Molecular NOR Logic Gate

Piotr Wąsiewicz[1], Anna Wójtowicz-Krawiec[2], Andrzej Płucienniczak[2]

[1] Institute of Electronic Systems, Warsaw University of Technology
Nowowiejska 15/19, 00-665 Warsaw, email: pwasiewi@elka.pw.edu.pl
[2] Institute of Biotechnology and Antibiotics
Staroscinska 5, 02-516 Warsaw, email: apl@iba.waw.pl

Abstract. Molecular computing created for implementing logic systems, solving NP-difficult problems on nanoscale depends on DNA self-assembly abilities and on modifying DNA with the help of enzymes during genetic operations. In the typical DNA computing a sequence of operations executed on DNA molecules in parallel is called an algorithm, which is also determined by a model of DNA chains. This methodology is similar to the soft hardware specialized architecture driven here by heating, cooling and enzymes, especially polymerases used for copying strings. This work presents a unique approach to implementation of OR, NOR logic gates on molecules. It requires the representation of signals by DNA molecules. The presented method allows for constructing logic gates with many inputs and for executing them at the same quantity of elementary operations, regardless of a number of input signals. The NOR gate was implemented with the help of modified polymerase Taq, which stops its activity, when it meets a molecular obstacle on its way. The appropriate experiment was conducted to confirm the possibilities of the suggested implementation. Laboratory results were discussed.

1 Introduction

Before implementing molecular solid-state circuits it is easier to test molecular system potentials during liquid computing driven by heating, cooling, and chemical reactions - so called molecular computing introduced by Adleman [1]. Solution to another NP-Hard problem of logic function satisfiability was invented by R.J. Lipton [5].

Other researchers used DNA to solve many computational tasks e.g. [2, 3, 4, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]. Those described in the next points works were arbitrarily chosen from molecular computing papers and present experiments, which results can influence on the next level molecular computation.

In this paper we propose a new approach to molecular negation, which leads to the NOR gate creation.

2 DNA description

A single-stranded DNA string has a phospho-sugar backbone with two different 5' and 3' ends and four bases Adenine, Thymine, Cytosine, Guanine denoted by the symbols:

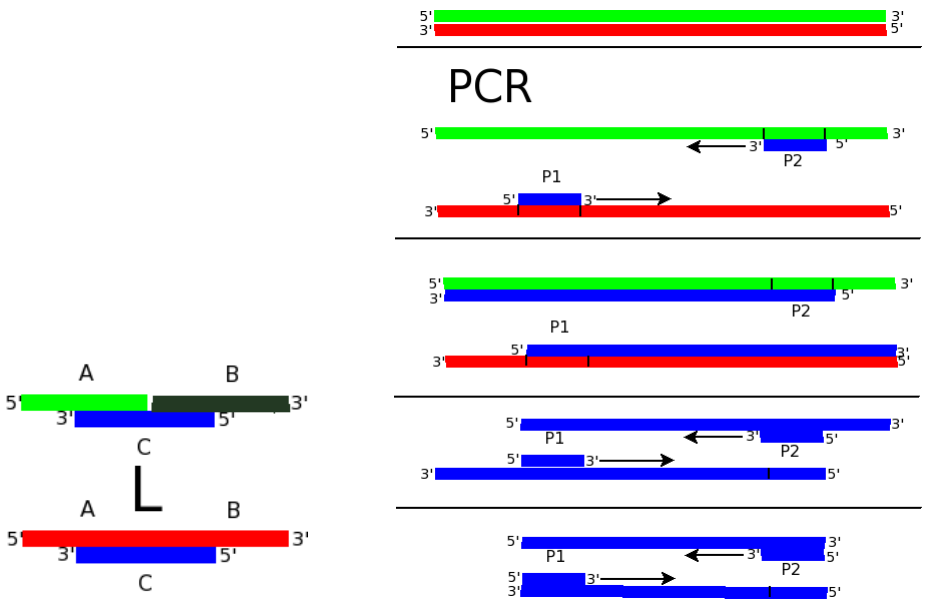

Figure 1. On the left: ligation of A and B oligos in the presence of hybridized third complementary one - C. On the right: the extension of primers' 3' ends in the amplification process of PCR

