Northwestern University
ME 385 Nanotechnology

Nanoscale computing and actuators, for potential use in Nanobots

Stefan Bracher

2005

Abstract:

When speaking about Nanotechnology, most people think about Nanobots. The idea of inserting small devices inside our body has fascinated scientists as well as science fiction writers.

But how far are we in the development of these Nanobots? After a short introduction, this paper will present four different types of actuators, that could be used for Nanobots. This is followed by a discussion of how the Nanobots could be controlled. Part four deals with devices to build the necessary control-logic and in the end there will be a conclusion section.

Table of Content

2. Actuators

3. Control

4. Computing in Nanoscale

1.Introduction

What are Nanobots and what can they be used for?

Nanotechnology can be defined as the technology that is "concerned with materials and systems whose structures and components exhibit novel and significantly improved physical, chemical and biological properties, phenomena and processes due to their nanoscale size" [16]. Thus it is a very broad field with no distinct borders, reaching from biology, chemistry and material science to electrical and mechanical engineering.

The “Proteus”, a nanoscale ship from the movie “Fantastic Voyage” [37]

Nevertheless most people think of Nanorobots, or short, Nanobots, when they hear Nanotechnology. But what are Nanorobots and, probably even more interesting, what can they be used for?
The Czech word "robota" stands for "work". Thus a Nanorobot can be seen as nothing else than a nanoscale machine that does some work.

There are many possible applications for Nanobots: Probably one of the oldest ideas is to insert nanoscale devices in our bodies, where they could do observations, deliver chemical substances, repair cells and much more.
There are plans to use nanorobots to make self-repairing suits for astronauts [8]. They could also be used to analyze structures and Integrated Circuits. Multi-purpose devices built of collaborating nanobots, that can change their shape and functionality are imaginable.
Though most of those ideas still are science fiction, some of them are quite close to be realized. According to C. A. Haberzettl [16], there are several companies currently working on nanostructures for drug delivery.

The construction of Nanobots

To build a nanorobot, like for their "normal sized" counterparts, sensors, actuators or manipulators and some sort of control would be needed. The construction itself is also a subject of research, as we can't just put the parts together with our hands. That nanoassebly is not impossible has been proved by some encouraging attempts using scanning probe microscopes. In this paper a few nanoscale actuators shall be presented as well as a discussion of how nanobots could be controlled and nanoscale computation possibilities.
Most of these devices or theories have been developed without the aim of constructing nanobots, but could be used for their implementation.

How will nanobots move?

For most applications it is sufficient for nanobots to be able to float in a fluid, what probably is easier to achieve than walking on a surface with big (microscale) obstacles.
But also floating in fluids can be tricky. A. A. G. Requicha [20] stated in one of his papers that already structures in the microscale range will have a Reinolds number in the range of 10^(-5) (Calculated with typical peed 10 micrometer/s and typical length of 1 micrometer), and thus will be in the low-Reynolds regime, which makes friction the dominant force. He also mentions that it appears that below 600nm no self-propelled organism can be found. Thus for small nanobots, diffusion should also be considered as a way of transportation.

2.Actuators

2.1 Nanotweezer

Carbon Nanotube Nanotweezer [6]

Manipulation of Nanoscale objects is mainly done by pushing the objects with the tip of STM and AFT-microscopes. For nanobots a tool to grasp an object and physically connect them to each other would be very useful. The Nanotweezers developed so far consist of a microscale and a nanoscale part. P. Kim and C. M. Lieber [6] attached carbon nanotubes to the end of two electrodes. The tweezer can thus be opened and closed by electrostatic forces due to an applied voltage.

Nanotweezer tip formed by electron bean angle change. [13]

P. Boggild et al. [13] though state that it might be a disadvantage to have an electric potential on the tip of the Nanotubes. Thus, not the tips themselves but two separated microscale electrode's serve for the actuation of the Nanotweezer produced by this group. Interesting is also the way Boggild et al. made the tips: They focused a electron beam on the end of the electrodes, stimulating the growth of the tips. By variation of the electron beam angle, they could enforce the tip growth direction. This is mainly interesting as this method could also be used to form a pure nanoscale tweezer by detaching the produced structure from the macrosscale part. Other possibilities for Nanotweezer actuation could be thermodynamic effects or a chemical actuation.

2.2 Nested nanotube actuators

The nested nanotube actuator [8] is a linear actuator. The idea is to place a carbon nanotube with a closed end within an other buckytube. By varying the pressure difference a linear motion could be produced.
Though easy in principle, the assembly of such a device might be very difficult.

Nested nanotube actuator [8]

2.3 The molecular ratchet

ATPase

Schematic of ATPase [29]

ATPase is a structure in organisms that produces adenosine triphosphate (ATP), the major energy transport molecule of cells, out of adenosine diphosphate (ADP), when a proton gradient is applied, or, by consuming ATP, drives a proton flow.

The key feature is a spinning shaft. The rotation is surprisingly not produced directly by ATP consumption or the proton gradient but by random Brownian Motion. The exact mechanism used in ATPase to convert this non directional fluctuations in a directional one, lies beyond the scope of this paper and is still subject of research.

Brownian Ratchet Motor Mechanism

The biological motor ATPase, based on Brownian Motion, inspired many research groups to work on a Brownian Ratchet Motor Mechanism. The results of J. V. Hernández, E. R. Kay and D. A. Leigh at the University of Edinburgh [25] shall be presented as an example.

