

On the mathematical architecture of a large fractal molecular RAM memory unit.

Gerard 't Hooft

Institute for Theoretical Physics
Utrecht University
and

Spinoza Institute
Postbox 80.195
3508 TD Utrecht, the Netherlands

e-mail: g.thooft@uu.nl

internet: <http://www.phys.uu.nl/~thooft/>

Abstract

If a limited number of chemical building blocks can be produced that can process and transport information, systematic chemical procedures can be devised for manufacturing large fractal computer memories. Repetitive chemical procedures are described that can lead to fractal shaped 3 dimensional structures holding perhaps up to 10^{18} bytes (one attobyte) of information that can be randomly accessed with little energy dissipation.

1. Introduction

A gram of matter contains approximately $6 \cdot 10^{23}$ (Avogadro's number) nucleons, or about $5 \cdot 10^{22}$ atoms if their average atomic weight is 12. Imagine a crystal containing a mixture of two chemically equivalent atomic elements that can be replaced arbitrarily. It would contain about $5 \cdot 10^{21}$ bytes of information. Such a molecular memory unit would dwarf anything existing today.

There are three reasons why it will be practically impossible to actually produce and operate such large and compact memory units. One is the problem of reading and writing information: we forgot the wiring. The second reason is that there are *thermal* and *quantum* fluctuations that would destabilize our memory. Thermal fluctuations occur spontaneously if E/kT is not large enough, where E is the energy needed for a single cell to modify its binary contents, k is Boltzmann's constant and T is the absolute temperature. The probability of such a transition is

$$P_{\text{Therm}} = e^{-E/kT} \mathcal{T}/\delta t, \quad (1.1)$$

where \mathcal{T} is the storage time and δt the time scale of a single chemical transition, which could be of the order of 10^{-14} seconds. To ensure that no spontaneous memory flips take place in the entire unit over its life span of the order of 10 years, we must have approximately

$$E > 95 kT \approx 2.8 eV. \quad (1.2)$$

Quantum fluctuations take place by tunneling through a barrier. The approximate tunneling probability is

$$P_{\text{quant}} = e^{-2\delta S/\hbar} \mathcal{T}/\delta t, \quad (1.3)$$

where δS is the total action for the transition between two classically allowed spots A and B of a potential curve $V(x)$,

$$\delta S = \int_A^B \sqrt{2m(V(x) - E)} dx, \quad (1.4)$$

E is the classical energy, and if this is an atom that flips to a new position, then m is the (reduced) mass of that atom. The classical action δS of a spontaneous memory flip must be at least $\approx 50\hbar$

A third problem with the huge molecular memories is that there simply may not be enough time to fill them up at all. If we imagine a computer to store information at a rate of one gigabyte per second, it would take over 30 years to use up 10^{18} bytes without erasing anything.

We imagine that, due to the energies and action values required, a single digital memory cell may take up at least 100, probably something like 1000 atoms. As we will see, the total amount of wiring may require not more than another few thousands of atoms

or so per memory cell. These estimates suggest that, for the kind of memory units that will be discussed now, 10^{18} bytes per gram is probably the upper limit. If we can limit the energy usage for filling and reading out a memory cell to stay below approximately 1000 eV (most of which will be energy dissipated in the wiring), we see that the total energy used is comparable to the total chemical energy of the unit. If this is spread over a life span of 30 years, clearly cells of this sort will not require any cooling.

This paper is about the properties required for the chemical molecules, the architecture of the construction and the chemical procedures, although we cannot go further into any of the chemical details. There are three, or possibly four, classes of objects to be constructed:

