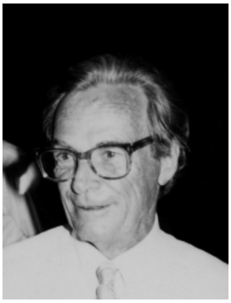
6
The Computing Machines in the Future

Richard P. Feynman

Abstract This address was presented by Richard P. Feynman as the Nishina Memorial Lecture at Gakushuin University (Tokyo), on August 9, 1985.



Richard P. Feynman
©NMF

It's a great pleasure and an honor to be here as a speaker in memorial for a scientist that I have respected and admired as much as Prof. Nishina. To come to Japan and talk about computers is like giving a sermon to Buddha. But I have been thinking about computers and this is the only subject I could think of when invited to talk.

The first thing I would like to say is what I am not going to talk about. I want to talk about the future computing machines. But the most important possible developments in the future, are things that I will not speak about. For example, there is a great deal of work to try to develop smarter machines, machines which have a better relationship with the humans so that input and output can be made with less effort than the complex programming that's necessary today. This goes under the name often of artificial intelligence, but I don't like that name. Perhaps the unintelligent machines can do even better than the intelligent ones. Another problem is the standardization of programming languages. There are too many languages today, and it would be a good idea to choose just one. (I hesitate to mention that in Japan, for what will happen will be that there will simply be more standard languages; you already have four ways of writing now and attempts to standardize anything here result apparently in more standards and not fewer.) Another interesting future

Richard P. Feynman (1918 – 1988). Nobel Laureate in Physics (1965)
California Institute of Technology (USA) at the time of this address

R. P. Feynman: *The Computing Machines in the Future*, Lect. Notes Phys. **746**, 99–113 (2008)
DOI 10.1007/978-4-431-77056-5_6

problem that is worth working on but I will not talk about, is automatic debugging programs; debugging means to fix errors in a program or in a machine. It is surprisingly difficult to debug programs as they get more complicated. Another direction of improvement is to make physical machines three dimensional instead of all on a surface of a chip. That can be done in stages instead of all at once; you can have several layers and then many more layers as the time goes on. Another important device would be a way of detecting automatically defective elements on a chip, then this chip itself automatically rewiring itself so as to avoid the defective elements. At the present time when we try to make big chips there are flaws, bad spots in the chips, and we throw the whole chip away. But of course if we could make it so that we could use the part of the chip that was effective, it would be much more efficient. I mention these things to try to tell you that I am aware of what the real problems are for future machines. But what I want to talk about is simple, just some small technical, physically good things that can be done in principle according to the physical laws; I would like in other words to discuss the machinery and not the way we use the machines.

I will talk about some technical possibilities for making machines. There will be three topics really. One is parallel processing machines which is something of the very near future, almost present, that is being developed now. Further in the future are questions of the energy consumption of machines which seems at the moment to be a limitation, but really isn't. Finally I will talk about the size; it is always better to make the machines smaller, and the question is how much smaller is it still possible to make machines according to the laws of Nature, in principle. I will not discuss which and what of these things will actually appear in the future. That depends on economic problems and social problems and I am not going to try to guess at those.

1. Parallel Computers

First about parallel programming, parallel computers, rather. Almost all the present computers, conventional computers, work on a layout or an architecture invented by von Neumann, in which there is a very large memory that stores all the information, and one central location that does simple calculations. We take a number from this place in the memory and a number from that place in the memory, send the two to the central arithmetical unit to add them and then send the answer to some other place in the memory. There is, therefore, effectively one central processor which is working very very fast and very hard while the whole memory sits out there like a vast filing cabinet of cards which are very rarely used. It is obvious that if there were more processors working at the same time we ought to be able to do calculations faster. But the problem is that some one who might be using one processor may be using some information from the memory that another one needs, and it gets very confusing. And so it has been said that it is very difficult to work many processors in parallel. Some steps in that direction have been taken in the larger conventional machines, what they call "vector processors". When sometimes you want to do exactly the same step on many different items you can do that perhaps at the same time. The ordinary hope is that the regular program can be written, and then an interpreter will discover automatically when it is useful to use this vector possibility. That idea is used in the Cray and in the super-computers in Japan. Another plan is to take what is effectively a large number of relatively simple (but not very simple) computers, and connect them all together in some pattern. Then they can all work on a part of the problem. Each one is really an independent computer, and they will transfer information to each other as one or another needs it. This kind of a scheme is in a machine for example called Cosmic Cube, and is one of the possibilities; many people are making such machines. Another plan is to distribute very large numbers of very simple central processors all over the memory. Each one deals with just a small part of the memory and there is an elaborate system of interconnections between them. An example of such a machine is the Connection Machine made at M.I.T. It has 64,000 processors and a system of routing in which every 16 can talk to any other 16 and thus 4000 routing connection possibilities. It would appear that scientific questions like the propagation of waves in some material might be very easily handled by parallel processing, because what happens in this part of space at a moment can be worked out locally and only the pressures and the stresses from the neighbor needs to be known for each section can be worked out at the same time, and communicate boundary conditions across. That's why this type of design is built for such a thing. But it has turned out that very large number of problems of all kinds can be dealt with in parallel. As long as the problem is big enough so that a lot of calculating has to be done, it turns out that a parallel computation can speed this up enormously, not just scientific problems.