A, T, C, and G, respectively. A double-stranded DNA string is in the form of oriented in opposite directions two single strings due to hybridization or in other words annealing reaction, because A is complementary with T, and C is complementary with G. Operations on DNA oligos [6] may be described in the following way:

1. Hybridization or renaturation means annealing of single complementary DNA strings to single standard DNA strings and forming double stranded molecules. This operation is caused by cooling down the test tube reaction solution.
2. Denaturation means changing double stranded DNA molecules into single complementary and standard strings. Heating of reaction solution causes it.
3. Cutting of a double DNA string into two parts is performed in DNA computing with the help of digesting restriction enzymes. The resulting double strings can have sticky ends (single stranded DNA) or blunt ends. Methylation of DNA usually has the effect of protecting the DNA from the related restriction endonuclease.
4. Concatenation of two strings is a string formed by placing the second string after the first string without any gap. In DNA computing joining of two strings is done during hybridization and ligation. They form together a longer single string. In order to concatenate two oligos a and b, the hybridized and complementary to them in the place of joint third string is needed as is seen on the left side of Fig. 1.
5. Amplification (PCR). In the cycle of amplification a single strings called primers are lengthened from its 3' end up to its complementary longer string 5' end. Thus, the primer has to be hybridized e.g. far away from longer string 5' end. Then a special enzyme called a polymerase lengthens the primer adding to the primer 3'

end nucleotides complementary to the longer string which this primer is attached to. The amplification result is in the double stranded form and its length is equal to the previous longer string one as is seen on the right side of Fig. 1.

A signal is represented by a DNA string called an oligo, an oligonucleotide, a strand, a DNA fragment, a DNA chain. A sequence of operations on DNA strings is called an algorithm. Together the genetic operations, driven by enzymes, heating and cooling, DNA sequence, and the model make computation possible.

3 Polymerases and their modification process

Utilized in amplification process all DNA polymerases share two general characteristics [6]:

1. They add nucleotides to the $3' - OH$ end of a primer. The order of the nucleotides in the nascent polynucleotide is complementary to a oligo, which is hybridized to the primer.

2. In addition to the $5' \rightarrow 3'$ polymerase activity, polymerases can contain exonuclease activity. This exonuclease activity can proceed either in the $5' \rightarrow 3'$ direction, or in the $3' \rightarrow 5'$ direction.
   - Exonuclease activity in the $3' \rightarrow 5'$ direction allows the polymerase to correct a mistake if it incorporates an incorrect nucleotide (so called "error correction activity"). It can also slowly degrade the 3' end of the primer.
   - Exonuclease activity in the $5' \rightarrow 3'$ direction will allow to degrade any other hybridized primer it may encounter. Without $5' \rightarrow 3'$ exonuclease activity, obstructing primers may or may not be physically deplaced, depending on the polymerase being used.

Futher, after adding all needed nucleotides in primer lengthening process polymerases usually begin to disassemble finished products, so it is very difficult with use of natural enzymes to construct nanodevices. Good synthesized enzymes are still not available and it is even impossible to synthesize new kinds of them. The only solution is to change properties of the existing ones.

The Taq polymerase was modified in the following steps described in this paper [18]:

- DNA sequence modification. A DNA fragment responsible for $5' \rightarrow 3'$ exonuclease activity was cut off by amplification reaction from the polymerase DNA sequence in order to remove this polymerase ability, which destroys 5' single oligo ends. This means only polymerase gene part responsible for polymerase activity was cut out of the whole Taq polymerase gene. The rest of polymerase peptides was lost.

- DNA structure evaluation. The amplified product was put in plasmid vector (circular DNA molecule) and cloned within bacteria colonies inside cells in order to reconstruct DNA structure, which is more suitable for living cell environment e.g. with all methylase DNA protection against restriction enzymes.

- Polymerase generation. After retrieving this new DNA from bacteria plasmid clones, it was placed in an expression vector. The obtained protein from next bacteria was just a modified Taq polymerase [6].

