Another expert system rule inference based on DNA molecule logic gates

Piotr Wąsiewicz[a]

[a]Institute of Electronic Systems, Warsaw

ABSTRACT

With the help of silicon industry microfluidic processors were invented utilizing nano membrane valves, pumps and microreactors. These so called lab-on-a-chips combined together with molecular computing create molecular-systems-on-a-chips. This work presents a new approach to implementation of molecular inference systems. It requires the unique representation of signals by DNA molecules. The main part of this work includes the concept of logic gates based on typical genetic engineering reactions. 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. Every microreactor of the lab-on-a-chip performs one unique operation on input molecules and can be connected by dataflow output-input connections to other ones.

Keywords: molecular system, genetic engineering, logic gates, lab-on-a-chip, molecular-systems-on-a-chip

1. INTRODUCTION

Recent research results rise a hope for very great miniaturization in the field of numerous molecular electronic devices. A single electronic gate would be like a chemical macromolecule. These devices could be interfaces between silicon circuits and molecular intelligent databases similar to human brains. The lab-on-chip technology is also very promising in sequencing, and analizing data on molecular level.

The idea of molecular-systems-on-a-chips was first introduced by McCaskill [1, 2]. He used the miniaturization lab-on-a-chip methodology to solve in microreactors computational problems, which were previously computed in genetic engineering laboratory tubes [3–5].

Molecules e.g. DNA oligonucleotides carry information, and chemical reactions are like computing processes. Sequences of such processes are called DNA computing algorithms. DNA Computing research scientists focus on implementing algorithms solving not only NP-complete problems (nondeterministic-polynomial-time), but also executing logic gates and inference rules [6–13], adding binary numbers [14–21], constructing nanodevices (nanoscissors, tic-tac-toe automatons, nanorobots, nanotile assembly) in molecule nanoassembly process [22–28], implementing computation on the molecular surfaces [29–31].

McCaskill works [1, 2] started research and discussions, whether this joint technology is viable alternative to computers based on silicon electronics (very advanced, but with technology limits) and to molecular computers based on chemical reactions (in embryonic state, but with unknown and very promising future based on miniaturization and massive parallelism) [32, 33].

In this paper we propose a new approach to dataflow logic systems designed for a lab-on-a-chip methodology.

Further author information:
Institute of Electronic Systems, Warsaw University of Technology, Nowowiejska 15/19, 00-665 Warsaw, Poland;
E-mails: pwasiewi@elka.pw.edu.pl, pwasiewi@gmail.com

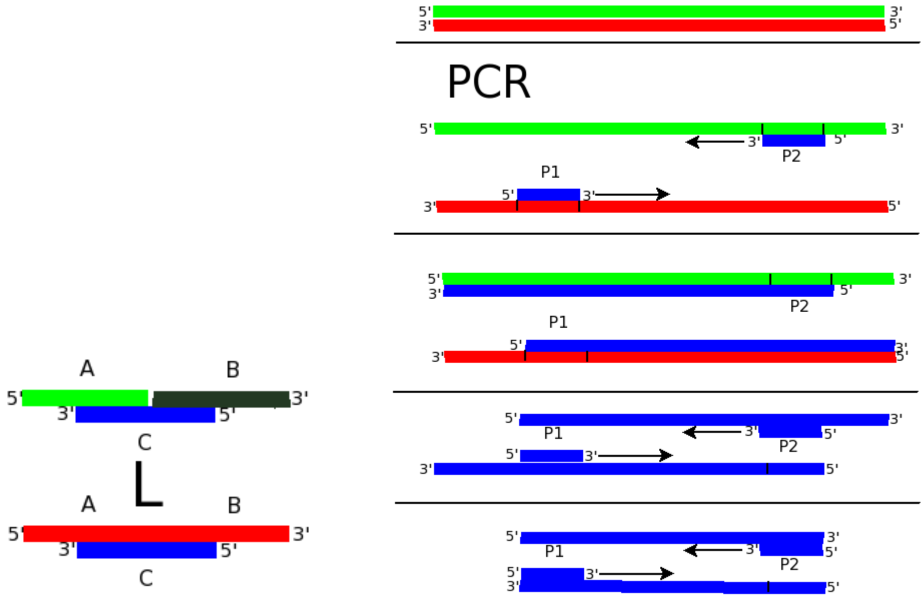

Figure 1. On the left: ligation of A and B oligos in the presence of hybridized third complementary one - C. On the right: the extension of primers' 3' ends in the process of PCR

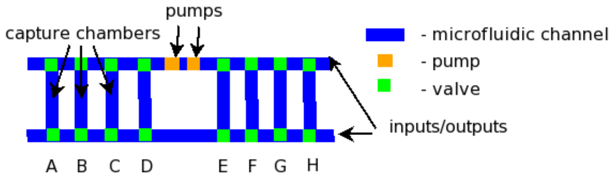

Figure 2. The microfluidic processor with valves, pumps, inputs and eigth capture chambers: $A - H$

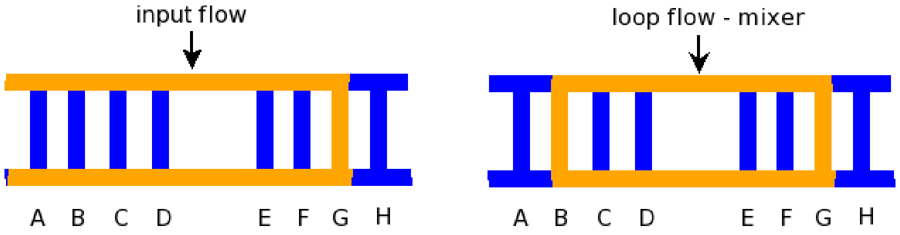

Figure 3. On the right the input solution flow. On the left the cycle flow in the microfluidic processor.