Principle

Molecular Ratchet Principle Step 1 and 2 [25]

Two molecules (blue ring and the orange-purple-red structure) are physically linked, but chemically not bonded. The blue ring will thus remain wherever it is energetically most favorable. In step one, this is the orange side.

If now somehow the purple structure can be changed (into the green structure) so that the energy balance is changed (step two), the blue ring should theoretically change it's position. However the energy provided by the Brownian Motion is not high enough to overcome the energy barrier posed by the red and purple side structures.

Molecular Ratchet Principle Step 3a / 3b [25]

But as soon as one of these structures is removed, the ring can either turn clockwise (the red structure is removed, 3a) or counterclockwise (the purple structure is removed, 3b).

Molecular Ratchet Principle Step 4 and 5 [25]

After the reattachment of the side structures the blue ring is encapsulated in the new minimal energy position over the green part (Step 4).

If now the energy balance is changed back to the initial conditions (As in step1), the orange side becomes again more attractive (step5).

After detachment and reattachment of one of the side structures to remove the energy barrier temporary (6a/6b), the blue structure is back to it's initial position over the orange part.

Molecular Ratchet Principle Step 6a / 6b [25]

By choosing the right sequence of balance braking and side structure detachment/reattachment, either a clockwise or counterclockwise rotation of the blue structure can be obtained.

Chemical implementation

Chemical implementation of a clockwise rotation [25]

The implementation was done with rotaxanes and catenanes molecules.

To obtain a clockwise rotation of the blue part, the following reaction sequence was performed:

1. Photoisomerization of the fum-E-1 isomer to the maleamide (mal-Z-1) (bottom right to bottom left)

2. De-silylation/resilylation to succ-Z-1 (top left)
3. Reisomerization to fumaramide succ-E-1 (top right)
4. De-tritylation/re-tritylation to regenerate fum-E-1

To prove that a rotation of the blue part was achieved, NMR spectroscopy was done at the various steps.

NMR spectroscopy with (bottom image) and without (top image) the blue ring over the orange structure. When residing over the orange part, the spectrum is changed because of shielding effects. [25]

As the blue molecule shields the response of the part over which it resides, it's position can be determined.

Results

The group at University of Edinburgh [25] stated that it takes three components to build a working molecular ratchet mechanism:

1. A randomizing element
2. An energy input (2nd thermodynamic law)
3. Asymmetry in energy or information in the direction of the motion

The randomizing element here was the Brownian Motion providing the movements of the blue ring. The fact that always one side of the molecule is energetically more favorable provides the asymmetry in energy and finally the energy input comes from the chemical reaction performed on the structure.

The work done shows that a molecular motor can be built based on the ratchet principle. However some major problems arise when this motor should be used for a nanorobot:

- The chemical reaction where performed in a solution, but what if the environment content of the robot can't be changed?
- The chemical reaction chain in the experiment above took hours to complete, what is far to slow for an implementation.
- How can the rotation be translated in useful mechanical work?

As seen a lot of work has still to be done. An other approach, instead of producing the motor artificially, might be to simply use the ATPase molecule. It has been proved that when it's exposed to a ATP containing solution, the shaft starts to spin. A group at Cornell University [32] was even able to attach a metal propeller to the ATPase to show it's rotation.

2.4 Enzymes as molecular motors

Schematic of an enzyme motor. ATP hydrolysis causes a small confirmation change, leading to a move of the enzyme.[19]

ATPase is not the only molecular motor present in biological systems. There is a wide variety of enzymes (myosins, dyneins, kinesins) that move along tracks within cells (filaments or microtubules). All of them consume ATP causing small confirmation changes, that mechanically amplified, lead in the end to a directional movement. Theories of how exactly the movement takes place vary from hand over hand and ratchet like to inchworm like movements.
M. Schliwa and G. Woehlke [19] divide this molecular motors in two main subcategories: processive motors, that always remain attached to the track and and non processive motors, that stay on their place and detach after a cycle, allowing collaborative work of several motors on one string. The second category is probably the more interesting one, regarding to possible use in nanobots, as the collaboratively achieved movement of a string could be used by attaching a object to it.

At the University of Portsmouth, UK [23], a special enzyme group, type I R-M enzymes, is studied for the use in a molecular magnetic switch. The special feature of this type is, that it does not move along a track. It bounds to a bounding site on a DNA string and remains bonded there, while retracting the DNA like a fishing rod. This is extremely useful as objects to be moved don't have to be attached to the enzyme itself but to the DNA strings on both sides.

Illustration of type I R-M enzymes retracting DNA like a fishing rod. [23]

There are also groups [24] [15], that instead of simply using the enzymes available from biology, try to copy the principle and build synthetic molecular motors.

3. Control

3.1 Central Control

Like "macroscopic" robots, Nanobots need to be controlled. Because of the small dimensions of the Nanobots, on board computation will be limited. One solution to that problem is to do the computation on a central computer that receives sensor data from the Nanobots and sends them orders.

For example, there are plans to use Nanobots to build a Marsuit that is able to repair itself, when damaged. [8]. The idea is that the Nanobots actually used in the suits will be produced in large thanks by assembly Nanorobots, equipped with pressure sensors to obtain acoustical orders from a central computer.

The advantage of this central control is that there are basically no limits to the computation and the program complexity. In a task where you have a homogeneous group of Nanobots, performing the same task simultaneously like in the Marsuit-Nanobots assemblers, this might be a good solution, as you could just send one order to the whole group. But what if the tasks are not performed simultaneously or you need feedback from the Nanobots? If so you will have to give every Nanobot a unique identification number, what will be a problem, first, as the Nanobot will need on board computation to distinguish whether or not a message is sent for them, and second, as soon as you have a larger group of Nanorobots, you might need thousands of these identifiers.