- 1) The actual memory cells, which must contain an element, typically a group of atoms, that can flip from one position into another, a writing unit allowing an incoming signal to produce the flip, and a reading device, emitting on demand a signal about its contents. In view of the fact that many of the memory cells will be written in not more than once, we could in principle consider to limit ourselves to read-only memory cells, but RAM memory will often be preferred. These molecules (or nano-particles) will be called M_1 particles. Binding several of these together, larger memory units, M_2, M_4, \dots are obtained. Memory cells of type M_n can hold n bits of data.
- 2) We need wiring. One-dimensional chemical structures need to be designed that can transmit a signal. The first kind of signal that comes to mind is an electric current, which may involve just a few electrons. Chains of alternating single and double chemical bonds could do the job. Besides this, there are alternative ways to transmit a signal. One can think of the transport of ions, as indeed happens in biological systems, but we have to keep in mind that ions may be rather slow. A third possibility is the use of spintronics or similar advanced techniques that are studied today in nanostructures. Most importantly, we need to be able to transmit such signals over various ranges. Often, a wire needs to be able to transmit several types of signals, so we have wires indicated as W_1, W_2, \dots . Wires of type W_n can transmit n distinct pulses of data, as if they consist of bundles of n primary wires W_1 .
- 3) We need switches. Upon a given signal, a switch will be made that re-routes future signals from one line to another. These are also memory cells, since the position of the switch will be retained over some time. The duration of this memory needs to be short compared to the information keeping cells, so perhaps the switches can be kept smaller. On the other hand, the switches will have considerably more complicated tasks than the memory cells, and for this reason we expect them actually to be bigger. Depending on some details as will be discussed, we have several types of switches, to be referred to as T_0, T_1, \dots .
- 4) Finally, a signal might have to be amplified. An amplifier cell A will be closely related to a switch cell. Actually, when larger scale units, of the order of microns, are involved, one may have to use slightly different signal carriers than the truly

molecular ones near the final memory cells. The need for amplification will depend on details of the chemical substances that can be made, and we shall not spend much time on these.

Graphically, a memory unit will look as in Fig. 1. It is configured as a fractal. Our procedure will always be of the repetitive kind: we build larger memory units by joining two smaller ones together, including wiring and switches, see Fig. 1 *b, c*.

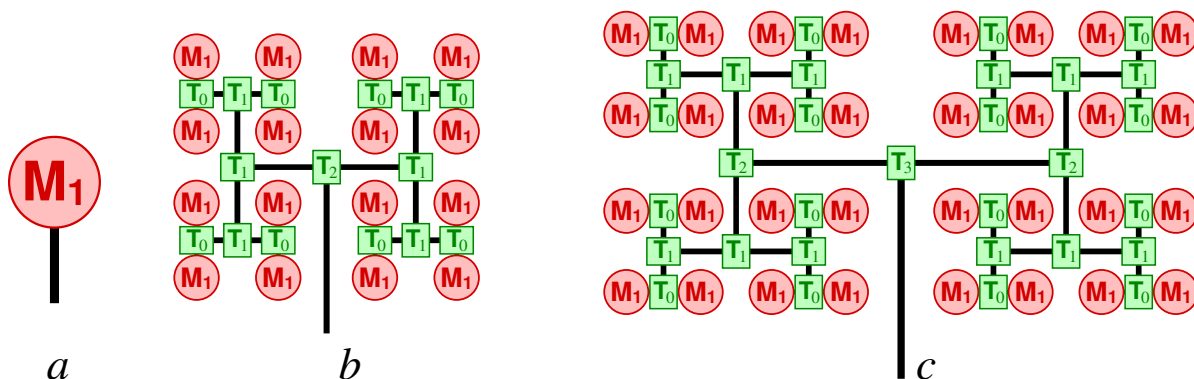


Figure 1: Constructing a simple fractal memory unit. *a)* M_1 is a basic memory molecule. T is a switch molecule. *b)* M_{16} unit. *c)* M_{32} consists of two units of the type M_{16} stitched together. This is a simplified picture; the leads actually consist of several fibres.

Observe that the total number of switches for each memory cell is

$$\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \dots \rightarrow 1, \quad (1.5)$$

so that the total number of T cells equals the total number of memory cells. For the total length of the wiring per memory cell we will have approximately

$$\frac{1}{2} + \frac{2^{1/3}}{4} + \frac{2^{2/3}}{8} + \dots \approx 1.3512 \quad (1.6)$$

times an atomic length scale. Thus we see that the total number of atoms needed per memory cell will stay quite limited.

Since the tiniest units can still be regarded as molecules, we imagine that they will be produced by chemical reactions. Stitching larger units together might be harder, but we think these are technical problems that can be solved. The structures will be allowed to fold in three dimensions, so that the memory occupies the entire volume. As stated, we expect cooling not to become a major problem, but this might depend on the required processing frequency.

