limits of computation simply from knowledge of the speed of light,  $c = 2.9979 \times 10^8$  meters per second, Planck's reduced constant,  $\hbar = 1.0545 \times 10^{-34}$  joule seconds, and the gravitational constant,  $G = 6.673 \times 10^{-11}$  meters cubed per kilogram second squared. Boltzmann's constant,  $k_B = 1.3805 \times 10^{-23}$  joules per degree Kelvin, will also play a key role in translating between computational quantities such as memory space and operations per bit per second, and thermodynamic quantities such as entropy and temperature. In addition to reviewing previous work on how physics limits the speed and memory of computers, I presents new results — which are new except as noted — of the derivation of the ultimate speed limit to computation, of trade-offs between memory and speed, and of the analysis of the behavior of computers at physical extremes of high temperatures and densities are novel except as noted.

Before presenting methods for calculating these limits, it is important to note that there is no guarantee that these limits will ever be attained, no matter how ingenious computer designers become. Some extreme cases such as the black-hole computer described below are likely to prove extremely difficult or impossible to realize. Human ingenuity has proved great in the past, however, and before writing off physical limits as unattainable, one should realize that certain of these limits have already been attained within a circumscribed context in the construction of working quantum computers. The discussion below will note obstacles that must be sidestepped or overcome before various limits can be attained.

*OK so far? It's really not so bad. Now the fun starts.*

## **1. Energy limits speed of computation**

To explore the physical limits of computation, let us calculate the ultimate computational capacity of a computer with a mass of one kilogram occupying a volume of one

liter, roughly the size of a conventional laptop. Such a computer, operating at the limits of speed and memory space allowed by physics, will be called the ‘ultimate laptop.’ (Figure 1)

(Figure 1: The ultimate laptop. The ‘ultimate laptop’ is a computer with a mass of one kilogram and a volume of one liter, operating at the fundamental limits of speed and memory capacity fixed by physics. The ultimate laptop performs  $2mc^2/\pi\hbar = 5.4258 \times 10^{50}$  logical operations per second on  $\approx 10^{31}$  bits. Although its computational machinery is in fact in a highly specified physical state with zero entropy, while it performs a computation that uses all its resources of energy and memory space it appears to an outside observer to be in a thermal state at  $\approx 10^9$  degrees Kelvin. The ultimate laptop looks like a small piece of the Big Bang.)

*The equations in the next paragraph are just a mathematical expression of the fact that it takes energy to make things happen. The more energy you have, the faster you can make things happen. Quantum mechanics, in the form of the Margolus-Levitin theorem mentioned above, puts precise limits on how fast you can make things happen with a given amount of energy.*

First, ask what limits the laws of physics place on the speed of such a device. As I will now show, to perform an elementary logical operation in time  $\Delta t$  requires an average amount of energy  $E \geq \pi\hbar/2\Delta t$ . As a consequence, a system with average energy  $E$  can perform a maximum of  $2E/\pi\hbar$  logical operations per second. A one kilogram computer has average energy  $E = mc^2 = 8.9874 \times 10^{16}$  joules. Accordingly, the ultimate laptop can perform a maximum of  $5.4258 \times 10^{50}$  operations per second.

### 1.1 Maximum speed per logical operation

For the sake of convenience, the ultimate laptop will be taken to be a digital computer. Computers that operate on non-binary or continuous variables obey similar limits to those that will be derived here. A digital computer performs computation by representing information in the terms of binary digits or bits, which can take the value 0 or 1, and then processes that information by performing simple logical operations such as *AND*, *NOT* and *FANOUT*. (*FANOUT* is a fancy name for a *COPY* operation, so called because in an electronic computer *COPY* is performed by making wires ‘fan out’ from a given wire. The actual physical device that performs a logical operation is called a logic gate. The operation, *AND*, for instance, takes two binary inputs  $X$  and  $Y$  and

returns the output 1 if and only if both  $X$  and  $Y$  are 1; otherwise it returns the output 0. Similarly, *NOT* takes a single binary input  $X$  and returns the output 1 if  $X = 0$  and 0 if  $X = 1$ . *FANOUT* takes a single binary input  $X$  and returns two binary outputs, each equal to  $X$ . Any boolean function can be constructed by repeated application of *AND*, *NOT* and *FANOUT*. A boolean function, named after George Boole (see chapter 1), is just a transformation of bits. A set of operations that allows the construction of arbitrary boolean functions is called universal. The actual physical device that performs a logical operation is called a logic gate.

How fast can a digital computer perform a logical operation? During such an operation, the bits in the computer on which the operation is performed go from one state to another.

*The equations coming up just say that quantum mechanics limits how fast a logic operation can be performed. The Heisenberg uncertainty principle says that to perform a logic operation faster, energy has to be more uncertain. The Margolus-Levitin theorem says that perform a logic operation faster, you need more energy.*

The problem of how much energy is required for information processing was first investigated in the context of communications theory by Levitin<sup>11–16</sup>, Bremermann<sup>17–19</sup>, Beckenstein<sup>20–22</sup> and others, who showed that the laws of quantum mechanics determine the maximum rate at which a system with spread in energy  $\Delta E$  can move from one distinguishable state to another. In particular, the correct interpretation of the time-energy Heisenberg uncertainty principle  $\Delta E \Delta t \geq \hbar$  is not that it takes time  $\Delta t$  to measure energy to an accuracy  $\Delta E$  (a fallacy that was put to rest by Aharonov and Bohm<sup>23–24</sup>) but rather that that a quantum state with spread in energy  $\Delta E$  takes time at least  $\Delta t = \pi \hbar / 2 \Delta E$  to evolve to an orthogonal (and hence distinguishable) state<sup>23–26</sup>. More recently, Margolus and Levitin<sup>15–16</sup> extended this result to show that a quantum system with average energy  $E$  takes time at least  $\Delta t = \pi \hbar / 2 E$  to evolve to an orthogonal state.

*On to quantum mechanics and computation! The next paragraph introduces bracket notation for qubits, and calculates the average energy needed to flip a qubit. You've seen most of this before in the previous chapter.*

## 1.2 Performing quantum logic operations

As an example, consider the operation *NOT* performed on a quantum bit or ‘qubit’ with logical states  $|0\rangle$  and  $|1\rangle$ . (For readers unfamiliar with quantum mechanics, the ‘bracket’ notation  $| \rangle$  signifies that whatever is contained in the bracket is a quantum-mechanical variable;  $|0\rangle$  and  $|1\rangle$  are vectors in a two-dimensional vector space over the complex numbers.) To flip the qubit, one can apply a potential  $H = E_0|E_0\rangle\langle E_0| + E_1|E_1\rangle\langle E_1|$  with energy eigenstates  $|E_0\rangle = (1/\sqrt{2})(|0\rangle + |1\rangle)$  and  $|E_1\rangle = (1/\sqrt{2})(|0\rangle - |1\rangle)$ . Since  $|0\rangle = (1/\sqrt{2})(|E_0\rangle + |E_1\rangle)$  and  $|1\rangle = (1/\sqrt{2})(|E_0\rangle - |E_1\rangle)$ , each logical state  $|0\rangle$ ,  $|1\rangle$  has spread in energy  $\Delta E = (E_1 - E_0)/2$ . It is easy to verify that after a length of time  $\Delta t = \pi\hbar/2\Delta E$  the qubit evolves so that  $|0\rangle \rightarrow |1\rangle$  and  $|1\rangle \rightarrow |0\rangle$ . That is, applying the potential effects a *NOT* operation in a time that attains the limit given by quantum mechanics. Note that the average energy  $E$  of the qubit in course of the logical operation is  $\langle 0|H|0\rangle = \langle 1|H|1\rangle = (E_0 + E_1)/2 = E_0 + \Delta E$ . Taking the ground-state energy  $E_0 = 0$  gives  $E = \Delta E$ . So the amount of time it takes to perform a *NOT* can also be written  $\Delta t = \pi\hbar/2E$ . It is straightforward to show<sup>15–16</sup> that no quantum system with average energy  $E$  can move to an orthogonal state in a time less than  $\Delta t$ . That is, the speed with which a logical operation can be performed is limited not only by the spread in energy, but by the average energy. This result will prove to be a key component in deriving the speed limit for the ultimate laptop.

*Now we look at how to perform AND and COPY operations in terms of Toffoli or Controlled-Controlled-NOT gates. If you read the last chapter, you already know all about this.*

*AND* and *FANOUT* can be enacted in a way that is analogous to the *NOT* operation. A simple way to perform these operations in a quantum-mechanical context is to enact a so-called Toffoli or Controlled-Controlled-*NOT* operation<sup>31</sup>. This operation takes three binary inputs,  $X$ ,  $Y$ , and  $Z$  and returns three outputs,  $X'$ ,  $Y'$ , and  $Z'$ . The first two inputs pass through unchanged:  $X' = X$ ,  $Y' = Y$ . The third input passes through unchanged unless both  $X$  and  $Y$  are 1, in which case it is flipped. This is universal in the sense that suitable choices of inputs allows the construction of *AND*, *NOT*, and *FANOUT*. When the third input is set to zero,  $Z = 0$ , then the third output is the *AND* of the first two:  $Z' = X \text{ AND } Y$ . So *AND* can be constructed. When the first two inputs are 1,  $X = Y = 1$ , the third output is the *NOT* of the third input,  $Z' = \text{NOT } Z$ . Finally, when the second input is set to 1,  $Y = 1$ , and the third to zero,  $Z = 0$ , the first and third output are the *FANOUT* of the first input,  $X' = X$ ,  $Z' = X$ . So arbitrary boolean functions can

be constructed from the Toffoli operation alone.

*Now we see how fast you can perform AND and COPY logic gates. The answer is the same as for a NOT gate: the more energy you have, the faster you can do it. Then there's a discussion of where the energy you need to flip bits might come from in an actual quantum computer.*

By embedding a Controlled-Controlled-*NOT* gate in a quantum context, it is straightforward to see that *AND* and *FANOUT*, like *NOT*, can be performed at a rate  $2E/\pi\hbar$  times per second, where  $E$  is the average energy of the logic gate that performs the operation. More complicated logic operations that cycle through a larger number of quantum states (such as those on non-binary or continuous quantum variables) can be performed at a rate  $E/\pi\hbar$  — half as fast as the simpler operations<sup>15–16</sup>. Existing quantum logic gates in optical-atomic and NMR quantum computers actually attain this limit. In the case of *NOT*,  $E$  is the average energy of interaction of the qubit's dipole moment (electric dipole for optic-atomic qubits and nuclear magnetic dipole for NMR qubits) with the applied electromagnetic field. In the case of multi-qubit operations such as the Toffoli or simpler two bit Controlled-*NOT* operation, which flips the second bit if and only if the first bit is 1,  $E$  is the average energy in the interaction between the physical systems that register the qubits.

*Here it is: the ultimate speed limit for computation. You can't compute faster than this.*

### 1.3 Ultimate limits to speed of computation

We are now in a position to derive our first physical limit to computation: energy limits speed. Suppose that one has a certain amount of energy  $E$  to allocate to the logic gates of a computer. The more energy one allocates to a gate, the faster it can perform a logic operation. The total number of logic operations performed per second is equal to the sum over all logic gates of the operations per second per gate. That is, a computer can perform no more than

$$\sum_{\ell} 1/\Delta t_{\ell} \leq \sum_{\ell} 2E_{\ell}/\pi\hbar = 2E/\pi\hbar \quad (1)$$

operations per second. In other words, the rate at which a computer can compute is limited by its energy. (Similar limits have been proposed by Bremmerman in the context of the

minimum energy required to communicate a bit<sup>17–19</sup>. These limits have been criticized, however, for misinterpreting the energy-time uncertainty relation<sup>21</sup>, and for failing to take into account the role of degeneracy of energy eigenvalues<sup>13–14</sup> and the role of nonlinearity in communications<sup>7–9</sup>. *It's important to clarify how the limits presented here differ from similar ideas proposed elsewhere.) The result of the next sentence is very satisfying to present to an audience: it's really fun to write down  $E = mc^2$  on a blackboard!* Applying this result to a one kilogram computer with energy  $E = mc^2 = 8.9874 \times 10^{16}$  joules show that our ultimate laptop can perform a maximum of  $5.4258 \times 10^{50}$  operations per second.

*Now we look at what happens if we try to speed up part of the computation at the expense of another part. There is no net gain in computational speed: one part speeds up, but the other parts have to slow down.*

#### 1.4 Parallel and serial operation

An interesting feature of this limit is that it is independent of computer architecture. One might have thought that a computer could be sped up by parallelization, i.e., by taking the energy and dividing it up amongst a large number of subsystems computing in parallel. This is not the case: if one spreads the energy  $E$  amongst  $N$  logic gates, each one operates at a rate  $2E/\pi\hbar N$ . The total number of operations per second,  $N2E/\pi\hbar N = 2E/\pi\hbar$ , remains the same. If the energy is allocated to fewer logic gates (more serial operation), the rate  $1/\Delta t_\ell$  at which they operate and the spread in energy per gate  $\Delta E_\ell$  go up. If the energy is allocated to more logic gates (more parallel operation) then the rate at which they operate and the spread in energy per gate go down. Note that in this parallel case, the overall spread in energy of the computer as a whole is considerably smaller than the average energy: in general  $\Delta E = \sqrt{\sum_\ell \Delta E_\ell^2} \approx \sqrt{N}\Delta E_\ell$  while  $E = \sum E_\ell \approx NE_\ell$ . Parallelization can help perform certain computations more efficiently, but it does not alter the total number of operations per second. As will be seen below, the degree of parallelizability of the computation to be performed determines the most efficient distribution of energy among the parts of the computer. Computers in which energy is relatively evenly distributed over a larger volume are better suited for performing parallel computations. More compact computers and computers with an uneven distribution of energy are better for performing serial computations.