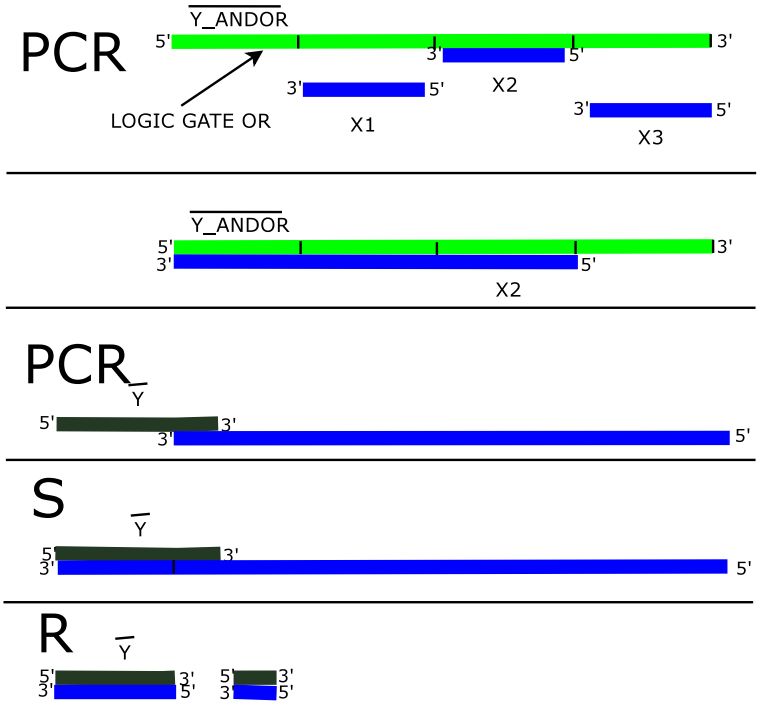

Figure 2. Molecular OR gate $X_1 \vee X_2 \vee X_3 \Rightarrow Y$

4 Molecular system implementations

Molecules can play and have intelligent behaviour. The first game-playing DNA computer was revealed a year ago. An enzyme-powered tic-tac-toe machine that cannot be beaten was developed by Milan Stojanovic, and Darko Stefanovic [14]. Faulhammer, Lipton, Landweber [3] developed RNA-based computing approach to a 9-bit instance of the "Knight Problem". Wood, Bi, Kimbrough, Wu and Chen made DNA learn Poker by using independent populations of strategies that adapt and learn from their experiences in competition.

Molecules can be utilised in developing of a nanomachine molecular automaton and molecular state transitions. Design of nanomechanical devices based on DNA was proposed by Reif [15]. Simmel, and Yurke prepared experiments with DNA nanoactuators [12]. DNA nanomachine "scissors" were invented in Bell laboratories by Yurke and Mitchell [7]. In [10] Sakamoto showed that combined with parallel overlap assembly, a single series of successive transitions can solve NP-complete problems.

Ogihara and Ray [8] first demonstrated that DNA computers can simulate Boolean circuits. Klein, Leete and Rubin [4] created universal three input logic gate based on PCR. Amos and Dunne [2] described the abstract model and its own laboratory implementation. Hagiya et al [16] designed one molecule DNA computer with data and operations on one DNA strand. Computation of logic function satisfiability was driven

by PCR reaction. Wąsiewicz [20] also proposed the evolutionary programming of logic function graphs, the evaluation of which is based on PCR.

Surface-based methods were presented by Liu, Smith and their research group [13]. Complex combinatorial mixtures of DNA molecules encoding all possible answers to a computational problem were synthesized and attached to the surface of a solid support, especially designed for executing logic gates.

Molecules can add integer numbers. Wąsiewicz, Rudnicki, Mulawka and Lesyng [19] presented new algorithm of DNA computing for adding binary integer numbers, which add numbers at the same quantity of elementary operations, regardless of a number of bits used for representation.