And what happened to the prejudice of 2 years ago, which was that the parallel programming is difficult? It turns out that what was difficult, and almost impossible, is to take an ordinary program and automatically figure out how to use the parallel computation effectively on that program. Instead, one must start all over again with the problem, appreciating that we have parallel possibility of calculation, and rewrite the program completely with a new attitude to what is inside the machine. It is not possible to effectively use the old programs. They must be rewritten. That is a great disadvantage to most industrial applications and has met with considerable resistance. But, the big programs belong usually to scientists or others, unofficial intelligent programmers who love computer science and are willing to start all over again and rewrite the program if they can make it more efficient. so what's going to happen is that the hard programs, vast big ones, will first be programmed by experts in the new way, and then gradually everybody will have to come around, and more and more programs will be programmed that way, and programmers will just have to learn how to do it.

2. Reducing the Energy Loss

The second topic I want to talk about is energy loss in computers. The fact that they must be cooled is the limitation apparently to the largest computers; a good deal of the effect is spent in cooling machine. I would like to explain that this is simply a result of very poor engineering and is nothing fundamental at all. Inside the computer a bit of information is controlled by a wire which either has a voltage of one value or another value. It is called “one bit”, and we have to change the voltage of the wire from one value to the other and have to put change on or take charge off. I make an analogy with water: we have to fill a vessel with water and get one level or to empty it to get to the other level. It’s just an analogy. If you like electricity better you can think more accurately electrically. What we do now is analogous, in the water case, to filling the vessel by pouring water in from a top level (Fig. 6.1), and lowering the level by opening the valve at the bottom and letting it all run out. In both cases there is a loss of energy because of the drop of the water, suddenly, through a height say from top level where it comes in to the low bottom level when you start pouring it in to fill it up again. In the cases of voltage and charge, there occurs the same thing.

It’s like, as Mr. Bennett has explained, operating an automobile which has to start and stop by turning on the engine and putting on the brakes, turning on the engine and putting on the brakes; each time you lose power. Another way with a car would be to connect the wheels to flywheels. Stop the car and speed up the flywheel saving the energy, which can then be reconnected to start the car again. The analogy electrically or in the water would be to have a U-shaped tube with a valve at the bottom in the center connecting the two arms of the U (Fig. 6.2). When it is full here on the right but empty on the left with the valve closed, if we open that valve the water will slip out to the other side, and we close it just in time to catch it. Then when we want to go the other way we open the valve again and it slips to the other side and we catch it. There is some loss and it doesn’t climb as high as it did before, but all we have to do is to put a little water in to correct the little loss, a much smaller