#### 1.5 Comparison with existing computers

*Not surprisingly, the ultimate laptop has spanking performance when compared to a PC or Macintosh.*

Conventional laptops operate much more slowly than the ultimate laptop. There are two reasons for this inefficiency. First, most of the energy is locked up in the mass of the particles of which the computer is constructed, leaving only an infinitesimal fraction for performing logic. Second, a conventional computer employs many degrees of freedom (billions and billions of electrons) for registering a single bit. From the physical perspective, such a computer operates in a highly redundant fashion. There are good technological reasons for such redundancy: conventional designs rely on redundancy for reliability and manufacturability. In the present discussion, however, the subject is not what computers are but what they might be. The laws of physics do not require redundancy to perform logical operations: recently constructed quantum microcomputers use one quantum degree of freedom for each bit and operate at the Heisenberg limit  $\Delta t = \pi\hbar/2\Delta E$  for the time needed to flip a bit<sup>64–65,76–80</sup>. Redundancy is required for error correction, however, as will be discussed below.

In sum, quantum mechanics provides a simple answer to the question of how fast information can be processed using a given amount of energy. Now it will be shown that thermodynamics and statistical mechanics provide a fundamental limit to how many bits of information can be processed using a given amount of energy confined to a given volume. Available energy necessarily limits the rate at which computer can process information. Similarly, the maximum entropy of a physical system determines the amount of information it can process. Energy limits speed. Entropy limits memory.

## **2. Entropy limits memory space**

*You've seen everything in this next paragraph before, in chapters 1 and 2.*

The amount of information that a physical system can store and process is related to the number of distinct physical states accessible to the system. A collection of  $m$  two-state systems has  $2^m$  accessible states and can register  $m$  bits of information. In general, a system with  $N$  accessible states can register  $\log_2 N$  bits of information. But it has been known for more than a century that the number of accessible states of a physical system,  $W$ , is related to its thermodynamic entropy by the formula:  $S = k_B \ln W$ , where  $k_B$  is Boltzmann's constant. (Although this formula is inscribed on Boltzmann's tomb, it is originally due to Planck: before the turn of the century,  $k_B$  was often known as Planck's

constant.)

*The temperature of a physical system is essentially the amount of energy per bit. Since energy limits how fast you can perform logical operations, the temperature of a system limits how fast each bit can flip on average. The equations just give mathematical expression to this fact.*

The amount of information that can be registered by a physical system is  $I = S(E)/k_B \ln 2$ , where  $S(E)$  is the thermodynamic entropy of a system with expectation value for the energy  $E$ . Combining this formula with the formula  $2E/\pi\hbar$  for the number of logical operations that can be performed per second, we see that when it is using all its memory, the number of operations per bit per second that our ultimate laptop can perform is  $k_B 2 \ln 2 E / \pi \hbar S \propto k_B T / \hbar$ , where  $T = (\partial S / \partial E)^{-1}$  is the temperature of a kilogram of matter in a maximum entropy in a liter volume. The entropy governs the amount of information the system can register and the temperature governs the number of operations per bit per second it can perform.

*This paragraph raises a technical point that confused several readers. Basically, the point is that though the ultimate laptop may look like it's in a state of high entropy, it's actually in a state of low entropy. In other words, it may look hot, but it's not.*

Since thermodynamic entropy effectively counts the number of bits available to a physical system, the following derivation of the memory space available to the ultimate laptop is based on a thermodynamic treatment of a kilogram of matter confined to a liter volume, in a maximum entropy state. Throughout this derivation, it is important to keep in mind that although the memory space available to the computer is given by the entropy of its thermal equilibrium state, the *actual* state of the ultimate laptop as it performs a computation is completely determined, so that its entropy remains always equal to zero. As above, we assume that we have complete control over the actual state of the ultimate laptop, and are able to guide it through its logical steps while insulating it from all uncontrolled degrees of freedom. As the following discussion will make clear, such complete control will be difficult to attain.

*The next section and the one following the box insert deal with the technical details of how to calculate entropy. Please feel free to bleep over them. You might be interested in the material in the box, however, which gives a brief history of the physics of computation.*

## 2.1 Entropy, energy, and temperature

In order to calculate the number of operations per second that could be performed by our ultimate laptop, we assumed that the expectation value of the energy was  $E$ . Accordingly, the total number of bits of memory space available to the computer is  $S(E, V)/k_B \ln 2$ , where  $S(E, V)$  is the thermodynamic entropy of a system with expectation value of the energy  $E$  confined to volume  $V$ . The entropy of a closed system is normally given by the so-called microcanonical ensemble, which fixes both the average energy *and* the spread in energy  $\Delta E$ , and assigns equal probability to all states of the system within a range  $[E, E + \Delta E]$ . In the case of the ultimate laptop, however, we wish to fix only the average energy, while letting the spread in energy vary according to whether the computer is to be more serial (fewer, faster gates, larger spread in energy) or parallel (more, slower gates, smaller spread in energy). Accordingly, the ensemble that should be used to calculate the thermodynamic entropy and the memory space available is the canonical ensemble, which maximizes  $S$  for fixed average energy with no constraint on the spread in energy  $\Delta E$ . The canonical ensemble tells how many bits of memory are available for all possible ways of programming the computer while keeping its average energy equal to  $E$ . In any given computation with average energy  $E$  the ultimate laptop will be in a pure state with some fixed spread of energy, and will explore only a small fraction of its memory space.

In the canonical ensemble a state with energy  $E_i$  has probability  $p_i = (1/Z(T))e^{-E_i/k_B T}$  where  $Z(T) = \sum_i e^{-E_i/k_B T}$  is the partition function, and the temperature  $T$  is chosen so that  $E = \sum_i p_i E_i$ . The entropy is  $S = -k_B \sum_i p_i \ln p_i = E/T + k_B \ln Z$ . The number of bits of memory space available to the computer is  $S/k_B \ln 2$ . The difference between the entropy as calculated using the canonical ensemble and that calculated using the microcanonical ensemble is minimal. There is some subtlety involved in using the canonical ensemble rather than the more traditional microcanonical ensemble, however. The canonical ensemble is normally used for open systems that interact with a thermal bath at temperature  $T$ . In the case of the ultimate laptop, however, it is applied to a *closed* system to find the maximum entropy given a fixed expectation value for the energy. As a result, the temperature  $T = (\partial S/\partial E)^{-1}$  plays a somewhat different role in the context of physical limits of computation than it does in the case of an ordinary thermodynamic system interacting with a thermal bath. Integrating the relationship  $T = (\partial S/\partial E)^{-1}$  over  $E$  yields  $T = CE/S$ , where  $C$  is a constant of order unity (e.g.,  $C = 4/3$  for black-body

radiation,  $C = 3/2$  for an ideal gas, and  $C = 1/2$  for a black hole). Accordingly, the temperature governs the number of operations per bit per second,  $k_B \ln 2E/\hbar S \approx k_B T/\hbar$ , that a system can perform. As will become clear, the relationship between temperature and operations per bit per second is useful in investigating computation under extreme physical conditions.

*Now for a bit of history.*

**(Box 1: The role of thermodynamics in computation.** The fact that entropy and information are intimately linked has been known since Maxwell introduced his famous ‘demon’ well over a century ago<sup>1</sup>. Maxwell’s demon is an hypothetical being that uses its information-processing ability to reduce the entropy of a gas. The first results in physics of information processing were derived in attempts to understand how Maxwell’s demon could function<sup>1–4</sup>. The role of thermodynamics in computation has been repeatedly examined over the last half century. In the 1950’s, von Neumann<sup>10</sup> speculated that each logical operation performed in a computer at temperature  $T$  must dissipate energy  $k_B T \ln 2$ , thereby increasing entropy by  $k_B \ln 2$ . This speculation proved to be false. The precise, correct statement of the role of entropy in computation was due to Landauer<sup>5</sup>, who showed that reversible, i.e. one-to-one, logical operations such as *NOT* can be performed without dissipation in principle, but that irreversible, many-to-one operations such as *AND* or *ERASE* require dissipation at least  $k_B \ln 2$  for each bit of information lost. (*ERASE* is a one-bit logical operation that takes a bit, 0 or 1, and restores it to 0.) The argument behind Landauer’s principle can be readily understood<sup>37</sup>. Essentially, the one-to-one dynamics of Hamiltonian systems implies that when a bit is erased the information that it contains has to go somewhere. If the information goes into observable degrees of freedom of the computer, such as another bit, then it has not been erased but merely moved; but if it goes into unobservable degrees of freedom such as the microscopic motion of molecules it results in an increase of entropy of at least  $k_B \ln 2$ .

In 1973, Bennett<sup>28–30</sup> showed that all computations could be performed using reversible logical operations only. Consequently, by Landauer’s principle, computation does not require dissipation. (Earlier work by Lecerf<sup>27</sup> had anticipated the possibility of reversible computation, but not its physical implications. Reversible computation was discovered independently by Fredkin and Toffoli<sup>31</sup>.) The energy used to perform a logical operation can be ‘borrowed’ from a store of free energy such as a battery, ‘invested’ in

the logic gate that performs the operation, and returned to storage after the operation has been performed, with a net ‘profit’ in the form of processed information. Electronic circuits based on reversible logic have been built and exhibit considerable reductions in dissipation over conventional reversible circuits<sup>33–35</sup>.

*The next two paragraphs show quantitatively how hard it would be to tolerate errors in the ultimate laptop. For even a very low error rate, the laptop would have to use a lot of power. Even if the computation itself takes place at low temperature, the energy radiated out to reject errors to the environment would make the ultimate laptop way too hot to hold on your lap!*

Under many circumstances it may prove useful to perform irreversible operations such as erasure. If our computer is subject to an error rate of  $\epsilon$  bits per second, for example, then error-correcting codes can be used to detect those errors and reject them to the environment at a dissipative cost of  $\epsilon k_B T_E \ln 2$  joules per second, where  $T_E$  is the temperature of the environment. (  $k_B T \ln 2$  is the minimal amount of energy required to send a bit down an information channel with noise temperature  $T$ .<sup>14</sup>) Such error-correcting routines in our ultimate computer function as working analogs of Maxwell’s demon, getting information and using it to reduce entropy at an exchange rate of  $k_B T \ln 2$  joules per bit. In principle, computation does not require dissipation. In practice, however, any computer – even our ultimate laptop – will dissipate energy.

The ultimate laptop must reject errors to the environment at a high rate to maintain reliable operation. To estimate the rate at which it can reject errors to the environment, assume that the computer encodes erroneous bits in the form of black-body radiation at the characteristic temperature  $5.87 \times 10^8$  K of the computer’s memory.<sup>21</sup> The Stefan-Boltzmann law for black-body radiation then implies that the number of bits per unit area than can be sent out to the environment is  $\mathcal{B} = \pi^2 k_B^3 T^3 / 60 \ln 2 \hbar^3 c^2 = 7.195 \times 10^{42}$  bits per meter<sup>2</sup> per second. Since the ultimate laptop has a surface area of  $10^{-2}$  square meters and is performing  $\approx 10^{50}$  operations per second, it must have an error rate of less than  $10^{-10}$  per operation in order to avoid over-heating. Even if it achieves such an error rate, it must have an energy throughput (free energy in and thermal energy out) of  $4.04 \times 10^{26}$  watts — turning over its own rest mass energy of  $mc^2 \approx 10^{17}$  joules in a nanosecond! The thermal load of correcting large numbers of errors clearly suggests the necessity of operating at a slower speed than the maximum allowed by the laws of physics. )

*This next section is more technical stuff on how to calculate entropy. The bottom line is that the ultimate laptop can store ten thousand billion billion billion ( $10^{31}$ ) bits. Feel free to bleep to section (2.3)*

## 2.2 Calculating the maximum memory space

To calculate exactly the maximum entropy for a kilogram of matter in a liter volume would require complete knowledge of the dynamics of elementary particles, quantum gravity, etc. We do not possess such knowledge. However, the maximum entropy can readily be estimated by a method reminiscent of that used to calculate thermodynamic quantities in the early universe<sup>87</sup>. The idea is simple: model the volume occupied by the computer as a collection of modes of elementary particles with total average energy  $E$ . The maximum entropy is obtained by calculating the canonical ensemble over the modes. Here, we supply a simple derivation of the maximum memory space available to the ultimate laptop. A more detailed discussion of how to calculate the maximum amount of information that can be stored in a physical system can be found in the work of Bekenstein<sup>19–21</sup>.

For this calculation, assume that the only conserved quantities other than the computer's energy are angular momentum and electric charge, which we take to be zero. (One might also ask that baryon number be conserved, but as will be seen below, one of the processes that could take place within the computer is black hole formation and evaporation, which does not conserve baryon number.) At a particular temperature  $T$ , the entropy is dominated by the contributions from particles with mass less than  $k_B T/2c^2$ . The  $\ell$ 'th such species of particle contributes energy  $E = r_\ell \pi^2 V (k_B T)^4 / 30 \hbar^3 c^3$  and entropy  $S = 2r_\ell k_B \pi^2 V (k_B T)^3 / 45 \hbar^3 c^3 = 4E/3T$  where  $r_\ell$  is equal to the number of particles/antiparticles in the species (i.e., 1 for photons, 2 for electrons/positrons) times the number of polarizations (2 for photons, 2 for electrons/positrons) times a factor that reflects particle statistics (1 for bosons, 7/8 for fermions). As the formula for  $S$  in terms of  $T$  shows, each species contributes  $(2\pi)^5 r_\ell / 90 \ln 2 \approx 10^2$  bits of memory space per cubic thermal wavelength  $\lambda_T^3$  where  $\lambda_T = 2\pi \hbar c / k_B T$ . Re-expressing the formula for entropy as a function of energy, our estimate for the maximum entropy is

$$S = (4/3)k_B(\pi^2 r V / 30 \hbar^3 c^3)^{1/4} E^{3/4} = k_B \ln 2I, \quad (2)$$

where  $r = \sum_\ell r_\ell$ . Note that  $S$  depends only insensitively on the total number of species with mass less than  $k_B T/2c^2$ .