3.2 Simple control circuits

In many applications complex computing might not be needed. For drug delivery, it could be sufficient for the nanobot to diffuse in the blood circuit until a certain environmental condition (some molecules, ph-value..) is present and then release the drug.

A good example for amazing behavior based on simple circuits are the Braitenberg vehicles [2]:

These vehicles are simple robots consisting of two wheels with independent motors and two light sensors. The more light is shining on a sensor, the more power will be delivered to the motor to which it is connected.

Light attracted Braitenberg vehicle [2], image [20]

Depending on how the connections are made complete different behavior is showed. So will the vehicle on the right always steer towards a light source, while a vehicle with switched sensor-motor connections always steers away from it.

Another example of simple, but successful behavior is the E.coli bacterium [20]. What it does is just running straight on for a certain time and then turning on the spot to a random direction. Chemical sensors sense is the nutrition content of the environment. If the content is increasing, the bacterium will run straight for a longer time, if decreasing for a shorter time. Without ever knowing where the nutrition is, the bacterium finally finds it's way to it.

3.3 Swarming

Termite nest in Sri Lanka [Source: http://de.wikipedia.org/wiki/Benutzer:Sebastianjude]

There is an other, bio-inspired concept, called swarming. The main point about swarming is, that you have relatively simple agents with limited capacities that, in collaboration with other agents, achieve a performance that is much more than the sum of the individual performances.
A termite swarm, for example, can do amazing things, while the single termites are quite simple. And that is exactly what is needed for Nanorobot swarms.

That the concept also works outside nature has first been showed by Craig Reinolds in 1986 [30], who made a bird-swarm animation based on it. Instead of controlling the trajectory of every "boid", how he called the agents, he just set three basic rules for them:

1. Try not to be too close to any other agents or obstacles
2. Steer in the average direction of the agents in your area
3. Try to go to the center of gravity of the agents in your area

Rule 1: Separation [30]

Rule 2: Alignment [30]

Rule 3: Cohesion [30]

The resulting animation showed that central control is not needed and the swarm is "self-organized".
The most amazing fact about ants is that they are able to build complicated structures without having a general plan. This ability is based on a phenomena

called stigmergy. It describes "the influence on behavior of the persisting environmental effects of previous behavior" [7], each action is triggered by the environmental effects of the previous action.

Bio-inspired swarm behavior and stigmergy have already been successfully implemented in several "Macroscopic scale" robot swarms. (Irobot SwarmBot [27], Swarm-bots at EPFL [18], Pherobots [28] and others).
To achieve a good communication inside the swarm (around edges, obstacles, ...), some of them ([28][27]) use the single robots as relays, that resend all incoming transmission. The messages thus dissipate inside the swarm like "Virtual Pheromones".
The tricky part of using swarm behavior is to find rules, that lead to the demanded behavior. One solution to this problem is to use evolutionary engineering techniques, but even then, the outcome is uncertain, apart from the fact that evolutionary techniques in nanoscale might be dangerous.

4. Computing in Nanoscale

I-Swarm, a project at University of Karlsruhe [31] is not yet in the nanoscale range but goes in that direction.

No matter if central control, swarming or an other concept is used to control Nanobots, a certain on-board computation (and if it is only a switch) will always be needed. In the following sites, different computing approaches are presented.

4.1 The universal turing machine

Before we can discus different possibilities of computing, it is essential to know how our modern computers work.
In 1936 A.M Turing [1] presented the concept of an "Universal Computing Machine", today known as the "Universal Turing Machine" (UTM). Most of our

computers still work with that concept.
The Universal Turing Machine consists of a tape (or any other mass storage device like a hard drive, disc etc) and a head that reads and writes data and can move in any direction on the tape. The tape itself is clustered into cells that contain the data.
The head of the machine has an internal state. This state together with the cell content at the actual position determine the next transition (right or left), the change of the state as well as other actions to take.

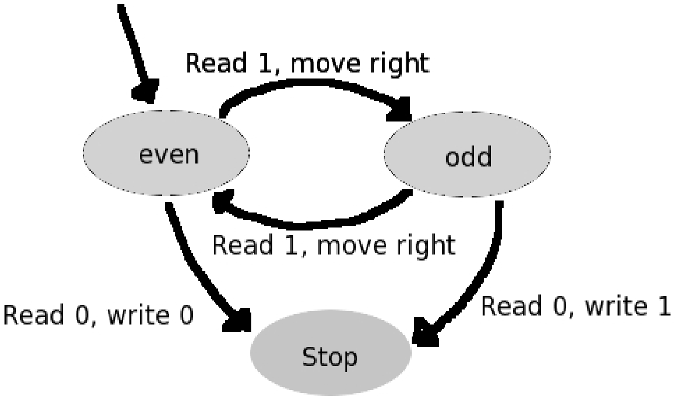


Diagram of the universal turing machine computing if there is an odd or even number of ones on the tape

Example: UTM computing if there is an odd or even number of ones on the tape

The diagram on the left is "the software" of the UTM that tells the machine what to do:
When started the UTM goes directly in the state "even".
If there is a "1" in the first cell of the tape, the head will move right and the new state is "odd".
Again, if a "1" is read, the head will move right and the state is switched.
This continues until a "0" is read, what in this case means "end of tape".
At this point, the UTM outputs the last state and halts. In this example, the output of a "1" stands for odd and "0" for even.

If the inserted tape was "1110" the output tape will be "1111" where the last 1 means that there was an odd number of ones on the input tape.