We use the word *molecules* rather than *nanoparticles*, even if their sizes are similar; nanoparticles usually are more homogeneous while molecules are very special compositions of atoms. In principle, it will be possible to fabricate memory units where every single

bit of memory has a separate address and can be addressed individually – the ultimate RAM. In practice, however, it will be more convenient to define the notion of *microfiles*, files of a specified length, varying from one byte (8 bits) to several kilobytes or perhaps megabytes, each of which can be stored and read all at once and carries a single address. This can be arranged by making use of slightly different types of switches T_0 , T_1 , etc. In the next sections we specify precisely what the memory cells, the wires and the switch cells must be able to do. Then we show how the memory can be used, and finally how, in principle, it can be fabricated.

We do imagine that the construction described here might be streamlined and improved in other ways, but we wished here to show how the principle works.

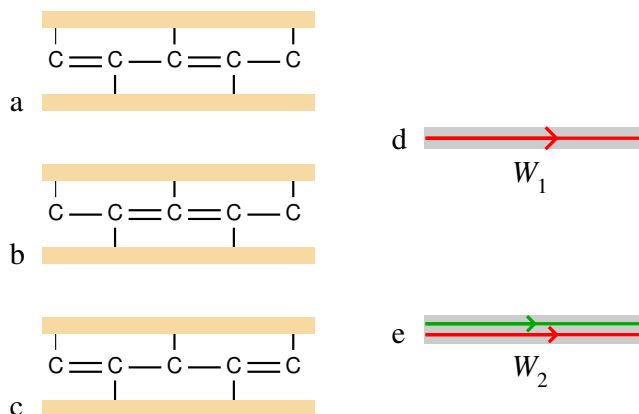


Figure 2: *a*) Wires could consist of alternating double and single chemical bonds. *b*) A surplus electron will generate an extra bond, representing a negative charge, and *c*) holes will correspond to a missing bond. In the next figures, a single strand (*d*) will only be able to transmit one type of pulse (a ‘zero’) in one direction (arrow), *e*) a double strand can transmit both ‘zeros’ and ‘ones’) in one direction. We do not indicate a possible third wire (“ground”) that may be needed to compensate for the charges, like in ordinary household electricity wires. The wires will require some layers of surrounding molecules for isolation and support.

2. The wires

Signals will have to be transmitted. One could think of wireless signals such as infalling and emitted light, but these would require relatively large amounts of energy. In this paper, it will be assumed that signals can be sent through wires. These wires have molecule-sized diameters and the lengths vary from a few nanometers to microns or millimeters. We’ll assume that electrical pulses will be transmitted although there are other possibilities. An electrical pulse can be transmitted along sequences of alternating double and single chemical bonds, as sketched in Fig. 2a. The extra lines represent surplus electrons that may move around sufficiently freely, as in graphite. In Fig. 2b, there is an extra

electron in the middle of the wire, in Fig. 2c, we have a hole (this picture is simplified; in reality, the electron and the hole may occupy several bonds).

A wire that can transmit just one kind of pulse will henceforth be indicated as a single strand (Fig. 2d). In our applications, it will frequently be needed that two different kinds of signals can be transmitted (a ‘0’ or a ‘1’). Wires that can do that are indicated as a double strand (Fig. 2e). Of course, double strand wires may be seen as a pair of wires.

A wild variety of lengths will be needed for the wires. A very long wire (millimeters) may need to be more highly conducting than very short wires, and perhaps amplifiers will be needed for handling signals over greater distances. For simplicity, we ignore the need for amplification for the time being. Manufacturing wires with precisely specified lengths should be possible, and these processes will be discussed briefly.

3. Memory cells

A basic memory cell, indicated as M_1 has to obey the following specifications:

- 1) It carries one bit of information, indicated as $m = 0$ or $m = 1$. Typically, we imagine it to be a switch that can transport an incoming signal either to one strand of a wire or another.
- 2) There is an incoming wire that feeds it data, written as $D_{in} = \nu$, $D_{in} = 0$ or $D_{in} = 1$. Here, ν is the neutral position: no signal is transmitted. If a signal $D_{in} = 0$ comes in, the switch sets m to 0. If $D_{in} = 1$, the memory is set to $m = 1$.
- 3) There is an incoming single strand wire, called C_{in} . It is a ‘control wire’. Normally, it has the neutral position $C_{in} = \nu$. If a signal comes in from here, which we refer to as $C_{in} = 1$, then the memory cell fires, as follows:
- 4) there is an outgoing data wire D_{out} . When the memory cell fires, this means that the signal $D_{out} = m$ is delivered there. After this signal has been given, with a carefully chosen time delay, the control wire C_{out} fires. This latter signal, and the wire over which it is transmitted, may seem to be redundant here, and in other versions of our construction they might indeed be left out, but this feature may nevertheless turn out to be useful, and we keep it for reasons to become evident soon.