A lower bound on the entropy can be obtained by assuming that energy and entropy are dominated by black body radiation consisting of photons. In this case,  $r = 2$ , and for a one kilogram computer confined to a volume of a liter we have  $k_B T = 8.10 \times 10^{-15}$  joules, or  $T = 5.87 \times 10^8$  K. The entropy is  $S = 2.04 \times 10^8$  joule/K, which corresponds to an amount of available memory space  $I = S/k_B \ln 2 = 2.13 \times 10^{31}$  bits. When the ultimate laptop is using all its memory space it can perform  $2 \ln 2 k_B E / \pi \hbar S = 3 \ln 2 k_B T / 2 \pi \hbar \approx 10^{19}$  operations per bit per second. As the number of operations per second  $2E/\pi\hbar$  is independent of the number of bits available, the number of operations per bit per second can be increased by using a smaller number of bits. In keeping with the prescription that the ultimate laptop operates at the absolute limits given by physics, in what follows, I assume that all available bits are used.

This estimate for the maximum entropy could be improved (and slightly increased) by adding more species of massless particles (neutrinos and gravitons) and by taking into effect the presence of electrons and positrons. Note that  $k_B T / 2c^2 = 4.51 \times 10^{-32}$  kilograms, compared with the electron mass of  $9.1 \times 10^{-31}$  kilograms. That is, our kilogram computer in a liter is close to a phase transition at which electrons and positrons are produced thermally. A more exact estimate of the maximum entropy and hence the available memory space would be straightforward to perform, but the details of such a calculation would detract from our general exposition, and could only serve to alter  $S$  slightly.  $S$  depends insensitively on the number of species of effectively massless particles: a change of  $r$  by a factor of 10,000 serves only to increase  $S$  by a factor of 10.

### 2.3 Comparison with current computers

*Perhaps not surprisingly, the memory capacity of the ultimate laptop compares favorably with currently available laptops.*

The amount of information that can be stored by the ultimate laptop,  $\approx 10^{31}$  bits, is much higher than the  $\approx 10^{10}$  bits stored on current laptops. This is because conventional laptops use many degrees of freedom to store a bit where the ultimate laptop uses just one. There are considerable advantages to using many degrees of freedom to store information, stability and controllability being perhaps the most important. Indeed, as the above calculation indicates, in order to take full advantage of the memory space available, the ultimate laptop must turn all its matter into energy. A typical state of the ultimate laptop's memory looks like a plasma at a billion degrees Kelvin: the laptop's memory looks like

a thermonuclear explosion or a little piece of the Big Bang! Clearly, packaging issues alone make it unlikely that this limit can be obtained, even setting aside the difficulties of stability and control.

Even though the ultimate physical limit to how much information can be stored in a kilogram of matter in a liter volume is unlikely to be attained, it may nonetheless be possible to get a fair way along the road to such bit densities. In other words, the ultimate limits to memory space may prove easier to approach than the ultimate limits to speed. Following Moore's law, the density of bits in a computer has gone down from approximately one per  $\text{cm}^2$  fifty years ago to one per  $\mu\text{m}^2$  today, an improvement of a factor of  $10^8$ . It is not inconceivable that a similar improvement is possible over the course of the next fifty years. In particular, there is no physical reason why it should not be possible to store one bit of information per atom. Indeed, existing NMR and ion-trap quantum computers already store information on individual nuclei and atoms (typically in the states of individual nuclear spins or in hyperfine atomic states). Solid-state NMR with high gradient fields or quantum optical techniques such as spectral hole-burning provide potential technologies for storing large quantities of information at the atomic scale. A kilogram of ordinary matter holds on the order of  $10^{25}$  nuclei. If a substantial fraction of these nuclei can be made to register a bit, then one can get quite close to the ultimate physical limit of memory without having to resort to thermonuclear explosions. If, in addition, one uses the natural electromagnetic interactions between nuclei and electrons in the matter to perform logical operations, one is limited to a rate of approximately  $10^{15}$  operations per bit per second, yielding an overall information processing rate of  $\approx 10^{40}$  operations per second in ordinary matter. Although less than the  $\approx 10^{51}$  operations per second in the ultimate laptop, the maximum information processing rate in 'ordinary matter' is still quite respectable. Of course, even though such an 'ordinary matter' ultimate computer need not operate at nuclear energy levels, other problems remain: for example, the high number of bits still suggests substantial input/output problems. At an input/output rate of  $10^{12}$  bits per second, an Avogadro-scale computer with  $10^{23}$  bits would take about 10,000 years to perform a serial read/write operation on the entire memory. Higher throughput and parallel input/output schemes are clearly required to take advantage of the entire memory space that physics makes available.

*Now let's look at what happens if we try to continue Moore's law further and cram even more components into a smaller and smaller space.*

### 3. Size limits parallelization

Up until this point, we have assumed that our computer occupies a volume of a liter. The previous discussion, however, indicates that benefits are to be obtained by varying the volume to which the computer is confined. Generally speaking, if the computation to be performed is highly parallelizable or requires many bits of memory, the volume of the computer should be greater and the energy available to perform the computation should be spread out evenly amongst the different parts of the computer. Conversely, if the computation to be performed is highly serial and requires fewer bits of memory, the energy should be concentrated in particular parts of the computer.

*This part compares the amount of time it takes to flip a bit in the ultimate laptop with the amount of time it takes to send a signal from one side of the computer to the other. The longer it takes to send such a signal, the more ‘parallel’ the operation of the computer has to be: that is, the more the different parts of the computer have to work on their own. The time it takes to send a signal across the computer can be made smaller by compressing the computer to make it smaller. But if you compress it too far, bad things begin to happen: the computer becomes a black hole!*

A good measure of the degree of parallelization in a computer is the ratio between time it takes to communicate from one side of the computer to the other, and the average time it takes to perform a logical operation. The amount of time it takes to send a message from one side of a computer of radius  $R$  to the other is  $t_{\text{com}} = 2R/c$ . The average time it takes a bit to flip in the ultimate laptop is the inverse of the number of operations per bit per second calculated above:  $t_{\text{flip}} = \pi\hbar S/k_B 2 \ln 2E$ . Our measure of the degree of parallelization in the ultimate laptop is then

$$t_{\text{com}}/t_{\text{flip}} = k_B 4 \ln 2RE/\pi\hbar cS \propto k_B RT/\hbar c = 2\pi R/\lambda_T. \quad (3)$$

That is, the amount of time it takes to communicate from one side of the computer to the other, divided by the amount of time it takes to flip a bit, is approximately equal to the ratio between the size of the system and its thermal wavelength. For the ultimate laptop, with  $2R = 10^{-1}$  meters,  $2E/\pi\hbar \approx 10^{51}$  operations per second, and  $S/k_B \ln 2 \approx 10^{31}$  bits,  $t_{\text{com}}/t_{\text{flip}} \approx 10^{10}$ . The ultimate laptop is highly parallel. A greater degree of serial computation can be obtained at the cost of decreasing memory space by compressing the size of the computer or making the distribution of energy more uneven. As ordinary matter

obeys the Beckenstein bound<sup>20–22</sup>,  $k_B RE/\hbar cS > 1/2\pi$ , however, as one compresses the computer  $t_{\text{com}}/t_{\text{flip}} \approx k_B RE/\hbar cS$  will remain greater than one: i.e., the operation will still be somewhat parallel. Only at the ultimate limit of compression — a black hole — is the computation entirely serial.

*OK, let's see just what happens when our computer becomes a black hole. In depressing fact, many non-ultimate computers can suddenly start behaving like a black hole: no matter what you type in, they don't respond.*

### 3.1 Compressing the computer allows more serial computation

Suppose that one wants to perform a highly serial computation on few bits. Then it is advantageous to compress the size of the computer so that it takes less time to send signals from one side of the computer to the other at the speed of light. As the computer gets smaller, keeping the energy fixed, the energy density inside the computer goes up. As the energy density in the computer goes up, different regimes in high energy physics are necessarily explored in the course of the computation. First the weak unification scale is reached, then the grand unification scale. Finally, as the linear size of the computer approaches its Schwarzschild radius, the Planck scale is reached. (No known technology could possibly achieve such compression.)

*The next bit of numbers describe the Planck scale. The main thing to note is that the Planck scale is mighty small!*

At the Planck scale, gravitational effects and quantum effects are both important: the Compton wavelength of a particle of mass  $m$ ,  $\lambda_C = 2\pi\hbar/mc$  is on the order of its Schwarzschild radius,  $2Gm/c^2$ . In other words, to describe behavior at length scales of the size  $\ell_P = \sqrt{\hbar G/c^3} = 1.616 \times 10^{-35}$  meter, time scales  $t_P = \sqrt{\hbar G/c^5} = 5.391 \times 10^{-44}$  second, and mass scales of  $m_P = \sqrt{\hbar c/G} = 2.177 \times 10^{-8}$  kilograms, a unified theory of quantum gravity is required. We do not currently possess such a theory. Nonetheless, although we do not know the exact number of bits that can be registered by a one kilogram computer confined to a volume of a liter, we do know the exact number of bits that can be registered by a one kilogram computer that has been compressed to the size of a black hole<sup>90</sup>. This is because the entropy of a black hole has a well-defined value.

In the following discussion, we use the properties of black holes to place limits on the speed, memory space, and degree of serial computation that could be approached by

compressing a computer to the smallest possible size. Whether or not these limits could be attained, even in principle, is a question whose answer will have to await a unified theory of quantum gravity. (See Box 2)

*Even if we don't have a unified theory of quantum gravity, we can still say quite a lot about just how much information a black hole can store and process. The next couple of paragraphs put quantitative bounds on how much computation a one kilogram black hole could perform. Please bleep to the next section if you don't want to know the answer.*

The Schwarzschild radius of a 1 kilogram computer is  $R_S = 2Gm/c^2 = 1.485 \times 10^{-27}$  meters. The entropy of a black hole is Boltzmann's constant times its area divided by 4, as measured in Planck units. Accordingly, the amount of information that can be stored in a black hole is  $I = 4\pi Gm^2/\ln 2\hbar c = 4\pi m^2/\ln 2m_P^2$ . The amount of information that can be stored by the 1 kilogram computer in the black-hole limit is  $3.827 \times 10^{16}$  bits. A computer compressed to the size of a black hole can perform  $5.4258 \times 10^{50}$  operations per second, the same as the 1 liter computer.

In a computer that has been compressed to its Schwarzschild radius, the energy per bit is  $E/I = mc^2/I = \ln 2\hbar c^3/4\pi mG = \ln 2k_B T/2$ , where  $T = (\partial S/\partial E)^{-1} = \hbar c/4\pi k_B R_S$  is the temperature of the Hawking radiation emitted by the hole. As a result, the time it takes to flip a bit on average is  $t_{\text{flip}} = \pi\hbar I/2E = \pi^2 R_S/c \ln 2$ . In other words, according to a distant observer, the amount of time it takes to flip a bit,  $t_{\text{flip}}$ , is on the same order as the amount of time  $t_{\text{com}} = \pi R_S/c$  it takes to communicate from one side of the hole to the other by going around the horizon:  $t_{\text{com}}/t_{\text{flip}} = \ln 2/\pi$ . In contrast to computation at lesser densities, which is highly parallel as noted above, computation at the horizon of a black hole is highly serial: every bit is essentially connected to every other bit over the course of a single logic operation. As noted above, the serial nature of computation at the black-hole limit can be deduced from the fact that black holes attain the Beckenstein bound<sup>20-22</sup>,  $k_B RE/\hbar cS = 1/2\pi$ .

*Just what happens in black holes is something of a mystery. Truly to understand black holes would require a theory of quantum gravity, which we don't have (see, however, later on in this chapter for a theory of quantum gravity expressed in terms of quantum computation). The classical picture of a black hole is that what goes in, does not come out. But there is another, quantum-mechanical picture that suggests what goes into a black hole does come out, but in profoundly altered form. Which of these pictures is right is the subject of*

a well-known bet between the quantum cosmologist Stephen Hawking and the black-hole and quantum computing expert John Preskill. Hawking says the information doesn't come out; Preskill says it does. If Preskill is right, then a black hole certainly transforms and processes information. But whether this information processing constitutes a computation is anybody's guess. For other ways that black holes can be used to aid computation, see footnote 99.

**(Box 2: Can a black hole compute?)**

No information can escape from a classical black hole: what goes in does not come out. The quantum mechanical picture of a black hole is different, however. First of all, black holes are not quite black: they radiate at the Hawking temperature. In addition, the well-known statement that ‘a black hole has no hair’—i.e., from a distance all black holes with the same charge and angular momentum look essentially alike — is now known to be not always true<sup>91–93</sup>. Finally, recent work in string theory<sup>94–96</sup> suggests that black holes do not actually destroy the information about how they were formed, but instead process it and emit the processed information as part of the Hawking radiation as they evaporate: what goes in does come out, but in an altered form.