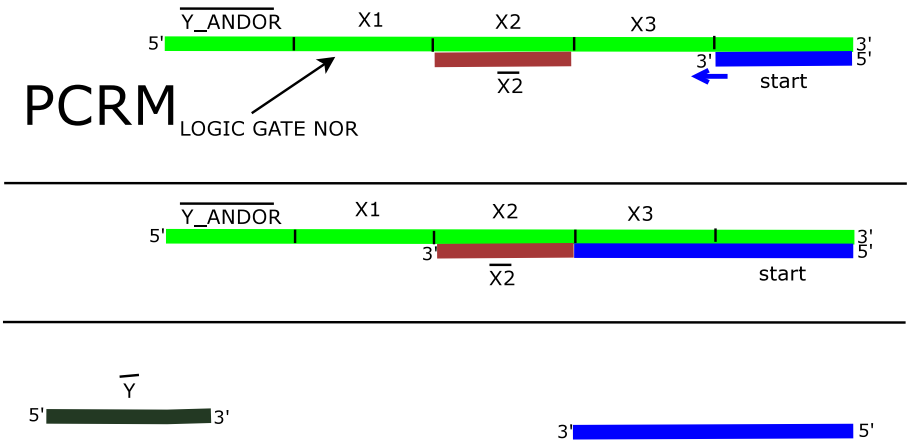

Figure 3. Molecular NOR gate $\neg(X_1 \vee X_2 \vee X_3) \Rightarrow Y$

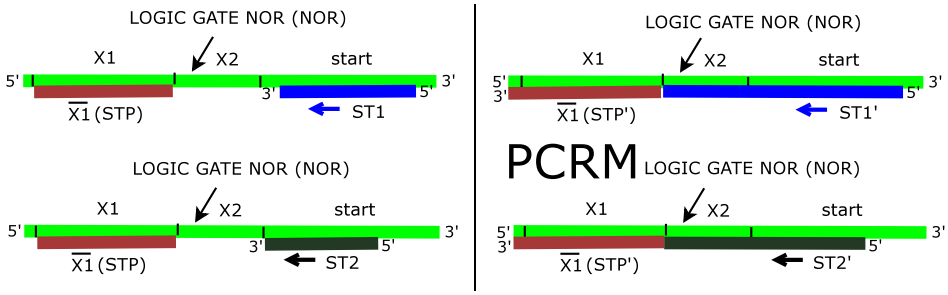

Figure 4. DNA molecules used in polimerase tests on the left, assumed results in polimerase tests on the right

Molecules can even conduct electrical current, which rises a hope for creating state solid molecular circuits. Experiments [11] showed that the resistivity values derived from

these measurements are comparable to those of conducting polymers and indicate that DNA transports electrical current as efficiently as a good semiconductor.

5 Molecular OR gate

A signal is represented by a DNA molecule called an oligo, an oligonucleotide, a strand, a DNA fragment. In [4] a signal equal to 0 is also a DNA chain and all permutations of zero and one input signals are encoded in DNA molecules together with zero and one output signals. In our approach a signal equal to zero is in a case, when a signal molecule is absent. It simplifies process of encoding gates (one molecule per one gate, not all input signal value permutations in many one gate copies, therefore it requires much more material to encode many input signals), but it enables only AND gates and OR gates.

Signals can be formed in a line [17]. In a process of firing an output signal any input signal $\overline{X_1}$ or $\overline{X_2}$ or $\overline{X_3}$ have to be present this means equal to one in order to create during amplification a single string containing Y_ANDOR DNA fragments. Each signal can be a primer in the amplification reaction (Fig. 2). The resulting strings can hybridize with a string $\overline{Y}$ and in the next amplification operation they are extended, but the string $\overline{Y}$ remains the same e.g. it can be achieved by encoding signal sectors in the line string with only three nucleotides and the rest of sectors with all four and making amplification with a help of only three complementary nucleotides to the mentioned earlier three nucleotides. The S operation degrades single strings and after a restriction enzyme cutting between sectors Y and Y_ANDOR the output signal Y is obtained.

6 Molecular NOR gate

The mentioned line string can be also a NOR gate molecule waiting for any input signal as a primer (Fig. 3). In this case any input inactive during amplification stops most activities of modified polymerase Taq and no output is generated.

7 Experimental results

The NOR gate from Fig. 4 is represented by a molecule NOR 61bp long. A primer ST1 (20bp) is during amplification lengthened up to a 59bp DNA fragment, a primer ST1 (17bp) is during amplification lengthened up to a 54bp DNA fragment, in the presence of STP (22bp) they should be lengthened respectively to 32bp or 27 bp fragments. But in real laboratory conditions they were lengthened to about 40bp molecule.

