

EN291-Scribe Report
Werner Luis Camacho
Desta Tadesse

NOTES ON MOLECULAR COMPUTING

For this presentation, we have discussed the content of the following articles:

- Nanocell Logic Gates for Molecular Computing
- Logic and Memory With Nanocell Circuits

We are going to present a summary of the reading contents of both papers and an outline of the discussions brought up during class.

Nanocell Logic Gates for Molecular Computing

The primary goal of this paper is “to demonstrate the initial algorithmic feasibility of programming a nanocell to perform a logic operation”.

By implementing logic and memory with this approach, we can reduce the size of the device and fabrication cost. However, it is unclear how we’re going to be able to place and interconnect these diminutive systems.

The corner stone of this approach is the *nanocell*, which is a network of self assembled metallic particles connected by molecules that show reprogrammable NDR (negative differential resistance). In other words, it can be turn ON and OFF. Unlike typical chip fabrication, logic in the nanocell is created by training in postfabrication.

Different simulations have been performed using Spice making use of some assumptions for the sake of simplicity. Once the physical topology of the self-assembly is formed in the nanocell, it remains static. The only changeable behavior is in the molecular states: conducting ON or non-conducting OFF.

The functionality of a nanocell depends largely on the I(V) characteristics and placement of its molecular switches with respect to the nanoparticles. Recent simulations displayed a ratio of 1000:1. Though, it has not been obtained experimentally.

For the programming part, the nanocell is trained post fabrication by changing the states ON or OFF. The article lists 2 programming algorithms: Omnipotence and Mortal.

- Omnipotence means that the search algorithm knows the location of each molecular switch and has precise and selective access to reversibly set its ON and OFF.
- Mortal switching means that the algorithm doesn't know the connections within the nanocell or locations of the switches and switching is limited to voltage pulses applied to the I/O pins.

The optimum case consists on being able to use mortal training because it will not be necessary to uniquely access each molecule. "Theoretically, a line of molecular switches between two I/O pins, where there is some capacitance between two particles, can be set to any pattern of ON and OFF states by using these capacitances." This mode of programming from disorder is an exciting approach!!!

In terms of modeling the nanocell, GA and omnipotence switching has been used in training the nanocells.

Using GA training in simulations these researches have been able to discover inverters, NAND gates, half-adders and 1-bit adders. GAs work by taking population of individuals, represented as strings of "1s" and "0s" quantifying their fitness, then recombining them to generate a new population of children. First two parents must be selected, they must be recombined to form two new children. If the length of the chromosome (the string of "1"s and "0"s representing each individual) is m , then some point p between 1 and $m-1$ is chosen as the crossover point. To create the first child, the first p bits of the first parent are combined with the last $m-p$ bits of the second parent. After crossover, each of the two new children is mutated and the states of the nanocell's switches are stored as a chromosome of "1s" and "0s"

Logic and Memory With Nanocell Circuits

This article focuses on the nanocell post-fabrication process. To program a nanocell we basically need to follow two steps:

1. Apply voltages that change the internal switch states
2. Test the nanocell's logical function without changing the internal switch states.

(Therefore if we are to obtain logic gates exhibiting negating logic, NDR is essential because we do not want any molecules to switch "on" or "off" after the programming phase)

Directly following fabrication, the nanocell does nothing more than conduct electricity across metallic leads in a very nonlinear fashion. The next step in the overall approach is to program individual nanocells to perform certain logical or memory functions. This is

accomplished by applying voltage pulses in an undefined order to the metallic leads. Individual molecules will then switch states according to their voltage dependent switching rules.

The simulations require some assumptions:

1. The connections within the nanocell are known, that is, we know the graph within the nanocell
2. We have control over the state of each individual switch so that each switch is selectively, reversibly settable.

For nanocells to interconnect they ideally want:

- Fan-in and fan-out
- Restoration
- Uniformity of signal
- Isolation

Therefore, their approach is to use a bistable latch on the output, which converts the current output to a voltage that can serve as an input to another nanocell.

By attaching a latch to the output of a nanocell, it's possible to use the three bias voltages to reset-set-hold the output voltage.