If the latter picture is correct, then black holes could in principle be ‘programmed’: one forms a black hole whose initial conditions encode the information to be processed, lets that information be processed by the Planckian dynamics at the hole’s horizon, and gets out the answer to the computation by examining the correlations in the Hawking radiation emitted when the hole evaporates. Despite our lack of knowledge of the precise details of what happens when a black hole forms and evaporates (a full account must await a more exact treatment using whatever theory of quantum gravity and matter turns out to be the correct one), we can still provide a rough estimate how much information is processed during this computation. *The following few sentences give numerical estimates for how many elementary logical operations a black hole could conceivably accomplish during its lifetime. Bleep if you don't want to know the answer.* Using Page’s results on the rate of evaporation of a black hole<sup>88</sup>, we obtain a lifetime for the hole  $t_{\text{life}} = G^2 m^3 / 3C\hbar c^4$ , where  $C$  is a constant that depends on the number of species of particles with a mass less than  $k_B T$ , where  $T$  is the temperature of the hole. For  $O(10^1 - 10^2)$  such species,  $C$  is on the order of  $10^{-3} - 10^{-2}$ , leading to a lifetime for a 1 kilogram black hole of  $\approx 10^{-19}$  seconds, during which time the hole can perform  $\approx 10^{32}$  operations on its  $\approx 10^{16}$  bits. As the actual number of effectively massless particles at the Hawking temperature of a

one-kilogram black hole is likely to be considerably larger than  $10^2$ , this number should be regarded as an upper bound on the actual number of operations that could be performed by the hole. Interestingly, although this hypothetical computation is performed at ultra-high densities and speeds, the total number of bits available to be processed is not far from the number available to current computers operating in more familiar surroundings. )

*Crazy as all the discussion here sounds, simple computers that operate at the limits given by the laws of physics have been constructed. Existing quantum computers, in their own modest way, attain the ultimate physical limits to computation.*

#### 4. Constructing ultimate computers

Throughout this entire discussion of the physical limits to computation, no mention has been made of how to construct a computer that operates at those limits. In fact, contemporary quantum ‘microcomputers’ such as those constructed using nuclear magnetic resonance<sup>76–80</sup> do indeed operate at the limits of speed and memory space described above. Information is stored on nuclear spins, with one spin registering one bit. *The next two sentences just confirm mathematically that nuclear spins do indeed attain the ultimate physical limits to computation as they flip.* The time it takes a bit to flip from a state  $|\uparrow\rangle$  to an orthogonal state  $|\downarrow\rangle$  is given by  $\pi\hbar/2\mu B = \pi\hbar/2E$ , where  $\mu$  is the spin’s magnetic moment,  $B$  is the magnetic field, and  $E = \mu B$  is the average energy of interaction between the spin and the magnetic field. To perform a quantum logic operation between two spins takes a time  $\pi\hbar/2E_\gamma$ , where  $E_\gamma$  is the energy of interaction between the two spins.

Although NMR quantum computers already operate at the limits to computation set by physics, they are nonetheless much slower and process much less information than the ultimate laptop described above. This is because their energy is largely locked up in mass, thereby limiting both their speed and their memory. Unlocking this energy is of course possible, as a thermonuclear explosion indicates. Controlling such an ‘unlocked’ system is another question, however. In discussing the computational power of physical systems in which all energy is put to use, we assumed that such control is possible in principle, although it is certainly not possible in current practice. All current designs for quantum computers operate at low energy levels and temperatures, exactly so that precise control can be exerted on their parts.

As the discussion of error correction above indicates, the rate at which errors can be detected and rejected to the environment by error correction routines puts a fundamental

limit on the rate at which errors can be committed.

*Next, I derive a mathematical expression for the maximum error rate that an ultimate computer can tolerate if it is going to compute at the maximum rate allowed by the laws of physics. If the computer wishes to kick back and compute at a slower speed, it can tolerate more errors.*

Suppose that each logical operation performed by the ultimate computer has a probability  $\epsilon$  of being erroneous. The total number of errors committed by the ultimate computer per second is then  $2\epsilon E/\pi\hbar$ . The maximum rate at which information can be rejected to the environment is, up to a geometric factor,  $\ln 2cS/R$  (all bits in the computer moving outward at the speed of light). Accordingly, the maximum error rate that the ultimate computer can tolerate is  $\epsilon \leq \pi \ln 2\hbar cS/2ER = 2t_{\text{flip}}/t_{\text{com}}$ . That is, the maximum error rate that can be tolerated by the ultimate computer is the inverse of its degree of parallelization.

*While I was writing this article, I received an invitation to go give the physics colloquium at Brookhaven National Laboratory, on Long Island. I had worked there one summer when I was an undergraduate, cleaning out an old rusty particle detector for Jim Christiansen of NYU. To clear away the fumes, we used to go and stick our heads into the particle beam. As mentioned in the previous chapter, the beam was highly attenuated, so only a few particles were coming through each second. The particles go so fast, that they go faster than the speed of light inside the fluid of the eye. (Of course, nothing can go faster than light in vacuum; but when light hits the fluid in the eye, it slows down significantly, while the elementary particles do not.) The result is the light analog of a sonic boom, called Cherenkov radiation: it manifests itself as an early blue flash inside the eye. The old-timers used to align the particle beam by looking for the Cherenkov radiation.*

*Because of my fond associations with Brookhaven in the past, I was eager to give the physics colloquium. Naturally, I talked about the ultimate physical limits to computation. It was fun. During the talk, a Brookhaven scientist named Sydney Kahana gave me a hard time: ‘Why should I care about these ultimate limits?’ he said, ‘What are they good for?’ After the talk, he and I sat down and applied the formulae of the limits of computation to heavy ion collisions, in which atomic nuclei are bashed together at high speeds. We calculated the number of bits that could be registered by the elementary particles formed during the collision, and the number of logical operations that the interactions between those particles could perform. Those numbers are presented in the next paragraph. In the*

*end, Kahana admitted that it was in fact illuminating to think of a heavy-ion collision as a computation.*

Suppose that control of highly energetic systems were to become possible. Then how might these systems be made to compute? As an example of a ‘computation’ that might be performed at extreme conditions, consider a heavy-ion collision that takes place in heavy-ion collider at Brookhaven<sup>97</sup>. If one collides 100 on 100 nucleons at 200 GeV per nucleon, the operation time is  $\pi\hbar/2E \approx 10^{-29}$  seconds. The maximum entropy can be estimated to be approximately to be 4 per relativistic pion (to within a factor of less than 2 associated with the overall production rate per mesons) of which there are approximately  $10^4$  per central collision in which only a few tens of nucleons are spectators. Accordingly, the total amount of memory space available is  $S/k_B \ln 2 \approx 10^4 - 10^5$  bits. The collision time is short: in the center of mass frame the two nuclei are Lorentz contracted to  $D/\gamma$  where  $D = 12 - 13$  fermi and  $\gamma = 100$ , giving a total collision time of  $\approx 10^{-25}$  seconds. During the collision, then, there is time to perform approximately  $10^4$  operations on  $10^4$  bits — a relatively simple computation. (The fact that only one operation per bit is performed suggests that there is insufficient time to reach thermal equilibrium, an observation that is confirmed by detailed simulations.) The heavy ion system could be programmed by manipulating and preparing the initial momenta and internal nuclear states of the ions. Of course, one does not expect to be able do word processing on such a ‘computer.’ Rather one expects to uncover basic knowledge about nuclear collisions and quark-gluon plasmas: in the words of Heinz Pagels, the plasma ‘computes itself.’<sup>98</sup>

At the greater extremes of a black hole computer, we assumed that whatever theory (string theory, M theory?) turns out to be the correct theory of quantum matter and gravity, it is possible to prepare initial states of such systems that causes their natural time evolution to carry out a computation. What assurance do we have that such preparations exist, even in principle?

Physical systems that can be programmed to perform arbitrary digital computations are called computationally universal. Although computational universality might at first seem to be a stringent demand on a physical system, a wide variety of physical systems — ranging from nearest neighbor Ising models<sup>52</sup> to quantum electrodynamics<sup>84</sup> and conformal field theories<sup>86</sup> — are known to be computationally universal<sup>51–53,55–65</sup>. Indeed, computational universality seems to be the rule rather than the exception. Essentially any quantum system that admits controllable nonlinear interactions can be shown to be

computationally universal<sup>60–61</sup>.

*The next paragraph shows that the time it takes to perform a logical operation using the electromagnetic interaction between two elementary particles is related to the amount of time it takes to send a signal between the two. The ratio between these two times equal to the fine structure constant, one of the fundamental constants of nature. Feel free to bleep over this piece of numerology.*

For example, the ordinary electrostatic interaction between two charged particles can be used to perform universal quantum logic operations between two quantum bits. A bit is registered by the presence or absence of a particle in a mode. The strength of the interaction between the particles,  $e^2/r$ , determines the amount of time  $t_{flip} = \pi\hbar r/2e^2$  it takes to perform a quantum logic operation such as a Controlled-*NOT* on the two particles. Interestingly, the time it takes to perform such an operation divided by the amount of time it takes to send a signal at the speed of light between the bits  $t_{com} = r/c$  is a universal constant,  $t_{flip}/t_{com} = \pi\hbar c/2e^2 = \pi/2\alpha$ , where  $\alpha = e^2/\hbar c \approx 1/137$  is the fine structure constant. This example shows the degree to which the laws of physics and the limits to computation are entwined.

*I conclude by concluding.*

In addition to the theoretical evidence that most systems are computationally universal, the computer on which I am writing this article provides strong experimental evidence that whatever the correct underlying theory of physics is, it supports universal computation. Whether or not it is possible to make computation take place in the extreme regimes envisaged in this paper is an open question. The answer to this question lies in future technological development, which is difficult to predict. If, as seems highly unlikely, it is possible to extrapolate the exponential progress of Moore's law into the future, then it will only take two hundred and fifty years to make up the forty orders of magnitude in performance between current computers that perform  $10^{10}$  operations per second on  $10^{10}$  bits and our one kilogram ultimate laptop that performs  $10^{51}$  operations per second on  $10^{31}$  bits.

*What would a scientific paper be without copious references? The references are supposed to point the reader to relevant prior scientific work. This paper has more references than*

any other scientific paper I have ever written, and I could easily have put in a hundred more references that were only slightly less relevant. The fact is, concepts of information in physics go back more than a century; and physical treatments of information processing go back almost that long. The foundations for understanding the computational nature of the universe are old, broad, and deep. Of course, there's another reason for having lots of references. You never know who the referee is going to be; but you do know that if that referee's papers are not included in the references, he or she is going to be hopping mad.

For black hole aficionados I have included a final footnote, footnote 99, which was excluded from the published paper for length considerations. Think of it as part of the 'director's cut.'

## References

1. Maxwell, J.C., *Theory of Heat*, (Appleton, London), 1871.
2. Smoluchowski, F. *Vorträge über die kinetische Theorie der Materie u. Elektrizität*, Leipzig: 1914.
3. Szilard, L., 'Über die Entropieverminderung in einem thermodynamischen System bei Eingriffen intelligenter Wesen,' *Z. Physik* **53**, 840-856 (1929).
4. Brillouin, L., 'Science and Information Theory,' (Academic Press, New York), 1953.
5. Landauer, R., 'Irreversibility and Heat Generation in the Computing Process,' *IBM J. Res. Develop.* **5**, 183-191 (1961).
6. Keyes, R.W., Landauer, R., 'Minimal Energy Dissipation in Logic,' *IBM Journal of Research and Development*, **14**, no. 2, 152-157 (1970).
7. Landauer, R., 'Dissipation and Noise-Immunity in Computation and Communication,' *Nature*, **335**, 779-784 (1988).
8. Landauer, R., 'Information is Physical,' *Physics Today*, May 1991, 23-29 (1991).
9. Landauer, R., 'The physical nature of information,' *Phys. Lett. A*, **217**, 188-193 (1996).
10. von Neumann, J., 'Theory of Self-Reproducing Automata,' Lecture 3 (University of Illinois Press, Urbana), 1966.
11. Lebedev, D.S., Levitin, L.B., 'Information Transmission by Electromagnetic Field,' *Information and Control* **9**, 1-22 (1966).
12. Levitin, L.B., in 'Proceedings of the 3rd International Symposium on Radio Electronics,' part 3, 1-15 (Varna, Bulgaria), 1970.

13. L.B. Levitin, 'Physical Limitations of Rate, Depth, and Minimum Energy in Information Processing,' *Int. J. Theor. Phys.* **21**, 299-309 (1982).
14. Levitin, L.B. 'Energy cost of information transmission (along the path to understanding),' *Physica D* **120**, 162-167 (1998).
15. Margolus, N., Levitin, L.B., 'The Maximum Speed of Dynamical Evolution,' in *PhysComp96*, T. Toffoli, M. Biafore, J. Leao, eds. (NECSI, Boston) 1996.
16. Margolus, N., Levitin, L.B., 'The Maximum Speed of Dynamical Evolution,' *Physica D* **120**, 188-195 (1998).
17. Bremermann, H.J., in 'Self-Organizing Systems,' M.C. Yovits, G.T. Jacobi, and G.D. Goldstein, eds. (Spartan Books, Washington, D.C.), 1962.
18. Bremermann, H.J., in 'Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability,' (University of California Press, Berkeley), 1967.
19. Bremermann, H.J., 'Minimum Energy Requirements of Information Transfer and Computing,' *Int. J. Theor. Phys.* **21**, 203-217 (1982).
20. Bekenstein, J.D., 'Universal Upper Bound on the Entropy-to-Energy Ratio for Bounded Systems,' *Phys. Rev. D.* **23**, 287-298 (1981).
21. Bekenstein, J.D., 'Energy Cost of Information Transfer,' *Phys. Rev. Letters* **46**, 623-626 (1981).
22. Bekenstein, J.D., 'Entropy Content and Information Flow in Systems With Limited Energy,' *Phys. Rev. D.* **30**, 1669-1679 (1984).
23. Aharonov, Y., Bohm, D., 'Time in the Quantum Theory and the Uncertainty Relation for the Time and Energy Domain,' *Phys. Rev.* **122**, 1649-1658 (1961).
24. Aharonov, Y., Bohm, D., 'Answer to Fock Concerning the Time-Energy Indeterminacy Relation,' *Phys. Rev. B.* **134**, 1417-1418 (1964).
25. Anandan, J., Aharonov, Y. 'Geometry of Quantum Evolution,' *Phys. Rev. Lett.* **65**, 1697-1700 (1990).
26. Peres, A., *Quantum Theory: Concepts and Methods*, (Kluwer, Hingham) 1995.
27. Lecerf, Y. 'Machines de Turing réversibles,' *Comptes Rendus* **257**, 2597-2600 (1963).
28. Bennett, C.H., 'Logical Reversibility of Computation,' *IBM J. Res. Develop.* **17**, 525-532 (1973).
29. Bennett, C.H., 'Thermodynamics of Computation—a Review,' *Int. J. Theor. Phys.* **21**, 905-940 (1982).
30. Bennett, C.H., 'Demons, Engines and the Second Law,' *Sci. Am.* **257**, 108 (1987).