In Fig. 5 an electrophoretogram with results of experiments depicted in Fig. 4 with a modified new polimerase PCRM (N) and original polimerases PCR (L) was shown. Markers are denoted by M: M_1 is 31bp long, M_2 - 35bp, M_{19} - 17bp (ST2), M_{20} - 20bp (ST1), M_{21} - 22bp (STP), M_{22} - 56bp, M_{23} - 61bp (NOR), M_{24} - ST2,STP,NOR. In experiments different quantities of molecules were used: in lanes nr 3 and 4 - 10pmol of each molecule kind, in lanes nr 5 and 6 - 20pmol, in lanes nr 7 and 8 - 150 pmol, in lanes nr 9 and 10 - 10 pmol, in lanes nr 11 and 12 - 20 pmol, in lanes nr 13 and 14 - 150 pmol, in lanes nr 15 and 16 - 10 pmol of ST2, NOR and 100 pmol of STP, in lanes nr 17 and 18 - 10 pmol of NOR,STP, 100pmol of ST2. In lanes from 3 to 8 the start molecule ST1

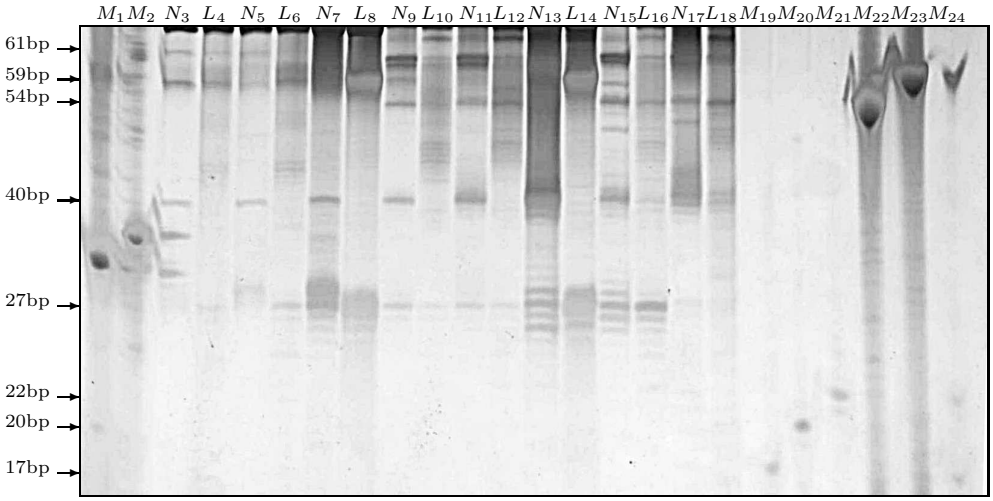

Figure 5. DNA electrophoretogram of the first experiment: M - marker lane, N, L - experiment lanes, but N with the new polimerase, 32 amplification cycles

was utilised and in lanes from 9 to 14 - ST2.

The obtained results confirm that the new polimerase use leads up to the unique DNA fragment (about 40bp long lengthened ST1 or ST2). In ideal conditions it should be about 27 bp long, but the modified polimerase managed to dehybridize a half of STP and stopped. Unfortunately, the undesired 59bp or 54bp bands are strong enough except lanes nr 5 and 16. To obtain more clear results in addition to already computer optimized DNA sequences the amplification process conditions should be optimized in order to remove mispriming artifacts which can add new bands in lanes (they are like a noise) and the polimerase should be much more modified by adding new abilities in the place of removed exonuclease activities. And for the complete NOR gate function the primer STP amplification activity should be completely removed e.g. by 3-amino-modified nucleotide chain terminators. It will not only stop firing of any another signal, but will not generate any signal itself.

8 Summary

Molecular computing in vitro is a rapidly developing research area with unknown to an end opportunities and challenges. However, it is sure that despite its embrionic state this methodology enables molecular level creation of new alternative computing architectures from which in future may emerge very powerful, massively parallel supercomputers or at least specialized lab-on-chip devices made for quick sequencing or cancer gene detection.

In this paper the process of implementing negation and OR gate together was introduced. The method of molecular negation still needs more improvements and experimental research, but the beginning is very promising. The molecular system engineer gains new level of molecule control.