This is of course a very simple example but, as said before, most of our modern computers are based on this principle.

The most important statement made by Turing [1] is, that if a problem is computable with a particular UTM, it is computable with any UTM. This means that, in principle, the machine described above can do anything a modern computer can do (Tough it may take hundreds of years).
For Nanotechnology this means that if we are able to build a Universal Turing Machine, no matter how simple it is, we will be able to compute whatever we can compute with our very complex PCs.

4.2 Electro-Mechanical

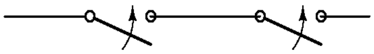
The head of a Turing machine has to be able to check the transition rules imposed by the "software". The rules of the UTM computing the odd or even number of ones on a tape can be translated in:

1. At start go to state "even"
2. If the state is "even" and you read "1", switch state and move right
3. If the state is "even" and you read "0", write "0" and stop
4. If the state is "odd" and you read "1", switch state and move right
5. If the state is "odd" and you read "0", write "1" and stop

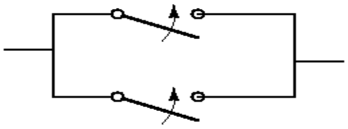
working principle of a relay: A mechanical switch is opened and closed by turning on and off the current in a magnetic coil. [www.wikipedia.org - Norman G.]

The implementation of these rules can be done in many ways. Early computers worked on purely mechanical basis or, later on, with vacuum tubes and relays that where used as amplifiers and switches.

Switches can be used to build all kind of logical elements. Two serial switches form for example an AND-Gate, as current can only flow through when both of them are closed, where two switches in parallel form an OR-Gate, as current can flow as soon as one of them is closed.

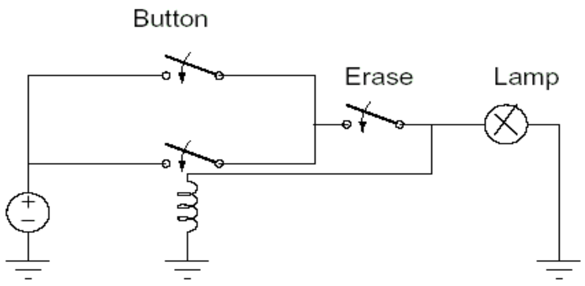


AND Gate



OR Gate

You can even build a storage, that could also be used as the "tape" of an UTM, just with switches and a relays:

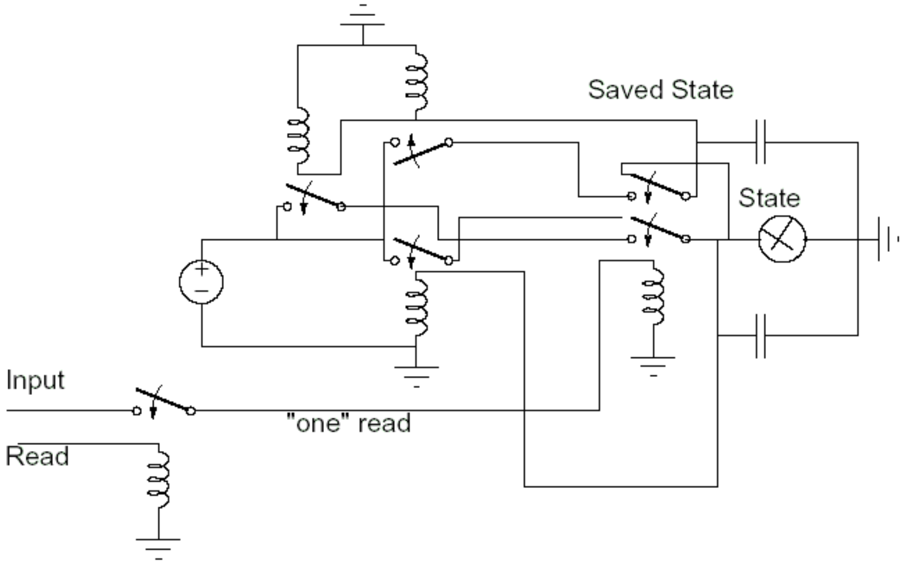


Event storage

If the "Erase" switch is closed, the circuitry on the left will store the event of closing the button. As soon as the button is closed, the light goes on and the relays closes a bypass to the switch so that even if released, the lamp remains on. As soon as the "Erase" switch is opened, the lamp goes off.

The "odd-even" machine could be build in the following way:

A paper tape with holes, representing "1" could be moved trough the machine slowly. Whenever the head is positioned above an information field, the read voltage is turned on (This could be done my marking the information field with an additional lane of holes.
A "1" (hole) will give a full voltage on the head input, while a "0" (no hole) gives zero input.
The head has two inputs, an "Input" port and a "Read" port.



Head of the odd-even machine

A possible logic of the head is shown on page 17. When "Input" and "Read" are on simultaneously , a "one" has been read.
As long as there is no "one" read, there will be no current on the lamp indicating the current state, what means we have an "even" number of ones.
When the first "one" is read, it will activate a relays and switch on the lamp

(indicating an odd number of ones). As the tape proceeds, the read position will be left and the "one read" signal disappear, and thus releasing the relays. But the current through the lamp will remain, as a second relays in parallel to the lamp voltage was turned on, offering the current a new way to the lamp.
In the "no one read" position, the current state is also forwarded to a "saved state storage".
As soon as the second one is read, the relays left of the lamp will be activated again, and the "saved state storage" will keep the old state information, by connecting itself to the voltage source, if the state was "odd" or remaining unconnected in case the old state was "even".
The saved state information is inverted by another relays and set as the current state.