31. Fredkin, E., Toffoli, T., ‘Conservative Logic,’ *Int. J. Theor. Phys.* **21**, 219-253 (1982).
32. Likharev K.K., ‘Classical and Quantum Limitations on Energy Consumption in Computation,’ *Int. J. Theor. Phys.*, **21**, 311-325 (1982).
33. Seitz, C.L. *et al.*, in *Proceedings of the 1985 Chapel Hill Conference on VLSI*, (Computer Science Press) 1985.
34. Merkle, R.C., ‘Reversible Electronic Logic Using Switches,’ submitted to *Nanotechnology*, 1992.
35. Younis, S.G., Knight, T.F., in *Proceedings of the 1993 Symposium on Integrated Systems*, Seattle, Washington, (MIT Press, Cambridge) 1993.
36. Lloyd, S., Pagels, H. ‘Complexity as Thermodynamic Depth,’ *Ann. Phys.* **188**, 186-213, (1988).
37. Lloyd, S., ‘Use of Mutual Information to Decrease Entropy - Implications for the Second Law of Thermodynamics,’ *Phys. Rev. A* **39**, 5378-5386, (1989).
38. Zurek, W.H., ‘Thermodynamic Cost of Computation, Algorithmic Complexity and the Information Metric,’ *Nature* **341**, 119-124 (1989).
39. Leff, H.S., Rex, A.F., *Maxwell’s Demon: Entropy, Information, Computing*, Princeton University Press, Princeton, 1990.
40. Lloyd, S., ‘Quantum Mechanical Maxwell’s Demon,’ *Phys. Rev. A* **56**, 3374-3382, (1997).
41. Benioff, P., ‘The Computer as a Physical System: A Microscopic Quantum Mechanical Hamiltonian Model of Computers as Represented by Turing Machines,’ *J. Stat. Phys.* **22**, 563-591 (1980).
42. Benioff, P., ‘Quantum Mechanical Models of Turing Machines that Dissipate No Energy,’ *Phys. Rev. Lett.* **48**, 1581-1585 (1982).
43. Feynman, R.P., ‘Simulating Physics with Computers,’ *Int. J. Theor. Phys.* **21**, 467 (1982).
44. Feynman, R.P., ‘Quantum Mechanical Computers,’ *Optics News* **11**, 11 (1985); reprinted in *Found. Phys.* **16**, 507 (1986).
45. Zurek, W.H., ‘Reversibility and Stability of Information-Processing Systems,’ *Phys. Rev. Lett.* **53**, 391-394 (1984).
46. Peres, A., ‘Reversible Logic and Quantum Computers,’ *Phys. Rev. A* **32**, 3266-3276 (1985).
47. Deutsch, D., ‘Quantum-Theory, the Church-Turing Principle, and the Universal Quan-

- tum Computer,' *Proc. Roy. Soc. Lond.* **A 400**, 97-117 (1985).
48. Margolus, N., 'Quantum Computation,' *Ann. N.Y. Acad. Sci.* **480**, 487-497 (1986).
  49. Deutsch, D., 'Quantum Computational Networks,' *Proc. Roy. Soc. Lond.* **A 425**, 73-90 (1989).
  50. Margolus, N., 'Parallel Quantum Computation,' *Complexity, Entropy, and the Physics of Information, Santa Fe Institute Studies in the Sciences of Complexity VIII*, W.H. Zurek, ed., (Addison Wesley, Redwood City, 1991) pp. 273-288.
  51. Lloyd, S., 'Quantum-Mechanical Computers and Uncomputability,' *Phys. Rev. Lett.* **71**, 943-946 (1993).
  52. Lloyd, S., 'A Potentially Realizable Quantum Computer,' *Science* **261**, 1569-1571 (1993).
  53. Lloyd, S., 'Necessary and Sufficient Conditions for Quantum Computation,' *J. Mod. Opt.* **41**, 2503-2520 (1994).
  54. Shor, P., 'Algorithms for Quantum Computation: Discrete Log and Factoring,' in *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, S. Goldwasser, Ed., IEEE Computer Society, Los Alamitos, CA, 1994, pp. 124-134.
  55. Lloyd, S., 'Quantum-Mechanical Computers,' *Sci. Am.* **273**, 140-145 (1995).
  56. Divincenzo, D., 'Quantum Computation' *Science* **270**, 255-261 (1995).
  57. DiVincenzo, D.P., '2-Bit Gates Are Universal For Quantum Computation' *Phys. Rev. A* **51**, 1015-1022 (1995).
  58. Sleator, T., Weinfurter, H., 'Realizable Universal Quantum Logic Gates,' *Phys. Rev. Lett.* **74**, 4087-4090 (1995).
  59. Barenco, A., Bennett, C.H., Cleve, R., DiVincenzo, D.P., Margolus, N., Shor, P., Sleator, T., Smolin, J.A., Weinfurter, H., 'Elementary Gates For Quantum Computation,' *Phys. Rev. A* **52**, 3457-3467 (1995).
  60. Lloyd, S., 'Almost Any Quantum Logic Gate is Universal,' *Phys. Rev. Lett.*, **75**, 346-349 (1995).
  61. Deutsch, D., Barenco, A., Ekert, A., 'Universality in Quantum Computation,' *Proc. Roy. Soc. A* **449**, 669-677 (1995).
  62. Cirac, J.I., Zoller, P. 'Quantum Computation with Cold Ion Traps,' *Phys. Rev. Lett.* **74**, 4091-4094 (1995).
  63. Pellizzari, T., Gardiner, S.A., Cirac, J.I., Zoller, P., 'Decoherence, Continuous Observation, and Quantum Computing - a Cavity QED Model,' *Phys. Rev. Lett.* **75**,

- 3788-3791 (1995).
64. Turchette, Q.A., Hood, C.J., Lange, W., Mabuchi, H., Kimble, H.J., ‘Measurement of Conditional Phase-Shifts for Quantum Logic,’ *Phys. Rev. Lett.* **75**, 4710-4713 (1995).
  65. Monroe, C., Meekhof, D.M., King, B.E., Itano, W.M., Wineland, D.J., ‘Demonstration of a Fundamental Quantum Logic Gate,’ *Phys. Rev. Lett.* **75**, 4714-4717 (1995).
  66. Grover, L.K., ‘A Fast Quantum Mechanical Algorithm for Database Search,’ in *Proceedings of the 28th Annual ACM Symposium on the Theory of Computing*, ACM Press, New York, 1996, pp. 212-218.
  67. Lloyd, S., ‘Universal Quantum Simulators,’ *Science* **273**, 1073-1078 (1996).
  68. Zalka, C., ‘Simulating Quantum Systems on a Quantum Computer,’ *Proc. R. Soc. Lond A. Mat.* **454**, 313-322 (1998).
  69. Shor, P.W., ‘A Scheme for Reducing Decoherence in Quantum Memory,’ *Phys. Rev. A* **52**, R2493-R2496 (1995).
  70. Steane, A.M., ‘Error correcting codes in quantum theory,’ *Phys. Rev. Lett.* **77**, 793-797 (1996).
  71. Laflamme, R., Miquel, C., Paz, J.P., Zurek W.H., ‘Perfect quantum error correcting code,’ *Phys. Rev. Lett.* **77**, 198-201 (1996).
  72. DiVincenzo, D.P., Shor, P.W., ‘Fault-tolerant Error Correction With Efficient Quantum Codes’ *Phys. Rev. Lett.* **77**, 3260-3263 (1996).
  73. Shor, P., ‘Fault-Tolerant Quantum Computation,’ in *Proceedings of the 37th Annual Symposium on the Foundations of Computer Science*, IEEE Computer Society Press, Los Alamitos, 1996, pp. 56-65.
  74. Preskill, J., ‘Reliable Quantum Computers,’ *Proc. Roy. Soc. Lond. Ser. A* **454**, 385-410 (1998).
  75. Knill M., Laflamme, R., Zurek, W.H., ‘Resilient quantum computation,’ *Science* **279**, 342-345 (1998).
  76. Cory, D.G., Fahmy, A.F., Havel, T.F., ‘Nuclear Magnetic Resonance Spectroscopy: an experimentally accessible paradigm for quantum computing,’ in *PhysComp96, Proceedings of the Fourth Workshop on Physics and Computation*, T. Toffoli, M. Biafore, J. Leão, eds., (New England Complex Systems Institute, Boston) 1996 87-91.
  77. Gershenfeld, N.A., Chuang, I.L. ‘Bulk spin-resonance quantum computation,’ *Science* **275**, 350-356 (1997).
  78. Chuang, I.L., Vandersypen, L.M.K., Zhou, X., Leung, D.W., Lloyd, S., ‘Experimental

- realization of a quantum algorithm,' *Nature* **393**, 143-146 (1998) May, 1998.
79. Jones, J.A., Mosca, M., Hansen, R.H., 'Implementation of a quantum search algorithm on a quantum computer,' *Nature* **393**, 344-346 (1998).
  80. Chuang, I.L., Gershenfeld, N., Kubinec, M., 'Experimental implementation of fast quantum searching,' *Phys. Rev. Lett.* **80**, 3408-3411 (1998).
  81. Kane, B., 'A Silicon-Based Nuclear-Spin Quantum Computer,' *Nature* **393**, 133 (1998).
  82. Nakamura, Y., Pashkin, Yu.A., Tsai, J.S., 'Coherent control of macroscopic quantum states in a single-Cooper-pair box,' *Nature* **398**, 786-788 (1999)
  83. Mooij, J.E., Orlando, T.P., Levitov, L., Tian, L., van der Wal, C.H., Lloyd, S., 'Josephson persistent-current qubit,' *Science* **285**, 1036-1039 (1999).
  84. Lloyd, S., Braunstein, S., 'Quantum computation over continuous variables,' *Phys. Rev. Lett.* **82**, 1784-1787 (1999).
  85. Abrams, D., Lloyd, S., 'Nonlinear quantum mechanics implies polynomial-time solution for NP-complete and #P problems,' *Phys. Rev. Lett.* **81**, 3992-3995, (1998).
  86. Freedman, M., unpublished.
  87. Zel'dovich, Ya. B., Novikov, I.D., *Relativistic Astrophysics*, (Univ. of Chicago Press, Chicago), 1971.
  88. Page, D.N., 'Particle Emission Rates From a Black-Hole: Massless Particles from an Uncharged Non-Rotating Black-Hole,' *Phys. Rev. D* **13**, 198 (1976).
  89. K.S. Thorne, R.H. Price, D.A. Macdonald, *Black Holes: The Membrane Paradigm*, (Yale, New Haven) 1986. In particular, chapter VIII, 'The Thermal Atmosphere of a Black Hole,' by K.S. Thorne, W.H. Zurek, and R.H. Price.
  90. Novikov, I.D., Frolov, V.P., *Black Holes*, (Springer-Verlag, Berlin) 1986.
  91. Coleman, S., Preskill, J., Wilczek, F., 'Growing Hair on Black-Holes,' *Phys. Rev. Lett.* **67** 1975-1978 (1991).
  92. Preskill, J., 'Quantum Hair,' *Phys. Scr. T* **36**, 258-264 (1991).
  93. Fiola, T.M., Preskill, J., Strominger A., 'Black-Hole Thermodynamics and Information Loss in 2 Dimensions,' *Phys. Rev. D* **50**, 3987-4014 (1994).
  94. Susskind, L., Uglum, J., 'Black-Hole Entropy in Canonical Quantum-Gravity and Superstring Theory,' *Phys. Rev. D* **50**, 2700-2711 (1994).
  95. Strominger A., Vafa, C., 'Microscopic origin of the Bekenstein-Hawking entropy,' *Phys. Lett. B* **37**, 99-104 (1996).
  96. Das, S.R., Mathur, S.D., 'Comparing decay rates for black holes and D-branes' *Nucl.*

*Phys. B* **478**, 561-576 (1996).