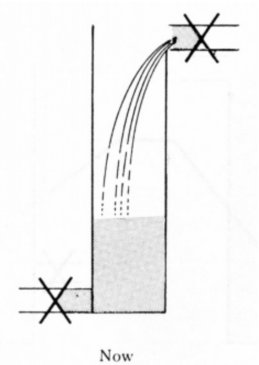


Fig. 6.1

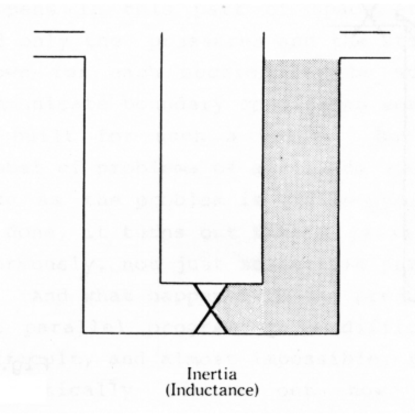


Fig. 6.2

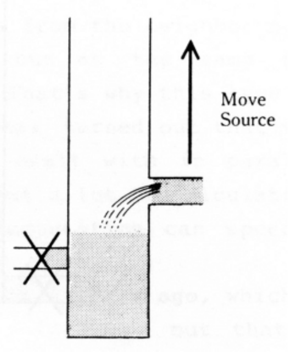


Fig. 6.3

energy loss than the direct fill method. But such a thing uses the inertia of the water and the analogue in the electricity, is inductance. However it is very difficult with the silicon transistors that we use today to make up inductance on the chips. So this is not particularly practical with the present technology.

Another way would be to fill the tank by a supply which stays only a little bit above the level lifting the water supply in time as we fill it up (Fig. 6.3), because then the dropping of water is always small during the entire effort. In the same way, we could use an outlet to lower it but just taking off the top and lowering the tube, so that the heat loss would not appear at the position of the transistor, or would be small; it will depend on how high the distance is between the supply and the surface as we fill it up. This method corresponds to changing the voltage supply with time (Fig. 6.4). So if we would use a time varying voltage supply, we could use this

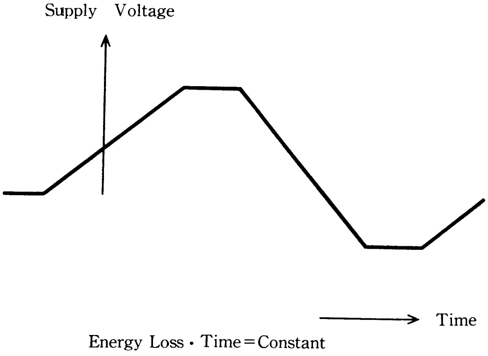


Fig. 6.4

method. Of course, there is energy loss in the voltage supply, but that is all located in one place, that is simple, and there we can make one big inductance. This scheme is called “hot clocking”, because the voltage supply operates also as the clock which times everything. And don’t need an extra clock signal to time the circuits as we do in conventional designs.

Both of these last two devices use less energy if they go slower. If try to move the water supply level too fast, the water in the tube doesn’t keep up with it and I have a big drop. So to work I must go slowly. Again. the U-tube scheme will not work unless that central valve can open and close faster than the time it takes for the water in the U-tube to slip back and forth. So my devices are slower. I’ve saved an energy loss but I’ve made the devices slower. The energy loss multiplied by the time it takes for the circuit to operate is constant. But nevertheless, this turns out to be very practical because the clock time is usually much larger than the circuit time for the transistors, and we can use that to decrease the energy. Also if we went, let us say, three times slower with our calculations, we could use one third the energy over three times the time, which is nine times less power that has to be dissipated. Maybe it is worth it. Maybe by redesigning using parallel computations or other devices, we can spend a little longer than we could do at maximum circuit speed, in order to make a larger machine that is practical and from which we could still get the energy out.

For a transistor, the energy loss multiplied by the time it takes to operate is a product of several factors (Fig. 6.5): (1) the thermal energy proportional to temperature, kT; (2) the length of the transistor between source and drain, divided by the velocity of the electrons inside (the thermal velocity $\sqrt{3kT/m}$); (3) the length of the transistor in units of the mean free path for collisions of electrons in the transistor; and finally (4) the total number of the electrons that are inside the transistor when it operates. All of these numbers come out to tell us that the energy used in the transistor today is somewhere around a billion or ten billions or more times the thermal energy kT. When it switches we use that much energy. It is very large amount of

ENERGY · TIME FOR TRANSISTOR

$$= kT \cdot \frac{\text{LENGTH}}{\text{THERMAL VELOCITY}} \cdot \frac{\text{LENGTH}}{\text{MEAN FREE PATH}} \cdot \text{NUMBER OF ELECTRONS.}$$

Energy $\sim 10^{9-11}\, kT$.

$\therefore$ DECREASE SIZE. · FASTER

LESS ENERGY

Fig. 6.5

energy. It is obviously a good idea to decrease the size of the transistor. We decrease the length between source and drain, and we can decrease the number of the electrons, and use much less energy. It also turns out that a smaller transistor is much faster, because the electrons can cross it and make their decisions to switch faster. For every reason, it is a good idea to make the transistor smaller, and everybody is always trying to do that.