The *power* needed for delivering these two signals must come from the $C_{in} = 1$ signal, which therefore has to be sufficiently powerful. Presumably, a sizeable percentage of this power will get lost, so that, later, these outgoing signals may have to be amplified.

Thus we see that the signal wires D_{in} and D_{out} have to be able to transmit two kinds of signals each¹, which is why we indicate them as double strands. The control wires C_{in}

¹In principle, this could be avoided, for instance by varying the duration between pulses. This would generate other technical problems that we decided not to address here.

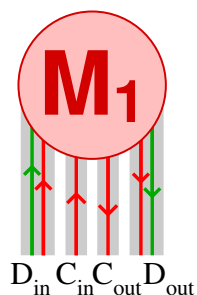


Figure 3: The basic memory cell M_1 requires 2 double strand wires and two single strand wires, as described in the text.

and C_{out} need only to transmit one kind of signal, so they are single wires. The cell is pictured in Fig. 3.

These details are necessary to be able to integrate the basic memory cells into larger configurations, as we shall see.

4. The signal splitters T_0 and T_1 .

The signal splitter is a small computer program, and as such perhaps the most complicated ‘molecule’ to produce along purely chemical routes. Still, this should be possible. The idea of the signal splitter is that it connects two memory units M_p and M_q , where p and q are the sizes, to form one memory unit M_{p+q} of size $p + q$. Usually, we will choose $p = q$. Clearly, this procedure can be repeated to form arbitrarily large units. The signal splitters described in this section will serve to make memories for microfiles (see Section 1); after this, we will design the splitters that control the addresses of the microfiles.

The notation will be as follows. If the entire memory unit is compared with a ‘tree’, the basic memory cells M_1 are the ‘leaves’ of the tree, and the input signals enter at the ‘stem’. All signals that go in the direction from the stem to the leaves are denoted as “in” signals, what goes in the direction from the leaves to the stem are “out” signals. When we talk of D_{in}^s and D_{out}^s , the superscript s refers to the part of the unit whose stem is labeled by s . We begin describing the signal splitter of type T_1 . Here, the sizes p and q are both greater than 1.

The memory cells M_p and M_q will both be connected to our T_1 unit by one double strand data wire inwards each, one double strand data wire outwards each, one double strand control wire in and one single strand control wire out each. At the stem of the T_1 cell, there will also be one double strand data wire going inwards, one outwards, one double strand control wire inwards and one single strand wire outwards. Thus, it looks like Fig. 4 *a*. It must be programmed to do the following. We indicate the leads branching off to the memory unit M_p at one side, with the entry (0), and the leads going to the

other side with the entry (1). The letter T will refer to the leads at the stem.

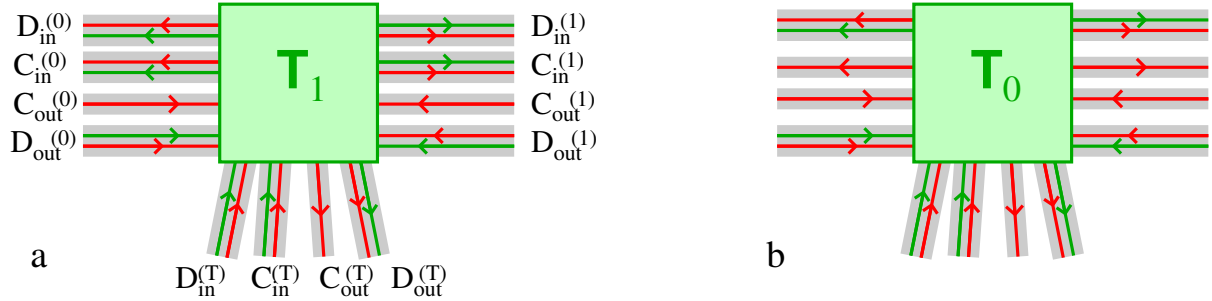


Figure 4: *a*) The signal splitter T_1 requires 3 double strand wires and one single strand wire for all three of its external connections. *b*) The T_0 variety connects to two basic memory cells with two double strands each, and two single-stranded control wires. Its stem is as in T_1 . T_0 can be regarded as a T_1 cell with a slight simplification in its branches.