97. Sidney H. Kahana, private communication.
98. Pagels, H., *The Cosmic Code: quantum physics as the language of nature*, (Simon and Schuster, New York) 1982.
99. In this paper we have explored only one aspect of how a black hole could be used for computation. Jim Hartle (private communication) has suggested some alternative methods:
  - (i) Set up a foundation to perform a very long computation to which you wish to know the answer. Then retire to near the horizon of the hole. Since your clock is redshifted with respect to the foundation's computer's clock, when you emerge from the hole a few minutes later by your clock, many years may have passed by the external clock (i.e., the black hole can be used as a time machine to travel into the future). As long as you have left the foundation in reliable hands so that it actually performed the desired computation, you can now get the answer.
  - (ii) If you really want the answer to the computation sufficiently badly, you can use the following feature of certain black holes. In some black holes there are time-like curves passing through the horizon that intersect all other time-like curves that pass the horizon at later external time (M. Simpson and R. Penrose, *Int. J. Theor. Phys.* **7**, 183-197 (1973); S. Chandrasekar and J.B. Hartle, *Proc. R. Soc. Lond. Ser. A* **384**, 301-315 (1982)): an observer travelling along such a curve will obtain during a finite interval of his time all information that subsequently falls into the hole. In this case, the foundation you set up can take an arbitrarily long time to perform the computation. Of course, you will be stuck inside the hole when you get the answer, which you will have to extract from the highly blue-shifted radiation falling into the hole. (In addition, the fact that you are obtaining a potentially infinite amount of information in a finite time suggests that travelling along such trajectories may be hazardous.)

## Figure 2: Computing at the Black-Hole Limit

The rate at which the components of a computer can communicate is limited by the speed of light. In the ultimate laptop, each bit can flip  $\approx 10^{19}$  times per second, while the time to communicate from one side of the one liter computer to the other is on the order of  $10^9$  seconds: the ultimate laptop is highly parallel. The computation can be speeded up and made more serial by compressing the computer. But no computer

can be compressed to smaller than its Schwarzschild radius without becoming a black hole. A one-kilogram computer that has been compressed to the black hole limit of  $R_S = 2Gm/c^2 = 1.485 \times 10^{-27}$  meters can perform  $5.4258 \times 10^{50}$  operations per second on its  $I = 4\pi Gm^2/\ln 2\hbar c = 3.827 \times 10^{16}$  bits. At the black-hole limit, computation is fully serial: the time it takes to flip a bit and the time it takes a signal to communicate around the horizon of the hole are the same.

I hope that you enjoyed reading that paper. I enjoyed writing it. Shortly after writing “Ultimate physical limits to computation,” I realized that I had set my sights too low. Why stop at the ultimate laptop? If you give up the requirement that your ultimate computer be portable (in German, a laptop is called a *Shleptop*), then you can gain considerable extra power. In particular, how much computation can you perform if you enlist the entire universe to help you out? The idea of the universe as a computer dates back to a story by Isaac Asimov from (?). Asimov talked of first making entire planets into computers, then whole solar systems, etc. The idea was made famous by Douglas Addams’ “The Hitchhiker’s Guide to the Galaxy,” in which the earth was revealed to be a piece of the universal computer that was computing the answer to the ultimate question. You may recall that the answer was 42, but that they had to go back to compute the question. The book of Zuse and the work of Fredkin then suggested that the universe might be a classical cellular automaton.

I began working on quantum computation in 1990. My motivation was to investigate the possibility that the universe as a whole might profitably be thought of as a quantum computer. I conjectured that almost any physical dynamics gave rise to universal quantum computation. It was only after working on this problem for several years that I realized that one might be able to use the techniques of physics of information processing actually to build quantum computers. Over the next few years, I showed in a series of papers published in the *Journal of Modern Optics* and in *Physical Review* that this conjecture was correct. My 1993 *Science* article which showed for the first time how quantum computers could be built by the methods described in the previous chapter. Then in 1994, I published a paper in *Physical Review Letters*, “Quantum computers and uncomputability,” in which

I noted that the if the universe was capable of universal quantum computation, then its future behavior was intrinsically unpredictable, because of the the Halting problem — the only way to see what that future will bring is to let the universe keep on computing it. In 1999, I published another paper, “Universe as quantum computer,” in which I showed that if the universe were a quantum computer, then it would automatically evolve complex, adaptive systems such as life (more on this below). In the meanwhile, I was working with experimentalists to build quantum computers: it was the insights gained through this quantum-mechanical engineering that allowed me to understand in detail how the universe processes information at the level of atoms, electrons, and photons.

After my experimental colleagues had taught me more about how the universe actually works, I was in a position to analyze the information-processing power of the universe in detail. The result was the following paper. Once again, I’ll comment along the way and provide suggestions for bleeping. If you have had enough scientific papers, then just read the abstract and bleep over the rest. (A paper by Mark Coffey that arrived independently at the same conclusions appeared several months after my paper was published.)

This paper was rejected by *Nature*: “Too much of a good thing.” But the referees at *Physical Review Letters* were more kindly. They verified the calculations, suggested further references to string theory, and retained a sense of humor throughout the refereeing process. As you can imagine from the descriptions of refereeing back and forth above, a sense of humor is rare in peer review. I do not know who they are, but I thank them.

### **Computational capacity of the universe**

*Once again, there’s no sense in having dull title. Far too many of my papers are entitled things like, “Compensation of decoherence from telegraph noise by means of bang-bang control.” When you have the chance to give a fun title, you should take it.*

Seth Lloyd

d’Arbeloff Laboratory for Information Systems and Technology

*Thank you Alex for the lunchtime conversation!*

MIT Department of Mechanical Engineering  
MIT 3-160, Cambridge, Mass. 02139  
slloyd@mit.edu

*Abstract:* All physical systems register and process information. The laws of physics determine the amount of information that a physical system can register (number of bits) and the number of elementary logic operations that a system can perform (number of ops). The universe is a physical system. This paper quantifies the amount of information that the universe can register and the number of elementary operations that it can have performed over its history. The universe can have performed  $10^{120}$  ops on  $10^{90}$  bits ( $10^{120}$  bits including gravitational degrees of freedom).

*In other words, a lot of ops on a lot of bits. Actually, when I calculated the numbers, my first response was “Those aren’t big numbers. I can think of much bigger numbers than that!” But those are indeed the numbers.*

*Though still designed for a broad audience, Physical Review Letters is a little more technical than Nature. So you’re allowed formulae in the first paragraph. Here the formulae just state how many elementary logical operations can have been performed by the universe and how many bits it can register, as a function of the age of the universe and the fundamental constants of nature. In this case, the number of ops and number of bits are simple functions of the age of the universe divided by the Planck time, the very short time at which quantum gravitational effects come into play.*

A recent paper by the author<sup>1</sup> put bounds on the amount of information processing that can be performed by physical systems. In particular, the Margolus-Levitin theorem<sup>2</sup> implies that the number of elementary logical operations per second that a physical can perform is limited by the system’s energy, and the amount of information that the system can register is limited by its maximum entropy<sup>1,3</sup>. As shown in (1), these bounds are actually attained by existing quantum computers.<sup>4–7</sup> The universe is a physical system. This paper applies these bounds to quantify the amount of information processing that can have be performed by the universe as a whole since the big bang. In particular, the universe can be shown to have the capacity to perform a maximum of  $(t/t_P)^2 \approx 10^{120}$

elementary quantum logic operations on  $(t/t_P)^{3/4} \approx 10^{90}$  bits registered in quantum fields (with a potential for  $(t/t_P)^2 \approx 10^{120}$  bits if gravitational degrees of freedom are taken into account). Here,  $t \approx 10^{10}$  years is the age of the universe and  $t_P = \sqrt{G\hbar/c^5} = 5.391 \times 10^{-44}$  seconds is the Planck time — the time scale at which gravitational effects are of the same order as quantum effects. If the universe is closed, then these numbers represent the amount ops and bits available in the entire universe, so that the total number of ops that can be performed over the entire lifetime  $T$  of a closed universe is  $\approx (T/t_P)^2$ . If the universe is open, and infinite in extent, then these numbers give the amount of computation that can have been performed within the part of the universe with which we are causally connected, i.e., the part within the horizon.

*The next paragraph tells how these numbers can be interpreted. These possible interpretations are complementary to each other: they all work together to give a unified picture of how much computation can take place in the universe.*

These numbers of ops and bits can be interpreted in three distinct ways:

1. They give upper bounds to the amount of computation that can have been performed by all the matter in the universe since the universe began.
2. They give lower bounds to the number of ops and bits required to simulate the entire universe on a quantum computer.<sup>8–10</sup>
3. If one chooses to regard the universe as performing a computation, these numbers give the numbers of ops and bits in that computation.

*The following paragraph indicates that for the purposes of calculating the computational capacity of the universe, I adopted a conservative and sober model of what has happened in the universe up until now. The paper is already wacky enough without having to use weird and speculative physics. In fact, despite the paper's ambitious purpose, all of its physics is straightforward and well-established.*

To calculate the number of bits available and the number of ops that can have been performed over the history of the universe requires a model of that history. The standard big-bang model will be used here.<sup>11</sup> In this model the universe began  $\approx 10^{10}$  years ago in what resembled a large explosion (the big bang). Since the big bang, the universe has expanded to its current size. The well-established inflationary scenario will be used to

investigate computation in the pre-big bang regime.<sup>12</sup> For the sake of compactness, the effects of possible extra dimensions, pre-big Bang physics, sub-Planck scale physics, etc., will not be considered here. In other words, we will content ourselves with evaluating the number of ops and number of bits available in the part of the universe that is accessible by observation (the part within the horizon) and for which well-established physical models exist. The techniques of ref. 1 could also be used to calculate computational capacities in more speculative models, as long those models obey the laws of quantum mechanics.

*To make the paper easier to read, I decided not to keep track explicitly of every little factor of two. This is a convention that I learned from attending conferences on astrophysics: if you're talking about really big numbers, and your answer is correct to within an order of magnitude or so, then you're doing fine.*

Now let us calculate the computational capacity of the universe. For clarity of exposition, the calculation will not explicitly keep track of factors of  $1/2, \pi$ , etc., that only affect the final results by an order of magnitude or so (i.e., a factor of  $2\pi$  will be ignored; a factor of  $(2\pi)^6$  will not be). The expression  $\approx$  will be used to indicate equality to within such factors. In other words,  $X \approx Y$  is equivalent to  $\log X = \log Y + O(1)$ .

*As noted in the introduction, the universe has gone through many phases in its existence. I calculated the number of bits and ops in all phases for which there are decent, well-established models. For the purposes of this paper, I concentrated on the most recent and longest stage, in which most of the energy in the universe is concentrated in matter, such as atoms. As I myself am made of atoms, I have an atomocentric view of the universe.*

For most of its history, the universe has been matter-dominated – most of the energy is in the form of matter. As will be seen below, most of the computation that can have taken place in the universe occurred during the matter-dominated phase. Accordingly, begin with the computational capacity of the matter-dominated universe, and then work back through the radiation-dominated and inflationary universe.

*Now let's get down to business. We saw in the previous paper how energy limits the rate of computation. The first paragraph below reviews this result.*

First, investigate the number of elementary logic operations that can have been performed. The maximum number of operations per second that can be performed by a

physical system is proportional to its energy.<sup>1-2</sup> This result follows from the Margolus-Levitin theorem, which states that the minimum time required for a physical system to move from one state to an orthogonal state is given by  $\Delta t = \pi\hbar/2E$ , where  $E$  is the average energy of the system above its ground state.<sup>2</sup> Since quantum logic operations involve flipping bits and moving from one state to an orthogonal state, the Margolus-Levitin theorem also gives the limit on how fast one can perform a quantum logic operation given energy  $E$ . Note that while energy must be invested in the spin-field interaction to flip the bit, it need not be dissipated.<sup>1-3</sup> The Margolus-Levitin bound also holds for performing many logic operations in parallel. If energy  $E$  is divided up among  $N$  quantum logic gates, each gate operates  $N$  times more slowly than a single logic gate operating with energy  $E$ , but the maximum total number of operations per second remains the same.

*This next paragraph establishes just what we are talking about when we say “the universe.” It is possible that the universe actually goes on forever, in which case the number of ops and number of bits are infinite. But even if it goes on forever, we can only see and interact with a finite part of it. The part we can see or interact with is called the part within the horizon. The amount of computation that can have taken place within the horizon since the big bang is finite.*

Now apply these results to the universe as a whole. In the matter-dominated universe, the energy within a co-moving volume is approximately equal to the energy of the matter within that volume and remains approximately constant over time. (A co-moving volume is one that is at rest with respect to the microwave background, and that expands as the universe expands.) Since the energy remains constant, the number of ops per second that can be performed by the matter in a co-moving volume remains constant as well. The total volume of the universe within the particle horizon is  $\approx c^3t^3$ , where  $t$  is the age of the universe. The particle horizon is the boundary between the part of the universe about which we could have obtained information over the course of the history of the universe and the part about which we could not. In fact, the horizon is currently somewhat further than  $ct$  away, due to the ongoing expansion of the universe, but in keeping with the approximation convention adopted above we will ignore this factor along with additional geometric factors in estimating the current volume of the universe.

*The following equations estimate the total amount of energy in the universe. The amount of matter is known to within a factor of ten or so, and the amount of energy is got from the*

amount of matter by Einstein's famous formula,  $E = mc^2$ . The amount of energy together with the age of the universe is then used calculate how many elementary quantum logical operations could have been performed since the big bang.

The total number of ops per second that can be performed in the matter-dominated universe is therefore  $\approx \rho c^2 \times c^3 t^3 / \hbar$ , where  $\rho$  is the density of matter and  $\rho c^2$  is the energy density per unit volume. Since the number of ops per second in a co-moving volume is constant, and since the universe has been matter dominated for most of its history, we have

$$\#\text{ops} \approx \rho c^5 t^4 / \hbar. \tag{1}$$

Insertion of current estimates for the density of the universe  $\rho \approx 10^{-27} \text{kg/m}^3$  and the age of the universe  $t \approx 10^{10}$  years, we see that the universe could have performed  $\approx 10^{120}$  ops in the course of its history.