This is continuing until the tape is at the end. A lit lamp at the end will thus indicate, that there was an odd number of ones while a unlit lamp stands for an even number. (The capacitors in parallel are here to keep the informations while the switches are in no distinct position)

4.3 Field effect transistors

"Modern" computers are mostly built upon semiconductor based integrated circuits of transistors.
A transistors is an electronic device that can be used as a switch or for amplification, thus the same logic-circuits as built before with relays and vacuum tubes can also be made of transistors, but with much less space needed. Also temporary storage devices known as "RAM" are based on those transistors.

There are several different kinds of transistors (BJT, JFET, MOSFET,...) , but they all have in common that there is a "source", a "gain" and a "drain" contact. The amount of current going from "source" to "drain" can be controlled by the voltage applied to the "gate".

In the following, the operation of one transistor type, the metal-oxide semiconductor field effect transistor (MOSFET) is explained.

Transistor schematic [5]

The "source" and the "drain" are inserted in a silicon plate and a "gate", separated from the silicon plate by metal-oxide is mounted in between them.

The poor conducting silicon is doped to reduce the band gap and allow some "electrons" or "holes" to move. But as long as there is no charge on the gate, no big current will flow between source and drain. (b)

But as soon as a charge is applied to the gate, the electric field generated by it will either attract electrons or holes in the area between source and drain (c). (-> metal-oxide semiconductor field effect transistor)

A voltage drop between source and drain will now lead to a current flow.

The transistor can either be used as an amplifier, because the stronger the electric field of the gate is, the more electrons and "holes" are available and thus, the more current will flow, or as a switch, using just the states "current is flowing" or "no current is flowing".

Transistor sizes have already been reduced to less than 50nm gate length (Intel, [10]...), so they are already in the nanoscale range.

30nm gate length transistor realized in 2000 [10]

But there are several problems that will probably soon stop the process of reducing transistor size.

In the 30nm gate size transistor presented on the right, attention must be posed on the word "gate size" as the whole transistor is bigger. Also the connection plates of "drain" and "source" need to be reduced.

D. Goldhaber-Gordon et al. [5] point out that at small dimensions uneven distribution of doping elements, tunneling from source to drain when "off" and leakage in the oxide layer will limit the functionality of the transistors. Also heat dissipation is seen as a major problem.

4.4 Quantum electronics

One effect that is a problem for further miniaturization of semiconductor field effect transistors, tunneling, can be used to create a new family of logic devices: the quantum electronic devices.

4.4.1 Resonant Tunneling Devices

Resonant Tunneling Diode (RTD)

The resonant tunneling Diode works in the following way:

Working principle of a RTD. [5]

Between a source and a drain contact an isolated area (island), enclosed by potential barriers, is placed. (a)
The potential wells prevent current flow between source and drain. But depending on the energy state of the electrons in the island, tunneling can occur or not. When the island is made of a material, with electron energy levels higher than the energy levels available on the source for tunneling (a), the diode is "Off" in the normal state. By lowering the bias voltage, the diode will switch it's state to "On", as soon as the energy levels inside the island are low enough to allow tunneling (c).

When a material with multiple, well separated electron bands is used for the island, a diode with multiple on-off states can be constructed, leading to a new kind of logic: multi value logic.

Resonant Tunneling Transistors (RTT)

By controlling the energy states in the isolated area, a Resonant Tunneling Transistor (RTT) can by built. First working RTT's have already been made in the late 1980s at the Central Research Laboratories of Texas Instruments [3] [33].
One way to control the energy states of the isolated area is by introducing a Schottky Gate [9]. By applying voltage on the gate the current flow can thus be controlled, alike in field effect transistors (FET).

Schematic of the vertical RTT produced by J. Stock and A. Förster [9]

SEM picture of a vertical RTT [9]

But compared to FET's, also the RTT's have, as the RTD's the possibility of multiple on-off states, what enables them to be used in multi value logic. As a result they do not only allow further miniaturization as tunneling is not a limiting factor, they also allow to built more compact and less complex logic, as less transistors are needed in multi value logic.
According to J.Stock and A. Förster [9] RTT's also work at higher speed and dissipate less power than conventional FET's.

4.4.2 Quantum Dot

Scaling the island of a resonant tunneling transistor down, finally ends up with a quantum dot, that can contain just a distinct number of N electrons with zero degrees of freedom. The number N depends on size and shape of the dot. As the electronic behavior of these Quantum Dots is comparable to single atoms, they are sometimes also referred as "artificial atoms" [36].

Different shapes of Quantum Dots [36]

To tunnel to the dot, energy is needed to overcome the electrostatic repulsion.
This effect is called "Coulomb Blockade". This additional energy can be delivered by a gate voltage near the dot. Whenever the gate voltage exceeds the limit of the Coulomb Blockade for an additional electron to tunnel to the dot, a jump in the current between source and drain can be observed.

The difference to the resonance tunneling transistor is, that while increasing the gate voltage on the RTT the current changes between "low" and "high", while for the quantum dot, the current increases in steps.

4.4.3 Single electron transistors (SET's)

AFM image of a SET [34]

At first sight, the structure of single electron transistor (SET) looks similar to resonant tunneling devices and quantum dots, consisting of an isolated area, a source, a drain and a gate. The difference is that the isolated island of a SET is much bigger containing more mobile electrons, forming an continuous energy spectrum rather than quantized states. Care has to be taken with the therm "bigger", as it does not only relate to the dimensions but also depends on the material used for the island. (Depending on the material, an island of the same size can contain distinct quantized energy levels or a continuous spectrum) [5].