- 1) Like the basic memory cell, the T_1 cell is also a memory, equipped with a similar switch t . It can be in the position $t = 0$ or $t = 1$.
- 2) if a signal $C_{\text{in}}^T = 0$ is received, the memory in the T cell is first switched to $t = 0$.
- 3) The ingoing control signal C_{in}^T and the ingoing data wire D_{in}^T will be connected to the corresponding external ingoing wires to which the memory t is pointing. The other external ingoing wires stay disconnected. Thus we have:

$$D_{\text{in}}^{(t)} = D_{\text{in}}^T, \quad C_{\text{in}}^{(t)} = C_{\text{in}}^T, \quad t = 0 \text{ or } 1. \quad (4.1)$$

$$D_{\text{in}}^{(1-t)} = C_{\text{in}}^{(1-t)} = \nu. \quad (4.2)$$

The signal $C_{\text{in}} = 0$ is allowed to travel all the way to the first leaf of the tree. If it accidentally would hit any of the other leaves this will have no effect, but we cannot allow the data signals to reach the wrong branches.

- 4) The outgoing data wire D_{out}^T transmits the signal it may receive from $D_{\text{out}}^{(t)}$. No harm is done if it would also transmit data from $D_{\text{out}}^{(1-t)}$, as we won't expect data to come from there.
- 5) If the T_1 cell receives a signal from the control wire $C_{\text{out}}^{(t)}$, then: if $t = 1$ it sends the signal through its own outgoing control wire. Otherwise, it switches to $t = 1$ and sends out a signal $C_{\text{in}}^{(1)} = 0$, that is:

$$\text{If } C_{\text{out}}^{(t)} = 1 \quad \text{then if } t = 1 \quad \text{then } C_{\text{out}}^T = 1; \quad (4.3)$$

$$\text{else } t = 1, \quad \text{and } C_{\text{in}}^{(1)} = 0. \quad (4.4)$$

All signals last for just enough time to transmit the required information; they subsequently return to neutral. Again, the power needed for these signals must come from the signal on the ingoing control wires at the stem. If too much loss of power takes place, further amplification will be needed. Further down the stem, the signal on the ingoing control wire could be used for this.

The signal splitter T_0 will be used when the two cells it connects to are basic memory cells M_1 . It acts the same way as the T_1 cell, except that the signal $C_{in}^T = 0$ is not transmitted any further. If a signal $C_{in}^{(t)} = 1$ is fired to the memory cell $M^{(t)}$, or if its data wire $D_{in}^{(t)}$ receives a signal, then after a short time interval, this acts also as if $C_{out}^{(t)} = 1$, so step 5 is always carried out in this case. The time interval must last just long enough to allow the input pulses to die out.

In Fig. 1 we see illustrated how the signal splitters T_1 and T_0 are used to produce a memory for two microfiles of one byte (8 bits) long (a) or 4 such microfiles (b). To write data in a microfile, the first T_1 splitter is fed with a 0 signal in its C_{in}^T wire. Subsequently, one bit (a 0 or a 1) is fed in the data wire D_{in}^T . We see that this is transported into the first memory cell. After this, we continue sending a 1 into the control wire C_{in}^T , followed by the next bit into D_{in}^T , until all bits are delivered. Note that the previous contents of the memory cells first emerges in the D_{out}^T wire. After the last bit of data emerged at the stem, and “end of file” signal $C_{out}^T = 0$ is received. It is this signal that enables us to combine small units into larger ones, repetitively, so that the shape of a fractal emerges.

Reading the memory goes the same way. If we do not write anything in the D_{in}^T wire, we just receive the next bit after every signal in the C_{in}^T wire. Timing is important; holding the control pulses too long would generate an uncontrolled, fast sequence of memory data.

Thus we have a memory unit whose number of bits is a power of 2 long, and the entire memory contents of this unit is written and read sequentially. It serves as a microfile, to be written and read all at once. Next, we will assemble the microfiles such that each of these can be addressed in any order for writing and reading, using an address code.

In Fig. 5 the optimal shape is shown of the T cells in three dimensions. This way, they will take a minimal amount of space when arranged in a continued, fractal configuration. Actually, this is a detail for later concern when entire fractal structures are being designed; we then wish to lead the wires through narrow spaces. At every step in the construction of a fractal the lengths of the wires will have to be longer, so the actual lengths of the wires are not displayed in Figs. 4 a, b .