*An interesting feature of our universe is that it seems to be poised just at the point between where it will expand forever, and where it recontracts into a big crunch. The density of the universe at this point is called the critical density. The critical density has a simple formula that allows us to relate the total number of ops performed over the history of the universe to the age of the universe divided by the Planck time. Note that although the Planck time usually refers to times at which quantum gravity is important, this paper makes no assumptions about quantum gravity. Instead the Planck time arises naturally from the large-scale structure of the universe, as embodied by the critical density of matter, combined with the fundamental limits of information processing given by quantum mechanics. Of course, the fact that the Planck time shows up in the context of the number of ops performed by the universe over its history suggests that there may well be a basic connection between quantum gravity and quantum computation. This connection will be explored in greater depth below.*

A more revealing form for the number of ops can be obtained by noting that our universe is close to its critical density. If the density of the universe is greater than the critical density, it will expand to a maximum size and then contract. If the density is less than or equal to the critical density, it will expand forever. Because of the expansion of the universe, a galaxy at distance  $R$  is moving away with velocity  $HR$ , where  $H$  is the Hubble constant. At the critical density, the kinetic energy  $mH^2 R^2 / 2$  of this galaxy is

equal to its gravitational energy  $4\pi Gm\rho_c R^3/3R$ , so that  $\rho_c = 3H^2/8\pi G \approx 1/Gt^2$ . So for a matter-dominated universe at its critical density, constant, the total number of ops that can have been performed within the horizon at time  $t$  is

$$\#\text{ops} \approx \rho_c c^5 t^4 / \hbar \approx t^2 c^5 / G \hbar = (t/t_P)^2. \quad (2)$$

*For the sake of completeness, I also analyze the case of a closed universe that recollapses to a big crunch. Such a universe is finite in both space and time. The total number of ops that can be performed in the history of closed universe takes on a particularly simple form.*

A matter-dominated universe whose density is higher than the critical density is closed:<sup>11</sup> it is spatially finite, expanding to a maximum length scale  $a_{max}$  over a time  $T = \pi a_{max}/2c$ , and temporally finite, recontracting to a singularity over a time  $2T$ . The energy available for computation is  $Mc^2 = 3\pi c^4 a_{max}/4G$ . As a result, the total number of ops that can be performed over the entire history of a closed, matter-dominated universe is

$$2TMc^2/\hbar = (3\pi^2/4)(a_{max}/\ell_P)^2 = 3(T/t_p)^2,$$

where  $\ell_p = t_p c = 1.616 \times 10^{-35}$  meters is the Planck length.

*It is customary to compare the computational capacity of an ultimate computer with the piddling power of conventional electronic computers.*

It is instructive to compare the total number of operations that could have been performed using all the matter in the universe with the number of operations that have been performed by conventional computers. The actual number of elementary operations performed by all human-made computers is of course much less than this number. Because of Moore's law, about half of these elementary operations have been performed in the last two years. Let us estimate the total number of operations performed by human-made computers, erring on the high side. With  $\approx 10^9$  computers operating at a clock rate of  $\approx 10^9$  Hertz performing  $\approx 10^5$  elementary logical operations per clock cycle over the course of  $\approx 10^8$  seconds, all the human-made computers in the world have performed no more than  $\approx 10^{31}$  ops over the last two years, and no more than approximately twice this amount in the history of computation.

*What is the universe computing? It computes itself.*

What is the universe computing? In the current matter-dominated universe most of the known energy is locked up in the mass of baryons. If one chooses to regard the universe as performing a computation, most of the elementary operations in that computation consists of protons, neutrons (and their constituent quarks and gluons), electrons and photons moving from place to place and interacting with each other according to the basic laws of physics. In other words, to the extent that most of the universe is performing a computation, it is ‘computing’ its own dynamical evolution.<sup>13</sup> Only a small fraction of the universe is performing conventional digital computations.

*Now turn to number of bits that the universe can register. This number is just equal to the maximum possible entropy of the universe.*

Now calculate the number of bits that can be registered by the universe. The amount of information, measured in bits, that can be registered by any physical system is equal to the logarithm to the base 2 of the number of distinct quantum states available to the system given its overall energy, volume, electric charge, etc.<sup>2</sup> In other words,  $I = S/k_B \ln 2$ , where  $S$  is the maximum entropy of the system and  $k_B = 1.38 * 10^{-23}$  joule/ $K$  is Boltzmann’s constant.

*The number of bits can be calculated by methods that are a century old. These formulae were derived by Planck in the very first papers on quantum mechanics around 1900.*

To calculate the number of bits that can be registered by the universe requires a calculation of its maximum entropy, a calculation familiar in cosmology. The maximum entropy in the matter-dominated universe would be obtained by converting all the matter into radiation. (Luckily for us, we are not at maximum entropy yet!) The energy per unit volume is  $\rho c^2$ . The conventional equation for black-body radiation can then be used to estimate the temperature  $T$  that would be obtained if that matter were converted to radiation at temperature  $T$ :  $\rho c^2 = (\pi^2/30\hbar^3 c^3)(k_B T)^4 \sum_{\ell} n_{\ell}$ . Here  $\ell$  labels the species of effectively massless particles at temperature  $T$  (i.e.,  $m_{\ell} c^2 \ll k_B T$ ), and  $n_{\ell}$  counts the number of effective degrees of freedom per species:  $n_{\ell} =$  (number of polarizations) \* (number of particles/antiparticles) \* 1 (for bosons) or 7/8 (for fermions). Solving for the temperature for the maximum entropy state gives  $k_B T = (30\hbar^3 c^5 \rho / \pi^2 \sum_{\ell} n_{\ell})^{1/4}$ . The maximum entropy per unit volume is  $S/V = 4\rho c^2 / 3T$ . The entropy within a volume  $V$

is then  $S = (4k_B/3)(\pi^2 \sum_\ell n_\ell/30)^{1/4}(\rho c/\hbar)^{3/4}V^{1/4}$ . The entropy depends only weakly on the number of effectively massless particles.

*Applying Planck's formulae to the question of the number of bits in the universe reveals a simple formula in terms of fundamental constants.*

Using the formula  $I = S/k_B \ln 2$  and substituting  $\approx c^3 t^3$  for the volume of the universe gives the maximum number of bits available for computation:

$$I \approx (\rho c^5 t^4 / \hbar)^{3/4} = (\#\text{ops})^{3/4} \quad (3)$$

The universe could currently register  $\approx 10^{90}$  bits. To register this amount of information requires every degree of freedom of every particle in the universe.

*We don't know all the fundamental laws of physics. In particular, we still don't have a universally accepted theory of quantum gravity. Luckily, we do know quite a lot about how much information can be registered by gravitational systems. What we know suffices to get an answer, which is reported here.*

The above calculation estimated only the number of ops that could be performed and the amount of information that could be stored by matter and energy and did not take into account information that might be stored and processed on gravitational degrees of freedom. That the energy available in the gravitational field is equal in magnitude and opposite in sign to the energy in the matter fields.<sup>11</sup> Applying the Margolus-Levitin theorem to the number of ops that can have been performed by this energy yields the same number of ops that can be performed by the matter fields. Similarly, the Bekenstein bound<sup>14</sup> together with the holographic principle<sup>15-18</sup> implies that the maximum amount of information that can be registered by any physical system, including gravitational ones, is equal to the area of the system divided by the the square of the Planck length,  $\ell_P^2 = \hbar G/c^3$ . This limit is in fact attained by black holes and other objects with event horizons. Applying the Bekenstein bound and the holographic principle to the universe as a whole implies that the maximum number of bits that could be registered by the universe using matter, energy, and gravity is  $\approx c^2 t^2 / \ell_P^2 = t^2 / t_P^2$ . That is, the maximum number of bits using gravitational degrees of freedom as well as conventional matter and energy is equal to the maximum number of elementary operations that could be performed in the universe,  $\approx 10^{120}$ .

*It's time again for invidious comparisons to existing computers*

Not surprisingly, existing human-made computers register far fewer bits. Over-estimating the actual number of bits registered in 2001, as above for the number of ops, yields  $\approx 10^9$  computers, each registering at  $\approx 10^{12}$  bits, for a total of  $\approx 10^{21}$  bits.

*Now go back further in the big bang to the time when the universe was made of radiation, and even further to the time when it was made of who knows what. The same formulae still apply.*

Using the same methods, one can calculate the number of ops and number of bits available in the radiation-dominated and inflationary universes. Despite the radically different forms of matter, formulae (2) and (3) can be shown to hold for the radiation-dominated universe, and formula (2) holds for the inflationary universe. The number of available bits in the inflationary universe is dominated by the number of bits in the horizon radiation, and is equal to  $\approx (t/t_P)^2$ . Of course, the character of the ‘computation’ in these universes is very different from the matter-dominated universe. In particular, the matter in the radiation dominated universe is primarily at thermal equilibrium, making it a hostile environment for complex processes such as life. The inflationary universe is divided into causally non-communicating sectors: the primary computational process in inflationary universe is ‘bit creation’ — the production of large quantities of spatial volume and of free energy that will later on be used for more complicated computation in the matter-dominated universe.<sup>16–18</sup> A more full account of the details of information processing in the radiation-dominated and inflationary universes can be found in (19).

*The following is basically cosmic numerology. Three quarters of a century ago, the astrophysicist Sir Arthur Eddington, and the Nobel laureate Paul Dirac noted that the universe is characterized by several large numbers, all of which happen to be approximately equal to ten raised to the fortieth power. Dirac hypothesized that something about the laws of physics required these large numbers to be approximately equal; however, this approximate equality is currently regarded by most scientist as a coincidence. The large numbers of ops and bits caclulated here are closely related to the Eddington-Dirac large number. This is not a coincidence. The Eddington-Dirac large number is also close to ten raised to the power 42, where 42, as note above, is the putative answer to the universal question raised by the Hitchhiker’s Guide. Coincidence?*

Before concluding, it is worth noting that the number of bits and number of operations possible in the universe are related to the Eddington/Dirac large number hypothesis. Three quarters of a century ago, Eddington noted that two large numbers that characterize our universe happen to be approximately equal.<sup>20</sup> In particular, the ratio between the electromagnetic force by which a proton attracts an electron and the gravitational force by which a proton attracts an electron is  $\alpha = e^2/Gm_em_p \approx 10^{40}$ . Similarly, the ratio between the size of the universe and the classical size of an electron is  $\beta = ct/(e^2/m_e c^2) \approx 10^{40}$ . The fact that these two numbers are approximately equal is currently regarded by most researchers as a coincidence. (A third large number, the square root of the number of baryons in the universe,  $\gamma = \sqrt{\rho c^3 t^3/m_p}$  is also  $\approx 10^{40}$ . This is not a coincidence given the values of  $\alpha$  and  $\beta$ :  $\alpha\beta \approx \gamma^2$  in a universe near its critical density  $\rho_c \approx 1/Gt^2$ .)

The astute reader may have noted that the number of operations that can have been performed by the universe is approximately equal to the Eddington-Dirac large number cubed. In fact, the number of ops is necessarily approximately equal to  $\beta\gamma^2 \approx \alpha\beta^2 \approx 10^{120}$ . This relation holds true whether or not  $\alpha \approx \beta \approx \gamma$  is a coincidence. In particular,

$$\beta\gamma^2 = (\rho c^5 t^4/\hbar)(\hbar c/e^2)(m_e/m_p) = \#ops * (137/1836) \quad (4a)$$

Similarly,

$$\alpha\beta^2 = (t/t^P)^2(\hbar c/e^2)(m_e/m_p) = \#ops * (137/1836) \quad (4b)$$

That is, the number of ops differs from the Eddington-Dirac large number cubed by a factor of the fine structure constant times the proton-electron mass ratio. Since the number of ops is  $\approx 10^{120}$ , as shown above, and the fine-structure constant times the proton-electron mass ratio is  $\approx 10$ , the number of ops is a factor of ten larger than the Eddington-Dirac large number cubed. In other words, whether or not the approximate equality embodied by the Eddington-Dirac large number is a coincidence, the fact that the number of operations that can have been performed by the universe is related to this large number is not.

*Just what do the number of ops and bits available to the universe mean? No one knows exactly. I close by discussing the various interpretations of the number of ops performed by the universe and the number of bits it registers. For fairness sake, I mention more conservative interpretations than my own, which is that the universe is in fact a quantum computer, and the numbers calculated here simply represent how much computation it has performed to date.*

The above sections calculated how many elementary logical operations that can have been performed on how many bits during various phases of the history of the universe. As noted above, there are three distinct interpretations of the numbers calculated. The first interpretation simply states that the number of ops and number of bits given here are upper bounds on the amount of computation that can have been performed since the universe began. This interpretation should be uncontroversial: existing computers have clearly performed far fewer ops on far fewer bits.<sup>21</sup>

The second interpretation notes that the numbers calculated give a lower bound on the number of bits and the number of operations that must be performed by a quantum computer that performs a direct simulation of the universe. This interpretation should also be uncontroversial: quantum computers can accurately simulate any physical system that evolves according to local interactions, using the same amounts of energy and Hilbert space volume as the system itself.<sup>7–10</sup> It is an open question as to how to simulate quantum gravity, but string theory and M theory provide potential theories of quantum gravity<sup>22</sup> and these theories should also be accessible to efficient simulation on a quantum computer. If so, then quantum computation might provide an alternative formulation for a ‘theory of everything.’

The third interpretation — that the numbers of bits and ops calculated here represent the actual memory capacity and number of elementary quantum logic operations performed by the universe — is more controversial. That the universe registers an amount of information equal to the logarithm of its number of accessible states seems reasonable. And virtually all physical interactions can operate as quantum logic gates. But whether or not it makes sense to identify an elementary quantum logic operation with the local evolution of information-carrying degrees of freedom by an average angle of  $\pi/2$  a question whose answer must await further developments in the relationship between physics and computation.