In future the modified Taq polymerase can be applied in nanoconstructions and nanocomputations. Such the polymerase can lengthen appropriate primers for many cycles and nanodevices can be created one layer over another one without unwanted polymerase side effects e.g. destroying previous layers.

Bibliography

[1] Adleman, L.M. Molecular computation of solutions to combinatorial problems. *Science*, 266, 1021–1024, 1994.

[2] Amos, M., Dunne, P. E. DNA simulation of boolean circuits. Dep. of Comp. Sc., Univ. of Liverpool, 1997. *http://www.csc.liv.ac.uk/˜ctag/archive/t/CTAG-97009.ps.gz*

[3] Faulhammer, D., Cukras,A.R., Lipton,R.J., Landweber,L.F. When the Knight Falls: On Constructing an RNA Computer. In *Proc. of the 5rd DIMACS Workshop on DNA Based Computers, held at MIT*, 1999.

[4] Klein, J.P., Leete, T.H., Rubin, H. A biomolecular implementation of logically reversible computation with minimal energy dissipation. Biosystems. 52(1-3), 15-23, 1999.

[5] Lipton, R.J. Solution of Hard Computational Problems. Science 268, 542–545, 1995.

[6] Maniatis, T., Fritsch, E.F., and Sambrook, J. Molecular cloning: a Lab. Manual. Cold Spring Harbor Lab. Press, Cold Spring Harbor, NY, 1982.

[7] Michell, J.C., Yurke, B. DNA Scissors LNCS 2340, 258-268.

[8] Ogihara, M., Ray, A. Simulating boolean circuits on a DNA computer. Tech. Report TR 631, Univ. of Rochester, Comp. Science Department, 1996.

[9] Mulawka, J.J., Nowak, R., Wąsiewicz, P., Pluciennicz ak, A. Processing DNA tokens in parallel computing. In *Proc. Int. Parallel and Dist. Processing Symposium*, 2001.

[10] Sakamoto, K., Kiga, D., Komiya, K., Gouzu, H., Yokoyama, S., Ikeda, S., Sugiyama, H., Hagiya, M. State transitions by molecules. In *Proc. 4th DIMACS Workshop on DNA Based Computers*, USA, 1998.

[11] Sen, D., Fahkman, R.P. DNA conformational switches as sensitive electronic sensors of analytes and as components for logic gates. In *DNA Computing, 8th int. Workshop on DNA-Based Computers, Sapporo, Japan*, 2002.

[12] Simmel, F.C., Yurke, B. Operation of a Purified DNA Nanoactuator. LNCS 2340, 248-257.

[13] Smith, L.M., Corn, R.M., Condon, A.E., Lagally, M.G., Frutos, A.G., Liu, Q., Thiel, A.J. A surface-based approach to DNA computation. *Journal of Computational Biology*, 5, 1988.

[14] Stojanovic, M.N., Stefanovic, D. A Deoxyribozyme-Based Molecular Automaton. Nature Biotechnology 21, 1069-1074, 2003.

[15] Reif, J.H. The Design of Autonomous DNA Nanomechanical Devices: Walking and Rolling DNA. LNCS 2568, 22-37.

[16] Uejima, H., Hagiya, M., Kobayashi, S. Horn Clause Computation by Self-Assembly of DNA Molecules. LNCS 2340, 308-320.

[17] Wąsiewicz, P., Pluciennicz ak, A. Molecular Inference Network Experimental Approximation. In *Elektronika* 156, 405-412, 2006.

[18] Wasiewicz, P., Malinowski, M., Pluciennicz ak, A. Enzyme Modification for Next Level Molecular Computing. *Proc. of SPIE 2006*, Wilga, 6347, part 2.

[19] Wąsiewicz, P., Rudnicki, R., Mulawka, J.J., Lesyng, B. Adding numbers with DNA. In *Proc. 2000 IEEE Int. Conf. on Systems*, 2000.

[20] Wąsiewicz, P., Malinowski, A., Nowak, R., Mulawka, J.J., Borsuk, P., Weglenski, P., Pluciennicz ak, A. DNA computing: Implementation of data flow logical operations. *Future Generation Computer Systems*, 17/4, 361-378, 2001.