With this setup, it is possible to fabricate a 1 bit adder with three nanocells, each with one pin per side. By training several nanocells in this fashion they came out with a half-adder, that were trained with output pins adjacent and with output pins on opposite sides of the nanocell. According to their calculations, the adders produce in this fashion are clearly extendable to an n-bit adder and the two-bit register is easily converted to an n-bit register. Moreover, they plan to use neuro-dynamic programming to mortally program nanocells. In this ideal scenario, design logic and memory circuits are dynamically reconfigurable so that the circuit functions as logic in some instances and memory in others.

Discussion

The concept of using molecular devices as switches and electronics is now becoming an accepted avenue of research in nanotechnology. However, not all avenues lead to realistic or practical devices for nanoelectronics. To put the cart before the horse, the class consensus was that the ideas presented in the paper are too far removed from practicality at this time. However, it was also noted that the paper is a proof of concept into a fundamentally different approach to circuit design. There are bound to be various issues, yet the outlook seems promising. Here are some of the issues that were raised in our discussion.

The main criticism against this idea of using a nanocell filled with molecular switches is that the authors have distilled their main idea to a very unrealistic level by making major assumptions. They start with the idea of using a nanocell and programming the switches to make a logic circuit. This idea was accepted by everyone and is not a novel topic, nor is it the main essence of the paper. The interesting aspect/contribution comes in how they propose to program these nanocells to burn in the logic without any knowledge of where the switches are located or even the state of the switches in a process they call mortal programming. However, they make two major assumptions that take them away from the original concept. They assume that they can control every switch and know its state at any time. The class concluded that these assumptions are far too removed from the main idea.

Another criticism that was raised was against their method of “experimentation.” None of the nanocells were fabricated (which we agreed is not a necessity to prove a concept), and even more they did not show any of their SPICE simulations. The I-V curves they showed were based on a SPICE model and seem to be “better” than what you would get for a solid-state semiconductor. This didn’t go well with the class, especially the two device people. And furthermore, the area comparison (of an adder) between the nanocell approach and a regular approach did not seem reasonable; they said it takes 50 transistors to make an adder (usually requires ~25 transistors for a regular adder.)

Attention was also focused on the resistance on/off ratio. The papers claimed that on/off ratio has improved (in simulations) from 2:1, to 1000:1. The class was concerned about how these ratios are a lot smaller than regular switches (with ratios $\sim 10^6$) and the problems this would cause with the gain. The major question arising is how do you make a circuit out of cells that have low gain? If we cannot have cells with good gain, then you would need to fit most of the functionality in a nanocell. Various questions were raised from this: How much functionality can you put in a nanocell? (Not too many!) Would putting in a lot more functionality increase the complexity of programming? (Yes, and it is already not clear on how you program the cells.) And if we can’t put in a lot more functionality, how big do these “nanocells” have to be to accommodate what we need? (Bigger than a nanocell, maybe.) Along the same lines of using a nanocell as a basic circuit block, interconnect issues were raised. However, it was pointed out that these nanocells are not really in the nano region allowing one to use current lithographic techniques to make interconnects. However, how these molecular devices would respond to the lithographic process is not known as none have been fabricated yet.

An interesting point that was raised in terms of understanding molecular switches is the idea of a redox potential. The I-V curve shows a stage where the current rises before it falls as it exhibits NDR behavior. Question was raised about why the current would rise at first when the device should be exhibiting negating functionality. The reason might be due to a redox potential; i.e., the switch needs some voltage to go through a redox process. At this stage, the device might not be in the “steady-state.”

Various other issues were raised such as the philosophical/general question of whether one needs to know what is going on inside a black box to program it, questions dealing

with the use of GA (which are used a lot in the CS world) and their applicability in this topic, etc...

In one part, the paper seems to be a bit unrealistic (at this time). Along with this, the gain problems along with the functionality that can be squeezed into the cells add to is also a bit of a problem. However, there is also plenty of reason to commend the paper. It is a fundamentally radical approach to circuit design and the whole nano craze. It is not using nano devices to make FETs or array (nanoPLAs), but a completely new approach. Also, the omniscient approach was demonstrated, not as a practical programming approach, but to determine if this idea of programming from disorder had any potential hope of being worthwhile to pursue. With this approach, they have given a credible proof of concept which now paves the way for working towards the idea of mortal programming.

The class concluded that this is the kind of thinking that is going to take us beyond the roadmap wall of 2016.

The moral of the story: *All great things sound crazy in the beginning.*