The amount of information the universe can register and the amount of information processing it can perform can be calculated using the physics of information processing. To date, the universe can have performed  $10^{120}$  ops on  $10^{90}$  bits ( $10^{120}$  bits if quantum gravity is taken into account), enough to factor a million bit number using the classical number field sieve algorithm, and enough to factor a  $10^{60}$  bit number using Shor’s quantum algorithm. Is the universe a computer? It is certainly *not* a digital computer running Linux or Windows. But the universe certainly does represent and process quantifiable amounts

of information in a systematic fashion.

## References:

Once more, references are key to allowing the reader to follow up on the work. Here I took pains to refer to Paola Zizzi and Jack Ng, who have also worked on the computational picture of the universe.

1. Lloyd, S., *Nature* **406**, 1047-1054 (2000).
2. Margolus, N., Levitin, L.B., in *PhysComp96*, T. Toffoli, M. Biafore, J. Leao, eds. (NECSI, Boston) 1996; *Physica D* **120**, 188-195 (1998).
3. Landauer, R., *Nature*, **335**, 779-784 (1988).
4. Turchette, Q.A., Hood, C.J., Lange, W., Mabuchi, H., Kimble, H.J., *Phys. Rev. Lett.*, **75**, 4710-4713 (1995).
5. Monroe, C., Meekhof, D.M., King, B.E., Itano, W.M., Wineland, D.J., *Phys. Rev. Lett.*, **75**, 4714-4717 (1995).
6. Cory, D.G., Fahmy, A.F., Havel, T.F., in *PhysComp96, Proceedings of the Fourth Workshop on Physics and Computation*, T. Toffoli, M. Biafore, J. Leão, eds., (New England Complex Systems Institute, Boston) 1996.
7. Gershenfeld, N.A., Chuang, I.L. *Science* **275**, pp. 350-356 (1997).
8. Lloyd, S., *Science*, **273**, 1073-1078 (1996).
9. Abrams, D., Lloyd, S., *Phys. Rev. Lett.* **81**, 3992-3995, (1998).
10. Zalka, C., *Proc. R. Soc. Lond A. Mat.* **454**, 313-322 (1998).
11. Peebles, P.J.E., *Principles of Physical Cosmology*, Princeton University Press, Princeton (1993).
12. Linde, A., *Particle Physics and Inflationary Cosmology*, Harwood Academic, New York (1990).
13. Pagels, H., *The Cosmic Code: quantum physics as the language of nature*, (Simon and Schuster, New York) 1982.
14. Bekenstein, J.D., *Phys. Rev. D.* **23**, 287 (1981); *Phys. Rev. Letters* **46**, 623 (1981); *Phys. Rev. D.* **30**, 1669-1679 (1984).
15. 't Hooft, G., gr-qc/9310026; Susskind, L., *J. Math. Phys.* **36**, 6377 (1995).
16. Zizzi, P., *Entropy* **2**, 36-39 (2000).
17. Zizzi, P., 'The Early Universe as a Quantum Growing Network,' in Proceedings of IQSA Fifth Conference, March 31-April 5, 2001, Cesena-Cesenatico, Italy; gr-qc/0103002.

18. Ng, J., *Phys. Rev. Lett.* **86**, 2946-2947 (2001).
19. Lloyd, S., ArXiv/quant-ph/0110141.
20. Eddington, A.S., *The Mathematical Theory of Relativity*, Cambridge University Press, Cambridge (1924).
21. Fredkin, E., ‘Ultimate Limits to Computation,’ MIT Laboratory for Computer Science Seminar (2000).
22. Banks, T., Fischler, W., Shenker, S.H., Susskind, L. ‘M Theory As A Matrix Model: A Conjecture,’ *Phys. Rev. D* **55**, S112-S118 (1997).

We know how the universe is computing. We know how much the universe is computing. “So what?” you may ask, “Just what does this picture of the universe as a quantum computer buy me that I didn’t have already?” After all, we have a perfectly good quantum-mechanical theory of elementary particles. So what if these particles are also processing information and performing computations? Do we really need a whole new paradigm for thinking about how the universe operates?

These are reasonable questions. Let’s start with the last. The conventional picture of the universe in terms of physics is based on the paradigm of the universe as a machine. A paradigm is a picture or model, a way of thinking about the world. The word, paradigm comes from ancient Greek,  $\pi\alpha\rho\alpha$ , beside or along (as in parallel) and  $\delta\iota\gamma\mu\omicron\varsigma$ , demonstration. Contemporary physics is based on the mechanistic paradigm, in which the world is analyzed in terms of its underlying mechanisms. The mechanistic paradigm is the basis for modern science. A beautiful expression of the mechanistic paradigm can be found in the opening paragraphs of Thomas Hobbes’s *Leviathan*, his massive treatise on the political state:

‘ Nature, the art whereby God hath made and governs the world, is by the *art* of man, as in many other things, so in this also imitated, that it can make an artificial animal. For, seeing life is but a motion of limbs, the beginning whereof is in some principal part within; why may we not say, that all *automata* (engines that move themselves by springs and wheels as doth a watch) have an artificial life? For what is the heart, but a spring; and the nerves, but so many strings, and the joints, but so many wheels, giving motion to the whole body, such as was intended by the artificer? Art goes yet

further, imitating that rational and most excellent work of nature, *man*.”

Note that in addition to espousing the mechanistic paradigm, Hobbes gives an early of the notion of artificial life. The 21st century scientific field of artificial life concentrates on creating computerized machines that emulate aspects of living creatures.

Paradigms are highly useful: they allow one to think about the world in a new way. Thinking about the world as a machine allowed virtually all advances in science, including physics, chemistry, and biology. The primary quantity of interest in the mechanistic paradigm is energy. This book advocates a new paradigm that is an extension of the powerful mechanistic paradigm: I suggest thinking about the world not just as a machine, but as a machine that processes information. In this paradigm there are two primary quantities, energy and information, standing on an equal footing and playing off each other.

Just in the way that thinking about the body in terms of clockwork allowed insight into physiology (and in Hobbes’s case, to the inner workings of the body politic), the computational universe paradigm allows new insights into the way the universe works. Perhaps the most important new insight afforded by thinking of the world in terms of information is the resolution of the problem of complexity. The conventional mechanistic paradigm gives no simple answer to why the universe in general, and life on earth in particular, is so complex. In the computational universe, by contrast, the innate information processing power of the universe systematically gives rise to all possible types of order, simple and complex. This feature of the computational universe was proposed in the introduction, and will be investigated in detail in the next and final chapter.

A second insight that the computational affords is into the question of how and why the universe began in the first place. As noted above, one of the great outstanding problems of physics is the problem of quantum gravity. In the beginning of the 20th century, Einstein proposed a beautiful theory for gravity, called general relativity. General relativity is one of the most elegant physical theories of all time, and accounts for many of the observed features of the universe at large scales. Quantum mechanics, counterintuitive as it is, accounts for virtually all observed features of the universe at small scales. To give a full picture of how the universe began, back when it was new, tiny, and incredibly energetic, requires a theory that unifies general relativity and quantum mechanics.

No such unifying theory currently exists. There have been many valiant attempts to construct a quantum-mechanical theory of gravity. A clear summary of these attempts can be found in Lee Smolin’s *Three Roads to Quantum Gravity*. But none of these roads

has yet reached its destination.

Quantum computation provides a fourth road to solving the problem of quantum gravity. As with the other approaches, lots of roadwork remains to be done. (And at any point along the development of such a theory, a fatal collision with experiment or observation can make the theory roadkill.) Here is a map to the quantum computational road to quantum gravity.

### **Quantum computation and quantum gravity**

Once you have grasped how quantum computations work, it is only a short additional distance to understanding how general relativity works, and how quantum computation could give rise to a unified theory of gravity together with elementary particles. To see how quantum computation leads to general relativity, think of the circuit diagram for a quantum computation (figure). The circuit diagram starts with a set of initial quantum bits, which can be taken to be all in the 0 state. (If one wants to start with some other state, then one can begin the computation with a circuit that constructs that state out of the state  $00\dots 0$ .) The qubits go down wires, which take them to quantum logic gates, where they interact. Additional wires then take them to other quantum logic gates, where they interact with other qubits. Any quantum computation can be built up from these simple elements.

The entire quantum computation is specified by its causal structure (the wires) together with its logical structure (the local quantum logic gates). To go from such a causal structure and logical structure to a general relativistic spacetime, corresponding, e.g., to our universe, now look and see what types of universes can correspond to the computation. General relativity is a theory of space and time, and of the interaction of space and time with matter. Each possible configuration of space and time interacting with matter is called a spacetime. For example, our universe is a particular spacetime.

In the computational universe paradigm, the concepts of space and time, together with their interaction with matter, are to be derived from an underlying quantum computation. That is, each quantum computation corresponds to a possible spacetime, whose features are derived from the features of the computation. Our goal is to show that the resulting computational spacetime obeys Einstein's theory of general relativity.

I'll now show that computational spacetime indeed implies general relativity. Imagine the quantum computation as embedded in space and time. Each logic gates now sits at a point in space and time, and wires represent paths along which quantum bits flow from

one point to another. The first feature to note is that there are many ways to embed the quantum computation in space and time. Each quantum logic gate can be put down at any point where there is not another quantum logic gate, and the wires can squiggle all over the place to connect the logic gates. What happens to quantum information in the computation is independent of how the quantum computation is embedded in the spacetime. In the language of general relativity, the dynamical content of the quantum computation is ‘generally covariant’ – the quantum computation just ‘doesn’t care’ how it is embedded in space and time as long as the qubits interact with each other in the right sequence.

The fact that a quantum computation doesn’t care about how it is embedded in spacetime automatically means that the spacetime derived from the quantum computation obeys almost all the laws of general relativity. Why? Because Einstein derived the laws of general relativity almost a century ago exactly by requiring that those laws don’t care how the underlying physical dynamics of matter is embedded in spacetime. Under the proper assumptions, general relativity is the *unique* theory of gravity that is generally covariant.

Now verify explicitly that the spacetime derived from the quantum computation obeys the laws of general relativity. The explicit derivation is somewhat mathematical, but can be summarized as follows. The wiring diagram for the quantum computation translates to a causal structure for spacetime. But it is a well-known result of general relativity that the causal structure of spacetime fixes almost all features of the spacetime. The only feature that remains to be fixed is the local length scale.

It’s straightforward to see why a local length scale is needed to determine the full structure of spacetime. Suppose that I measure a distance here in Cambridge using a stick marked off into equal subunits. I measure the distance along MIT’s ‘infinite corridor’ (a very long, but finite corridor that runs the length of the main building in which my office sits). I find that it is 42 units long. Now I send you an email message, wherever you are: “The infinite corridor is 42 units long.”

This email message conveys no information to you about the infinite corridor’s actual length unless you know the set of units that I am using. To convey to you the size of my unit, we need to establish a common standard of length. The way that this is conventionally done is to refer to the quantum-mechanical properties of matter, in particular, to the wave-like properties of quantum systems. For example, the meter is defined to be equal to  $X$  wavelengths of light emitted by a cesium atom. So if I tell you that my unit is equal to  $Y$

wavelengths of light emitted by a cesium atom, and if you have a cesium atom, then you now know how long the infinite corridor is in terms of your local length scale.

Now return to the computational universe. The only feature of spacetime that remains to be fixed is the local length scale, and this is to be fixed in terms of the wavelike properties of the local quantum-mechanical matter. The ‘matter’ in the computational universe arises out of the quantum logic gates. Recall that any form of quantum-mechanical matter that arises out of local interactions can effectively be simulated or constructed out of quantum logic gates. The quantum bits make up a sort of ‘quantum computronium’ that is capable of behaving like any elementary particle. Like a particle, each quantum logic gate corresponds to a wave, which wiggles up and down some number of times as the quantum bits are transformed by the quantum logic gate. The local length scale in the computational universe is determined by the number of wiggles in the quantum logic gate’s wave.

Once the quantum logic gates have fixed the local length scale, the structure of the computational spacetime is entirely fixed. Because the quantum computation is embedded in spacetime in a generally covariant way, the resulting spacetime necessarily obeys Einstein’s equations for general relativity. This agreement with general relativity can be verified by explicit calculation (which will not be presented here: see the published paper for the gnarly mathematical details).

In the computational universe, the structure of spacetime is derived from the structure of the underlying computation. The resulting spacetime obeys Einstein’s equations, which say how space and time respond to the presence of matter. The matter in the computational universe is constructed from quantum bits (“informatinos”) that interact via quantum logic gates. As noted above, these interacting quantum bits are perfectly capable of reproducing the phenomenology of the Standard Model for elementary particles. The resulting quantum computronium bends and warps the fabric of spacetime. Because the quantum computation doesn’t care about how it is embedded in spacetime, the resulting spacetime automatically obeys the laws of general relativity. To paraphrase John Wheeler, “It from qubit!”

The computational universe paradigm for the interaction of quantum-mechanics with general relativity represents a distinct road to quantum gravity: it travels through a very different landscape from Smolin’s three roads. But its final goal is the same. This paradigm is a work in progress: it makes explicit predictions for the behavior of the early universe

and for processes such as black hole evaporation. These explicit predictions can be tested by observation, for example, by detailed observations of the cosmic microwave background. Time will tell whether the computational universe paradigm for quantum gravity is a road to quantum gravity, or whether collision with observation and experiment will turn it into roadkill.

Despite the intrinsic uncertainty of science in the making, the theory of general relativity as a consequence of quantum computation has passed a scientific milestone that has not yet been passed along any of the other three roads. Because quantum computation so easily encompasses and reproduces quantum dynamics, the computational universe theory of quantum gravity is the first theory to combine general relativity and the Standard Model of elementary particles in a straightforward and self-consistent way. This accomplishment suggests that we follow the road of the computational universe with hope that it will lead us to our goal.