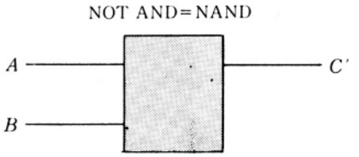
But suppose we come to a circumstance in which the mean free path is longer than the size of the transistor, then we discover that the transistor doesn't work right any more. It does not behave the way we expected. This reminds me, years ago there was something called the sound barrier. Airplanes cannot go faster than the speed of sound because, if you design them normally and try to put them in that speed, the propeller wouldn't work and the wings don't lift and nothing works correctly. Nevertheless, airplanes can go faster than the speed of sound. You just have to know what the right laws are under the right circumstances, and design the device with the correct laws. You cannot expect old designs to work in new circumstances. But new designs can work in new circumstances, and I assert that it is perfectly possible to make transistor systems, that is, that is to say more correctly, switching systems, computing devices in which the dimensions are smaller than the mean free path. I speak of course in principle and I am not speaking about actual manufacture. Therefore, let us discuss what happens if we try to make the devices as small as possible.

3. Reducing the Size

So, my third topic is the size of computing elements and now I speak entirely theoretically. The first thing that you would worry about when things get very small, is Brownian motion; everything is shaking and nothing stays in place, and how can you control the circuits then? And if the circuits did work, it has a chance of accidentally jumping back. But, if we use two volts for the energy of this electric system which is what we ordinarily use, that is eighty times the thermal energy ($kT = 1/40$ volt) and the chance that something jumps backward against 80 times thermal energy is e, the base of the natural logarithm, to minus eighty power, or 10^{-43}. What does that mean? If we had a billion transistors in a computer (which we don't have, we don't have that many at all), working all of them 10^{10} times a second, that is, tenth of a nanosecond switching perpetually, operating for 10^{9} seconds, which is 30 years, the total number of switching operations in that machine is 10^{28} and the chance of one of them going backward is only 10^{-43}; there will be no error produced by thermal oscillations whatsoever in 30 years. If you don't like that, use 2.5 volts and then it gets smaller. Long before that, the real failure will come when a cosmic ray accidentally goes through the transistor, and we don't have to be more perfect than that.

However, much more is in fact possible and I would like to refer you to an article in a most recent Scientific American by Bennett and Landauer. It is possible to make a computer in which each element, each transistor, can go forward and accidentally reverse and still the computer will operate. All the operation in succession in the computer go forward or backward. The computation proceeds for a while this way and then it undoes itself, uncalculates, and then goes forward again and so on. If we just pull it along a little, we can make it go through and finish the calculation by making it just a little bit more likely that it goes forward than backward.

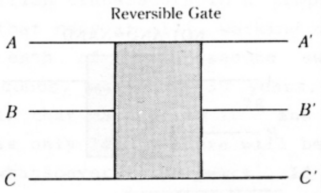
It is known that all the computations can be made by putting together some simple elements like transistors; or, if we be more logically abstract, a thing for instance called NAND gate (NAND means NOT-AND). It has two "wires" in and one out (Fig. 6.6). Forget the NOT first. What is AND? AND is: The output is 1 only if both input wires are 1, otherwise the output is 0. NOT-AND means the opposite. The output wire reads 1 (i.e. has the voltage level corresponding to 1) unless both input wires read 1, if both input wires read 1 then the output wire reads 0 (i.e. has the voltage level corresponding to 0). Here is a little table of inputs and outputs. A and B are inputs and C is the output. Unless A and B are both 1, the output is 1 otherwise 0. But such a device is irreversible. Information is lost. If I only know the output, I cannot recover the input. The device can't be expected to flip forward and then come back and compute correctly anymore. Because if we know for instance that the output is now 1, we don't know whether it came from A=0, B=1 or A=1, B=0 or A=0, B=0 and it cannot go back. Such a device is an irreversible gate. The great discovery of Bennett and, independently, of Fredkin is that it is possible to do computation with a different kind of fundamental gate unit, a reversible gate unit. I have illustrated their idea — with this unit which I could call a reversible NAND or whatever. It has three inputs and three outputs (Fig. 6.7). Of the outputs, two, A′ and B′, are the same as two of the inputs, A and B, but the third input works this way: C′ is the same as C unless A and B are both 1. Then it changes whatever C is.