5. The signal splitters T_2 and T_3 .

We now need signal splitters that allow for random access reading and writing of the data, which themselves are packaged in the form of microfiles. We will call these the RAM splitters. The microfiles are now assumed to be series of 2^m bits, to be steered in the right direction by the control wires. Suppose that the RAM splitters T_3 work as follows:

- 1) If a signal $C_{\text{in}}^T = 0$ is received, then the next signal $D_{\text{in}}^T = d$, forces the switch to go to $t = d$. Then a signal $C_{\text{in}}^t = 0$ is fired.
- 2) If a signal $C_{\text{in}}^T = 1$ is received, it is transferred to the daughter cell: $C_{\text{in}}^t = 1$. The same happens to all incoming signals $D_{\text{in}}^T = d$, which are passed on: $D_{\text{in}}^t = d$.
- 3) Any signal from C_{out}^t is passed on to C_{out}^T . All signals D_{out}^t are also passed on to D_{out}^T . Here also, it should not matter what signals come from the other daughter cell since it is expected to stay quiet.

The last RAM splitter, called T_2 , may be programmed just as the T_3 splitter described above, with one further function: after making its switch to $t = d$, as above, it also emits a signal $C_{\text{out}}^T = 1$.

We now expect the entire fractal to work as follows. Let us denote ingoing signals $C_{\text{in}} = c$ as $(c \rightarrow C)$, $D_{\text{in}} = d$ as $(d \rightarrow D)$, while the outgoing signals are denoted as $(C \rightarrow c)$ and $(D \rightarrow d)$. We denote the length of the microfiles as m and the number of bits of the addresses as k . The bits in the microfile are d_1, \dots, d_m and the address is denoted as a_1, \dots, a_k . Writing in the memory goes as follows:

$$\begin{aligned}
&(0 \rightarrow C) (a_1 \rightarrow D) (1 \rightarrow C) (a_2 \rightarrow D) \cdots (1 \rightarrow C) (a_k \rightarrow D) \\
&(C \rightarrow 1) \\
&(d_1 \rightarrow D) (1 \rightarrow C) (D \rightarrow d_1) (d_2 \rightarrow D) (1 \rightarrow C) \cdots (D \rightarrow d_m) (C \rightarrow 0) . \quad (5.1)
\end{aligned}$$

Reading goes the same way; we just omit the data input signals $(d_i \rightarrow D)$ in the last line; the outcoming signals then contain the data to be read.

We see that the signals $(C \rightarrow 1)$ and $(C \rightarrow 0)$ act as end-of-file signals. Like the other control signals, they ensure that signals do not interfere. However, one may imagine that, once systems with sufficient stability can be constructed, some of these control signals might be omitted. We were unable to test this program all the way, but such tests of course are possible, and errors can easily be corrected.

6. The three dimensional structure

The three dimensional structure of our fractal memory can be optimized. Assume that a memory M_k , of size k , occupies a box with dimensions (a, b, c) , with $a > b > c$, and assume the thickness of the wires to be d . Then we could combine two of these memories to give a memory M_{2k} , occupying a box of size $(2c + d, a, b)$. If d could be ignored, the volumes V_k would simply scale as k itself, so the memory is volume-filling. We could keep d as small as possible, by choosing the 3-dimensional shape of the T cells as in Figure 5.

As stated in the Introduction, the wires joining the larger memory structures will have to transmit signals over much larger distances, and this might require these to be bulkier, so that the out going signals will have to be amplified. Thus, at some points, amplification

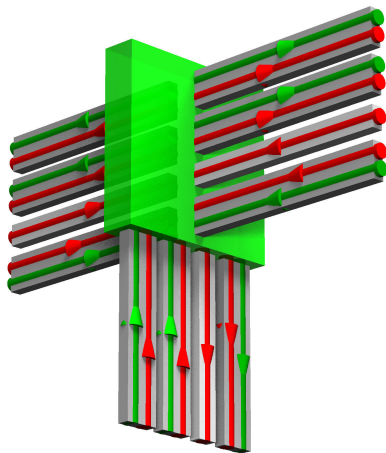


Figure 5: Optimal three dimensional shape of the beam splitters T_0, T_1, \dots . The lengths of the external wires are not drawn to scale.

cells may have to be included. Since their function is quite obvious we will not describe these further. The *ingoing* signals probably will not require any amplification, since they move towards tinier and tinier structures; possible small losses then do no harm.