As there is a continuous energy spectrum inside the island, tunneling is not limited by the energy state, as there are plenty of them available. What limits the current flow in a SET is the "Coulomb Blockade": The electrostatic repulsion of the electrons already on the islands prevents new electrons to tunnel in. Only if one electron leaves from the island to the drain, another one can tunnel in from the source (-> Single electron transistor). The number of electrons on the island can be controlled by an voltage applied to a nearby electrode (the gate), working like a capacitor. The more electrons that are on the island, the higher is the tunneling rate and thus, the higher the current from source to drain will be. The major advantage of SET's, apart from the possibility of further miniaturization, is their lower energy consumption, as they operate only with a few tunneling electrons.

SET circuit schematic

[35]

The critical point of single electron transistors is their operation temperature. The higher the temperature, the more energy is available for electrons to overcome the Coulomb Blockade. For a long time only SET's working at very low temperatures have been built, reducing their capability for practical applications. However successful construction of room temperature devices has been reported. For example a group at DEFT [11] was able to produce a room temperature SET

with a metallic single-wall carbon nanotube, based on the effect that buckles in the tube work as barriers for electrons.
However the huge drawback of complete different behavior at other temperatures remains.

Another problems of SET's, shared with Resonant Tunneling Transistor and Quantum Dot transistors is, that also here the miniaturization will come to a limit, as soon as the island dimension is so small that direct tunneling from drain to source becomes probable.

4.5 Molecular electronics

As chemistry offers an immense number of tools to manipulate structures on the molecular level, the construction of molecular electronics is an option.

Nano-Tube based Devices

Based on Nanotubes, different devices are imaginable: They can be used to implement the quantum-effect, like the single electron transistor described in the previous section [11], as well as mechanical principles can be used. In his paper about nano-electronics A.K. Ojha [26] lists several groups that developed crossbar switches: two arrays of conducting nanotubes with a small space between them can form either a open circuit, when they have no physical contact or, a closed circuit, when mechanical contact is made.

Molecular switches

Switches can also be made of molecules that change their structure such as bistable rotaxane [21] [14]. Rotaxanes molecules can either allow current to flow or not. When a current of more than 0.7V is applied to them, they become oxidized what makes them isolating.

Rotaxane monolayer used as a switch [14]

4.6 Chemical Computation

As computation does not necessarily need to be based on electric circuits, absolutely different methods are possible.

4.6.1 DNA-Turing machine

Since DNA can be seen as a tape of data, several researchers [[12][22] and others] came up with the idea to build a Universal Turing machine based on DNA.

As an example the "Autonomous DNA Turing Machine" (ADTM) built by P.Yin, A. Tuberfield, S. Sahu and H. Reif [22] shall be presented here:

Shematic drawing of the ADTM. [22]

This ADTM is working in solution and consists of two parallel arrays of dangling molecules connected to two rigid tracks. The upper array forms the "head" (see Universal Turing Machine) while the under array contains the symbols if the "tape"
In addition there are free floating DNA molecules in the solution: Rule molecules (The "transition rules", that represent the head logic) and other assisting molecules.

All the symbol molecules (bottom row) have sticky ends and have an information encoded in its DNA. The symbol molecule can be either long (the information is encoded in "xyz") or short, where the information is stored in "Txy". The information can have five different colors according to the table below. (Remark: Computers usually work in binary code where the single bits contain a "1" or a "0".)

Information encoding scheme

As said above the molecules can be long or short. This is used to encode the state. Thus this ADTM is only able to have two states.

Also the head molecule can be short or long, depending on the state.

State "long" on the left side and "short" on the right one. [22]

In the picture above, H stands for the head-molecule and S for the symbol-molecule.

On the left side the state is long, the head molecule has an AA-sequence while on the right side, the state "short" is encoded with AAA. This might seem strange but will become clear later on.
The symbol molecule DNA contains the current color in xyz.

While all the symbol-molecules have sticky ends, just one of the head molecules has an active sticky end.

The head molecule with the active sticky end
is marked with an arrow. [22]

Operation

1. The active head molecule ligates to the corresponding symbol molecule.

Ligated active head molecule (1) [22]

Ligation in state "long" (left) and short (right) [22]

2. An endonuclease cleaves the ligation product in two parts, which now contain both the information about state and color.

Cleaved parts S3-H3. [22]

The state (long) is represented by the fact that the color is encoded with xyz (and the complementary DNA respectively) in head- and symbol-molecule. [22]

The state (short) is represented in the head molecule by the fact that the color is encoded in it's short form Txy and in the symbol molecule by the additional A. [22]

3. Floating rule molecules which have sticky ends complementary to the ends of the products of stage 2 ligate to the ends.

As the sticky ends of the rule molecules are complementary, it is state and information selective, which rule molecule can ligate. This corresponds to a "if then.." construct in conventional programming language.

Ligating rule molecules.[22]

Rule molecule ligating to a symbol molecule (with state long). Process is similar in the head molecule. [22]

4. The rule molecules are again cleaved from head- and symbol-molecules. The length of the spacer region ("----") of the rule molecules determines where the cut takes place.

Rule molecules are cleaved away. [22]

Rule molecule cleaved from symbol molecule. The length of the spacer region "--------" determines the cleaving spot.. [22]

The sticky end of the new symbol molecule contains the information about the new color to take. The cleaving at the head molecule is similar, and leaves a new head molecule with the new state and a sticky end that encodes the moving direction of the head.