A	B	C′
0	0	1
0	1	1
1	0	1
1	1	0

Not Reversible
Information Lost

Fig. 6.6



$A' = A$

$B' = B$

$C' = C,$ unless $A = 1$ and $B = 1$

$C' = 1 - C$

$= \text{NOT } C,$ if $A = 1$ and $B = 1$

No Information Lost

Some Force Needed to Push Calculation
Predominantly Forward
Energy Loss · Time Used = Constant

Fig. 6.7

For instance, if C is 1 it is changed to 0, if C is 0 it is changed to 1 only if both A and B are 1. If you put two in succession, you see A and B will go through, and if C is not changed in both it stays the same or if C is changed twice it stays the same. So this gate reverses itself. No information has been lost. It is possible to discover what went in if you know what went out.

A device made entirely with such gates will make calculations if everything moves forward, but if things go back and forth for a while and then eventually go forward enough it still operates correctly. If the things flip back and then go forward later it is still all right. It's very much the same as a particle in a gas which is bombarded by the atoms around it, usually goes nowhere, but with just a little pull, a little prejudice that makes a chance to move one way a little higher than the other way, the thing will slowly drift forward and reach from one end to the other, in spite of the Brownian motion that is made. So our computer will compute provided we apply a force of drift to pull the thing more likely across the calculation. Although it is not doing the calculation in a smooth way, but calculating like this, forward and backward, it eventually finishes the job, As with the particle in the gas, if we pull it very slightly, we lose very little energy, but it takes a long time to get to one side from the other. If we are in a hurry, and we pull hard, then we loose a lot of energy. And the same with this computer. If we are patient and go slowly, we can make the computer operate with practically no energy, even less than kT per step, any amount as small as you like if you have enough time. But if you are in a hurry, you must dissipate energy, and again it's true that the energy lost to pull the calculation forward to complete it multiplied by the time you are allowed to make the calculation is a constant.

With these possibilities how small can we make a computer? How big must a number be? We all know we can write numbers in base 2 as strings of “bits” each a one or a zero. But how small can I write? Surely only one atom is needed to be in one state or another to determine if it represents a one or a zero. And the next atom could be a one or a zero, so a little string of atoms are enough to hold a number, one atom for each bit. (Actually since an atom can be in more states than just two we could use even fewer atoms, but enough is little enough!)

So now for intellectual entertainment we consider whether we could make a computer in which the bits writing is of atomic size, in which a bit is for example whether the spin in the atom is up for 1 or down for 0. And then our transistor changing the bits in different places would correspond to some interaction between some atoms, which will change their states. The simplest would be a kind of 3-atom interaction to be the fundamental element or gate in such a device. But again, it won’t work right if we design it with the laws appropriate for large objects. We must use the new laws of physics, quantum mechanical laws, the laws that they are appropriate to atomic motion. And so we have to ask whether the principles of quantum mechanics permit an arrangements of atoms so small in number as a few times the number of gates in a computer that could still be put together and operate as a computer. This has been studied in principle, and such an arrangement has been found. The laws of quantum mechanics are reversible and therefore we must use the invention of reversible gates, that principle, that idea of Bennett and Fredkin, but we know that’s alright now. When the quantum mechanical situation is studied it is found that quantum mechanics adds no further limitations to anything that Mr. Bennet has said from thermodynamic considerations. Of course there is a limitation, the practical limitation anyway, that the bits must be of the size of an atom and a transistor 3 or 4 atoms; the quantum mechanical gate I used has 3 atoms. (I would not try to write my bits on to nuclei, I’ll wait till the technological development reaches the atoms before I need to go any further!) That leads us just with (a) the limitations in size to the size of atoms, (b) the energy requirements depending on the time as worked out by Bennett, (c) and the feature that I did not mention concerning the speed of light; we can’t send the signals any faster than the speed of light. Those are the only physical limitations that I know on computers.

If we make an atomic size computer, somehow, it would mean that the dimension, the linear dimension is a thousand to ten thousands times smaller than those very tiny chips that we have now. It means that the volume of the computer is 100 billionth, 10^{-11} of the present volume, because the transistor is that much smaller 10^{-11} , than the transistors that we make today. The energy requirement for a single switch is also about eleven orders of magnitude smaller than the energy required to switch the transistor today, and the time to make the transitions will be at least ten thousands times faster per step of calculation. So there is plenty of room for improvement in the computer and I leave you, practical people who work on computers, this as an aim to get to. I underestimated how long it would take for Mr. Ezawa to translate what I said, and I have no more to say that I have prepared for today. Thank you!

I will answer questions if you’d like.