Producing the M and T molecules, and the wires W , will of course be far from easy. In particular the T 's are little computers by themselves. These will have to be manufactured as such, but the idea is that this is done chemically. Much will now depend on what can be achieved on this front. The philosophy of this paper is that, once we have the desired molecules, it will be possible to join them in the configurations described above. The general idea is explained in the following Section.

7. Chemistry

As a theoretical physicist, the author can only give very rough sketches about the chemical steps that have to be taken. The fundamental point is that we have a limited number of fundamental building blocks, and these will have to be joined according to very tight specifications. Each of our elements can be characterized as blocks with various sizes (a, b, c) (referring to length, width and height), that have to be attached at precisely specified points. In particular, the wires will have to be connected in such a way that the signals can pass through.

Much of the chemistry that we imagine to be applied here may not yet exist. We hope that this paper inspires chemists to study the possibilities further.

Let us begin with the wires. Of course, with “wires” we mean the one-dimensional substances that are supposed to transmit the signals in question. Imagine that we have succeeded in chemically preparing short pieces of wire with lengths L_1 and L_2 , which may possibly be equal. Assume both sets have different end points. This we will always

have to assume, but in the present case it will be important that *all four* end points are different. This should be easy: we prepare the L_1 wires and the L_2 wires separately, even if $L_1 = L_2$.

One end of each of the wires L_1 now has to be prepared to be of type A and one end of wires L_2 of type B . The solutions containing these “wire molecules” are then mixed, after which a chemical reaction must force the A ’s to connect with the B ’s, to obtain a solution with wires of length $L_1 + L_2$. Again, the resulting wires have two different end points.² This is important, because if its two end points were identical, either in the L_1 ’s or the L_2 ’s the final product will always have identical ends. This we wish to avoid. Clearly, the procedure can now be repeated, and wires with a very precisely defined length can be produced.

Now the above will work well particularly for single strand wires. Notice, however, that the wires that we will need to use frequently, have multiple strands, and all of these strands will have to connect correctly. In that case, we will have to see to it that the two outermost wires are given different chemical endings A_1 and A_2 for wire L_1 , and B_1 and B_2 for wire L_2 , requiring chemical reactions that only bind the A_1 ’s with the B_1 ’s and the A_2 ’s with the B_2 ’s. The wires in between the wires at the edges will end up in place anyway, but they still will have to be connected such that signals can be transferred.

The M cells and the T cells must now be connected to the wires. This, we imagine, can be done in very systematic way. First, the memory cells are to be connected to the first T cells, which we called T_0 . The memory cells may be given end points of type A_1 and A_2 each, while the T_0 cells are given end points B_1 and B_2 at each of their two branches (but not yet at the stem). Mixing solutions with molecular numbers in the ratio of 2 : 1 and allowing all A_i to bind with the corresponding B_i should give us the requires 2-bit memory cells. With these, we repeat the procedure a number of times, now using the T_1 cells, until we decided that the dimension of the desired microfiles are reached. Then we continue one step with T_2 and the desired number of steps with T_3 . Of course, the produced “molecules” will grow exponentially in the process, so that later steps may require entirely different chemistry than the earlier ones.³ Still, the number of chemical steps that will have to be made only grows logarithmically with the size of the required final product, so that the procedure may stay economically interesting.

In principle, the final product will reach atomic precision, but quite likely, bugs will appear, when a reaction did not proceed flawlessly. Again, at this point, the control signals that produce end-of-file- warnings may now be useful to allow the surrounding software to handle possible imperfections in the memories.

It is not obvious how far the last steps can go, where bulkier systems are obtained. It could be decided to glue the resulting molecules onto more conventional silicon chips as soon as sizes of a few microns are reached. Much will depend on details of what is chemically possible, as well as unforeseeable economical factors.

²The new length might be slightly different from $L_1 + L_2$ depending on the nature of the chemical reactions.

³In addition, more chemically active points A_3, A_4, \dots and B_3, B_4, \dots may be needed to obtain more robust structures and to make the interactions more efficient.

However, the most important conclusion made here is that for electronic uses it will be possible to reach atomic sizes while exploiting the third dimension. We do conclude that Moore's law can proceed far beyond the present limit.