5. An assisting molecule modifies the symbol so that the color information is again stored in the duplex portion rather than in the sticky end.

Possible sticky ends of the symbol molecule after transition 4 are GT, TG, GA, AA and AC which are transformed by the assisting molecule to TTA, CTT, CAA, AEA and CEA.

Ligation and cleaving of a assistance molecule to change the symbol molecule. *

*The table is a modification of Fig. 5 in [22] as there was a typing error in the original (CTT changed suddenly to ATA in the original paper).

The head hybridizes with it's right or left neighbor according to it's sticky end and cleaves again.

Hybridizing head and modification of symbol molecule. [22]

6. Assisting molecules modify the head molecule so that, in this example, H3

remains inactive and H3 becomes the new active head molecule with the current state encoded. (Head-modification similar to the symbol molecule modification)

Now we finished a whole loop and the procedure begins again.

Active head moved one step left. [22]

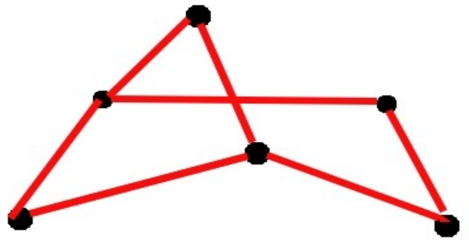
What we have here is indeed an Universal Turing Machine, as we have a tape (the symbol array) to read from and write to. A head that can move (the active head molecule) and "the software" encoded in the rule molecules.
With an accurate stop rule this UTM is able to perform computation. A challenge is that unlike the tapes used by computers a few decades ago, the DNA-output tape is very small, so that it needs some effort to read out the result.

4.6.2 DNA-Computing

Instead of trying to build a Universal Turing Machine, DNA can also be used for computation in a complete different way.
An example is the DNA based solution of the traveling salesman problem by L.M. Adleman back in 1994 [4] (Description in [17]).

The traveling salesman problem (Hamilton Path problem)

The task of the traveling salesman problem is to find a way to visit a distinct number of cities in the fastest way without visiting one city twice.



Left: The traveling salesman problem with 6 cities (black dots) and the routes (red) between the cities.

With a few cities the problem seam easy. But the challenge is increasing with the number of cities. To solve the problem with 100 or more cities, modern computers take years to calculate the optimal solution.

Adlemans DNA-based solution [4] (Description in [17])

In a first step, single stranded DNA molecules, each representing a city, are made. The first 10 nucleotides of each city molecule encode a possible way to enter the city, the last 10 a possible way out. A "city-molecule" looks the following:

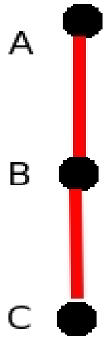
(road entering the city)-(city)-(road leaving the city)

Every in-out combination has to be represented by a molecule type. If a connection between two cities is possible, there must be molecules of the two cities with one complementary 10 nucleotide part, so that they can ligate to a complete "road".

(road entering A)-(A)-(road from A to B)*-(city B)-(road leaving B)

**double stranded part consisting of ligated 10 nucleotides single stranded parts*

Now the molecules are mixed and ligation occurs. If the number of molecules was high enough, the product will represent every possible connection. After filtering out all molecules, that are too short, too long or contain a "city" more than once, the remaining molecules are the solutions to the traveling salesman problem.



Three city problem

The different molecules for the three city problem on the left would be:

(B->A)-A, A-(A->B)*, B-(B->A), B-(B->C)*, (A->B)-B-(B->C)*, (C->B)-B-(B->A)*, (B->C)-C, C-(C->B)*

where the parts with stars * are the complementary codes of those without.

The only ligation solutions with all three cities in there will be:

A-(A->B)*-(A->B)-B-(B->C)*-(B->C)-C

and

C-(C->B)*-(C->B)-B-(B->A)*-(B-A)-A

Adleman made an experiment with 7 cities. The solution was computed after 1 sec. The "computation speed" was about 1,014 operations per seconds what is equal to 100 Teraflops (The fastest computer in 2002 achieved about 222 Teraflops).

However it took Adleman a week to filter the products until he had the solution molecules. This amazing speed is basically achieved by the "parallel processing" of the high number of molecules in the solution, what leads to another disadvantage mentioned my Jack Parker [17]: To compute the solution of a 200 city problem, the weight of the needed molecules would be bigger than the weight of the earth. Thus this technique will also not lead to solutions of big city number problems

5. Conclusion

Fictional Nanobot [37]

The idea of nanorobots seems futuristic, but the future is might nearer than one thinks.
But to expect nanoscale machines like the one on the right, is wrong too. Many science fiction authors that constructed the widely used image of nanobots just saw them as mechanical machines, scaled down.
But, as we have seen, in the nanoscale range, the dominant effects are different from the ones in the macroscopic world. Brownian Motion, friction and tunneling play minor roles in the world known to us.
Rather than those small scale mechanical devices with gears, simple organic structures should be expected.
In fact, the modified viruses used in genetic engineering to introduce DNA-sequences to genomes could already be seen as biology-based nanobots.
Of course the scaling down of devices will at some point reach it's definite limits as the size of atoms sets final barriers. But for most applications, virus-dimensions are small enough.