4. Questions and Answers

Q: You mentioned that one bit of information can be stored in one atom, and I wonder if you can store the same amount of information in one quark.

A: Yes. But we don't have control of the quarks and that becomes a really impractical way to deal with things. You might think that what I am talking about is impractical, but I don't believe so. When I am talking about atoms, I believe that someday we will be able to handle and control them individually. There would be so much energy involved in the quark interactions it would be very dangerous to handle because of the radioactivity and so on. But the atomic energies that I am talking about are very familiar to us in chemical energies, electrical energies, and those, that I am speaking of, are numbers that are within the realm of reality, I believe, however absurd it may seem at the moment.

Q: You said that the smaller the computing element is the better. But, I think equipments have to be larger, because....

A: You mean that your finger is too big to push the buttons? Is that what you mean?

Q: Yes, it is.

A: Of course, you are right. I am talking about internal computers perhaps for robots or other devices. The input and output is something that I didn't discuss, whether the input comes from looking at pictures, hearing voices, or buttons being pushed. I am discussing how the computation is done in principle, and not what form the output should take. It is certainly true that the input and the output cannot be reduced in most cases effectively beyond human dimension. It is already too difficult to push the buttons on some of the computers with our big fingers. But with elaborate computing problems that take hours and hours, they could be done very rapidly on the very small machines with low energy consumption. That's the kind of machine I was thinking of. Not the simple applications of adding two numbers but the elaborate calculations.

Q: I would like to know your method to transform the information from one atomic scale element to another atomic scale element. If you will use a quantum mechanical or natural interaction between the two elements then such a device will become very close to Nature itself. For example, if we make a computer simulation, a Monte Carlo simulation of a magnet to study critical phenomena, then your atomic scale computer will be very close to the magnet itself. What are your thoughts about that?

A:. Yes. All things that we make are Nature. We arrange it in a way to suit our purpose, to make a calculation for a purpose. In a magnet there is some kind of relation, if you wish, there are some kind of computations going on just like there is in the solar system in a way of thinking. But, that might not be the calculation we want to make at the moment. What we need to make is a device for which we can change the programs and let it compute the problem that we want to solve, not just its own magnet problem that it likes to solve for itself. I can't use the solar system for a computer unless it just happens that the problem that someone gave me was to find the motion of the planets, in which case all I have to do is to watch.

There was an amusing article as a joke. Far in the future the "article" appears discussing a new method of making aerodynamical calculations: Instead of using the elaborate computers of the day, the author invents a simple device to blow air past the wing. (He reinvents the wind tunnel.)

Q: I have recently read in an newspaper article that operations of the nerve system in a brain are much slower than present day computers and the unit in the nerve system is much smaller. Do you think that the computers you have talked about today have something in common with the nerve system in the brain?

A: There is an analogy between the brain and the computer in that there are apparently elements that can switch under the control of others. Nerve impulses controlling or exciting other nerves, in a way that often depends upon whether more than one impulse comes in; something like an AND or its generalization. The amount of energy used in the brain cell for one of these transitions? I don't know the number. The time it takes to make a switching in the brain is very much longer than it is in our computers even today, never mind the fancy business of some atomic computer. But, interconnection system is much more elaborate. Each nerve is connected to thousand other nerves, whereas we connect transistors to two or three others.

Some people look at the activity of the brain in action and see that in many respects it surpasses the computer of today, and in many other respects the computer surpasses ourselves. This inspires people to design machines that can do more. What often happens is that an engineer makes up how the brain works in his opinion, and then designs a machine that behaves that way. This new machine may in fact work very well. But, I must warn you that that does not tell us anything about how the brain actually works, nor is it necessary to ever really know that in order to make a computer very capable. It is not necessary to understand the way birds flap their wings and how the feathers are designed in order to make a flying machine. It is not necessary to understand the lever system in the legs of a cheetah, that is an animal that runs fast, in order to make an automobile with wheels that goes very fast. It is therefore not necessary to imitate the behavior of Nature in detail in order to engineer a device which can in many respects surpass Nature's abilities.

It is an interesting subject and I like to talk about it. Your brain is very weak compared to a computer. I will give you a series of numbers, one, three, seven, oh yes, ichi, san, shichi, san, ni, go, ni, go, ichi, hachi, ichi, ni, ku, san, go. I want you to repeat them back. But, a computer can take ten thousands numbers and give me them back in reverse every other one, or sum them or lots of things that we cannot do. On the other hand, if I look at a face, in a glance I can tell you who it is if I know that person, or that I don't know that person. But, we do not know how to make a computer system so that if we give it a pattern of a face it can tell us who he is, even if it has seen many faces and you try to teach it. We do not know how to make computers do that, yet.