The four types of actuators presented showed that actuation on the nanoscale level has already be done. Even the "new" effects like Brownian Motion have been exploited. The main problem here is to assemble those structures in a way that they produce a useful mechanical work. A motor floating in a solution and spinning around its own axis doesn't help much.
Control of nanobots, even limited by their size, should be possible, as also with limited computation resources, quite complex behavior can be achieved. In therms of the logical devices some problems of "classical" field effect transistors, like non-homogeneous doping and heat dissipation can be solved by tunneling devices (RTD, RTT, QD, SET), as they consume less power and are based on other principles. But also here, the assembly is a problem. To produce one quantum dot is one thing, but to built up a complete logic device is complete different.
Molecular electronics can help as this point, as chemistry already offers a broad range of tools to manipulate molecules. But even those elements are limited, as tunneling can occur directly from source to drain.
Chemical computation is probably the one that is most likely to be used in nanorobots. Drug delivery for example just needs a shell that opens when an environmental condition, like the presence of an other chemical, is met.

6. References

Papers

[1]
A.M. Turing, *On computable numbers, with an application of the entscheidungsproblem*, 1936

[2]
Valentino Braitenberg*, Vehicles, Experiments in Synthetic Psychology,* 1984

[3]
M.A Reed, W.R Frensley, R.J Matyi, J.N Randali, A.C Seabaugh, *Realisation of a trhee-terminal resonant tunneling device: The bipolar quantum resonant tunneling transistor*, 1989

[4]
L. M. Adleman, *Science 266*, 1994

[5]
D. Goldhaber-Gordon, M.S. Montemerlo, J. C. Love, G.J. Opiteck, J. C, Ellenbogen, *Overview of Nanoelectronic Devices,* 1997

[6]
P.Kim, C.M. Lieber, *Nanotube Nanotweezers*, 1999

[7]
Owen Holland, Chris Melhuish. Stigmergy, *Self-Organization, and Sorting in Collective Robotics*, 1999

[8]
Benjamin Chui, Lea Kissner, *Nanorobots for Mars EVA Repair,* 2000

[9]
J. Stock and A. Förster, *A vertical resonant tunneling transistor for application in digital logic circuits*, 2001

[10]
Robert Chau, *30nm and 20nm Physical Gate Length CMOS Transistor*, 2001

[11]
Henk W. Ch. Postma, Tijs Teepen, Zhen Yao, Milena Grifoni, Cees Dekker
*Carbon Nanotube Single-Electron Transistors at Room Temperatur*e, 2001

[12]
Yaakov Benenson, Tamar Paz-Elizur, Rivka Adar, Ehud Keinan, Zvi Livneh, Ehud Shapiro, *Programmable and autonobous computing made of biomolecules,* 2001

[13]
P Bøggild, T M Hansen, C Tanasa and F Grey, *Fabrication and actuation of customized nanotweezers with a 25 nm gap*, 2001

[14]
Eric KJ Lerner, *Making Molecular Switches*, 2001

[15]
Jianwei J. Li and Weihong Tan, *A Single DNA Molecule Nanomotor*, 2002

[16]
C.A. Haberzettl, *Nanomedicine: destination or journey?,* 2002

[17]
Jack Parker, *Computing with DNA*, 2003

[18]
F. Mondada, A. Guignard, M. Bonani, D. Bär, M. Lauria, D. Floreano. *SWARM-BOT: From Concept to Implementation*, 2003

[19]
M. Schliwa, G. Woehlke, *Molecular motors*, 2003

[20]
A. A. G. Requicha, *Nanorobots, NEMS, and Nanoassembly*, 2003

[21]
Yong Chen, Gun-Young Jung, Douglas A A Ohlberg, Xuema Li, Duncan R Stewart, Jan O Jeppesen, Kent A Nielsen, J Fraser Stoddart and R StanleyWilliams, *Nanoscale molecular-switch crossbar circuits*, 2003

[22]
Peng Yin, Andrew J. Turberfield, Sudheer Sahu, and John H. Reif*, Design of an Autonomous DNA Nanomechanical Device Capable of Universal Computation and Universal Translational Motion,* 2004

[23]
Keith Firman, *Enzymes as molecular motors*, 2004

[24]
Yi Chen, Mingsheng Wang, and Chengde Mao, *An Autonomous DNA Nanomotor Powered by a DNA Enzyme*, 2004

[25]
J. V. Hernández, E. R. Kay and D. A. Leigh, *A reversible synthetic rotary molecular motor*, 2004

[26]
Anand k. Ojha, Nano-electronics and Nano-Computing: Status, Prospects, and Challenges, 2004

[27]
McLurkin James D,*, Stupid Robot Tricks: A Behavior-Based Distributed Algorithm Library for Programming Swarms of Robots*, 2004

[28]
Shen WM, Will P, Galstyan A, Chuong CM. *Hormone-inspired self-organization and distributed control of robotic swarm*s, 2004

[29]
Richard L.Cross, *Turning the ATP motor,* 2004

Sources on the net

[30]
Craig Reinolds
http://www.red3d.com/cwr/boids/

[31]
I-Swarm
http://i60p4.ira.uka.de/~seyfried/tikiwiki-1.7.3/tiki-index.php?page=I-Swarm

[32]
Biomolecular motors with metal propellers, Cornell University
http://www.news.cornell.edu/releases/Nov00/propeller.hrs.html

[33]
RTT at TI
http://www.ti.com/corp/docs/company/history/quantum.shtml

[34]
MITRE Nanosystems Group http://www.mitre.org/tech/nanotech/single_electron_transistor.html

[35]
Single electron transistors, K. Matsumoto
http://snowmass.stanford.edu/~shimbo/set.html

[36]
Research in the group of Leo Kouwenhoven, Delft
http://vortex.tn.tudelft.nl/grkouven/qdotsite.html

[37]
http://www.foresight.com