Another interesting example is chess playing machines. It is quite a surprise that we can make machines that play chess better than almost everybody in the room. But, they do it by trying many many possibilities. If he moves here, then I could move here and he can move there and so forth. They look at each alternative and choose the best. Now, millions of alternatives are looked at. But, a master chess player, a human, does it differently. He recognizes patterns. He looks at only thirty or forty positions before deciding what move to make. Therefore, although the rules are simpler in Go, machines that play Go are not very good, because in each position there are too many possibilities to move and there are too many things to check and the machines cannot look deeply. Therefore the problem of recognizing patterns and what to do under the circumstances is the thing that the computer engineers (they like to call themselves computer scientists) still find very difficult, and it is certainly one of the important things for future computers, perhaps more important than the things I spoke about. Make a machine to play Go effectively.

Q: I think that any method of computation would not be fruitful unless it would give a kind of provision on how to compose such devices or programs. I thought the Fredkin paper on conservative logic was very intriguing, but once I came to think of making a simple program using such devices I came to a halt because thinking out such a program is far more complex than the program itself. I think we could easily get into a kind of infinite regression because the process of making out a certain program would be much more complex than the program itself and in trying to automate the process the automating program would be more complex and so on. Especially in this case where the program is hard wired rather than being separated as a software, I think it is fundamental to think of the ways of composition.

A: We have some different experiences. There is no infinite regression; it stops at a certain level of complexity. The machine that Fredkin ultimately is talking about and the one that I was talking about in the quantum mechanical case are both universal commuters in the sense that they can be programmed to do various jobs; this is not a hard-wired program; they are no more hard-wired than an ordinary computer that you can put information in, that the program is a part of the input, and the machine does the problem that it is assigned to do. It is hard-wired but it is universal like an ordinary computer. These things are very uncertain but I found a minimum. If you have a program written for an irreversible machine, the ordinary program, then I can convert it to a reversible machine program by a direct translation scheme, which is very inefficient and uses many more steps. Then in real situations, the number of steps can be much less. But at least I know that I can take a program with 2 steps where it is irreversible, convert it to 3^n steps of a reversible machine. That is many more steps. I did it very inefficiently; I did not try to find the minimum. Just a way. I don't really think that we'll find this regression that you speak of, but you might be right. I am uncertain.

Q: Won't we be sacrificing many of the merits we were expecting of such devices, because those reversible machines run so slow? I am very pessimistic about this point.

A: They run slower, but they are very much smaller. I don't make it reversible unless I need to. There is no point in making the machine reversible unless you are trying very hard to decrease the energy enormously, rather ridiculously, because with only 80 times kT the irreversible machine functions perfectly. That 80 is much less than the present day 10^9 or 10^{10}, so I have at least 10^7 improvement in energy to make, and can still do it with irreversible machines! That's true. That's the right way to go, for the present. I entertain myself intellectually for fun, to ask how far could we go in principle, not in practice, and then I discover that I can go to a fraction of a kT of energy and make the machines microscopic, atomically microscopic. But to do so, I must use the reversible physical laws. Irreversibility comes because the heat is spread over a large number of atoms and can't be gathered back again. When I make the machine very small, unless I allow a cooling element which is lots of atoms, I have to work reversibly. In practice there probably will never come a time when we will be unwilling to tie a little computer to a big piece of lead which contains 10^{10} atoms (which is still very small indeed), making it effectively irreversible. Therefore I agree with you that in practice, for a very long time and perhaps forever, we will use irreversible gates. On the other hand it is a part of the adventure of science to try to find a limitations in all directions and to stretch a human imagination as far as possible everywhere. Although at every stage it has looked as if such an activity was absurd and useless, it often turns out at least not to be useless.

Q: Are there any limitations from the uncertainty principle? Are there any fundamental limitations on the energy and the clock time in your reversible machine scheme?

A: That was my exact point. There is no further limitation due to quantum mechanics. One must distinguish carefully between the energy lost or consumed irreversibly, the heat generated in the operation of the machine, and the energy content of the moving parts which might be extracted again. There is a relationship between the time and the energy which might be extracted again. But that energy which can be extracted again is not of any importance or concern. It would be like asking whether we should add the mc^2, rest energy, of all the atoms which are in the device. I only speak of the energy lost times the time, and then there is no limitation. However it is true that if you want to make a calculation at a certain extremely high speed, you have to supply to the machine parts which move fast and have energy but that energy is not necessarily lost at each step of the calculation; it coasts through by inertia.

A (to no Q): Could I just say with regard to the question of useless ideas? I'd like to add one more. I waited, if you would ask me, but you didn't. So I answer it anyway. How would we make a machine of such small dimension where we have to put the atoms in special places? Today we have no machinery with moving parts whose dimension is extremely small or atomic or hundreds of atoms even, but there is no physical limitation in that direction either. And there is no reason why, when we lay down the silicon even today, the pieces cannot be made into little islands so that they are movable. And we could arrange small jets so we could squirt the different chemicals on certain locations. We can make machinery which is extremely small. Such machinery will be easy to control by the same kind of computer circuits that we make. Ultimately, for fun again and intellectual pleasure, we could imagine machines tiny like few microns across with wheels and cables all interconnected by wires, silicon connections, so that the thing as a whole, a very large device, moves not like the awkward motion of our present stiff machines but in a smooth way of the neck of a swan, which after all is a lot of little machines, the cells all interconnected and all controlled in a smooth way. Why can't we do that ourselves?

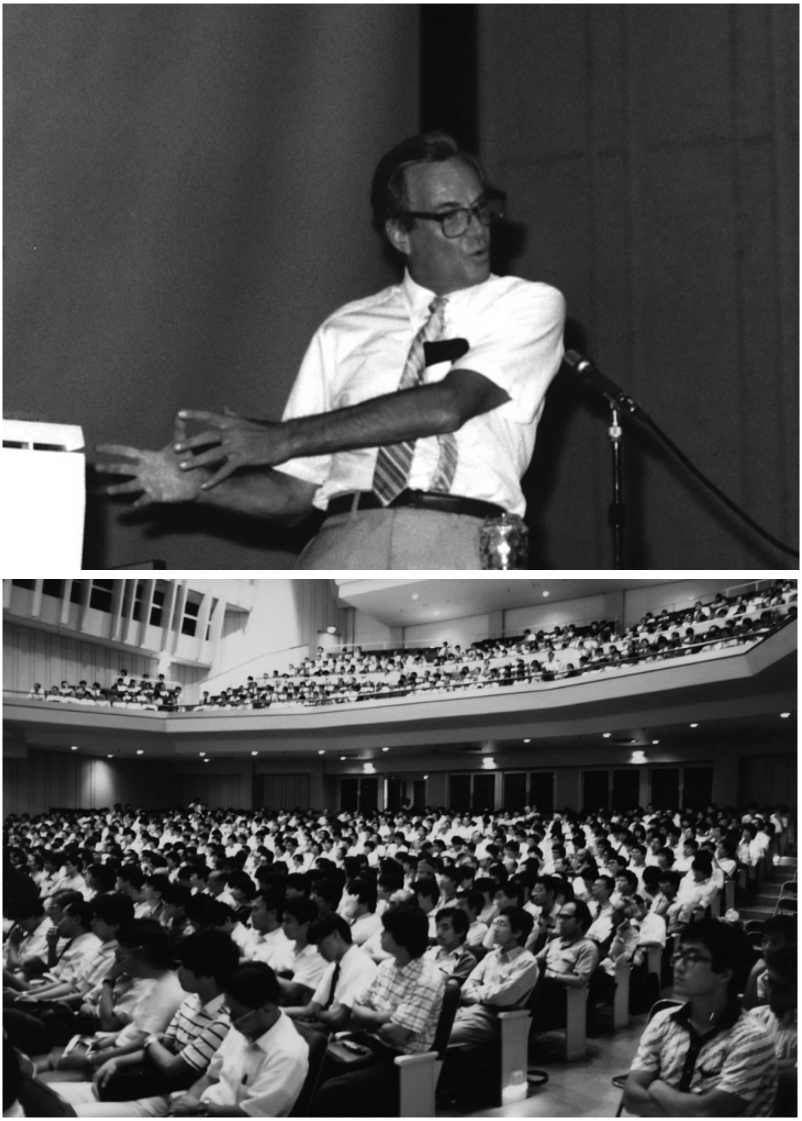


Fig. 6.8 Scenes of the lecture of Professor Richard Feynman at Gakushuin University in Tokyo in 1985