2. DNA COMPUTATION SYSTEMS

Molecules could be used as memory. Kashiwamura, Yamamoto, Kameda, Shiba, and Ohuchi described the memory based on nasted PCR in [34]. The paper [23] shows a possibility of building associative memory based on DNA strands. The method for suppressing DNA fragment amplification during PCR was used. Experimental construction of a very large scale DNA database is presented in [35]. Mills, Yurke, Platzman in [36] described a particular set of DNA operations to effect the interconversion of electrical and DNA data and to represent the Hopfield associative memory. This new type of DNA computing has the possible advantage of being fault tolerant and thus more immune to DNA hybridization errors than a Boolean DNA computer.

The molecule flow in mixtures can be automated within general-purpose DNA computers. McCaskill papers e.g. [1] described the microreactors implemented "in silico" with very small reaction nano chambers and input, output nanochannels. They could do make graph optimization on molecular level. Amenyo [37] showed that all proposed DNA computing algorithms can be run on parallel computer architectures configured from trellis/lattice banks, filter banks and switching banks. Thus, DNA computation can be re-interpreted as dataflow (or signal flow) networks and subject to conventional

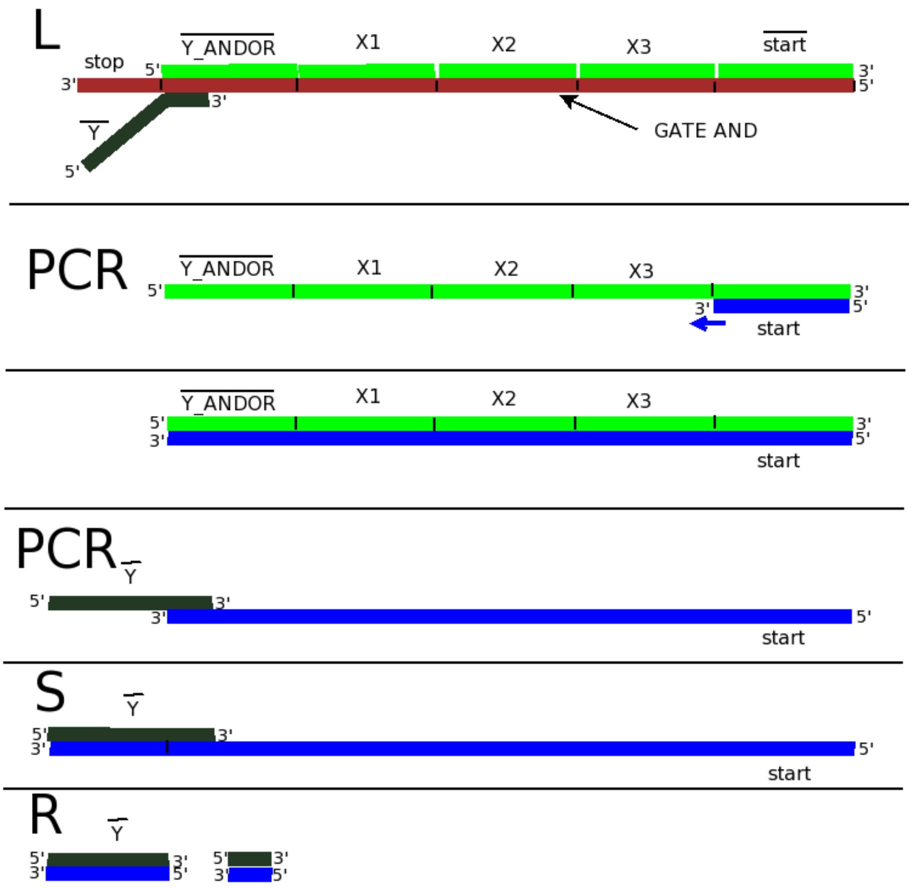

Figure 4. Molecular AND gate $X_1 \wedge X_2 \wedge X_3 \Rightarrow Y$

treatment.

The Warsaw research group in [23] described the finite state automaton based on PCR. This method is used for analyze DNA molecules whether they are described by specified regular expression. Presented ideas are confirmed by experiment performed in genetic engineering laboratory. Benenson, Paz-Elizur, Adar, Keinan, Livneh, and Shapiro [38] reported development of a programmable ’finite automaton’, using a restriction nucleaseand ligase as hardware and software consisting of transition rules encoded by DNA. Komiya, Sakamoto, Gouzo, Yokoyama, Arita, Nishikawa, and Hagiya [39] showed that a single-stranded DNA can serve as an independent machine by using a solid support technique in three experimental achievements in computation model based on ‘whiplash’ reactions, while Garzon, Gao, Rose, Murphy, Deaton, Franceschetti, and Stevens [40] used a ligation-based approach for in-vitro implementation of finite-state machines, which requires sequential input feed and different molecules for different machines. In their second implementation not based on ligation transitions are represented by reusable molecules and the input, coded as a molecule, can be introduced at once.

Molecules can exchange messages with each other. In paper [23] a new technique of sending data between molecular processors is presented. Its computation results have to be sent to other units in the form of addressed messages - tokens.

3. MOLECULAR LOGIC SYSTEM IMPLEMENTATIONS

Ogihara and Ray [9] first demonstrated that DNA computers can simulate Boolean circuits. Klein, Leete and Rubin [41] created universal three input logic gate based on PCR. Amos and Dunne [6] described the abstract model and its own laboratory implementation. Hagiya et al [10] designed one molecule DNA computer with data and operations on one DNA strand. Computation of logic function satisfiability was driven by PCR reaction. Wąsiewicz [11, 13] also proposed the evolutionary programming of logic function graphs, the evaluation of which is based on PCR.

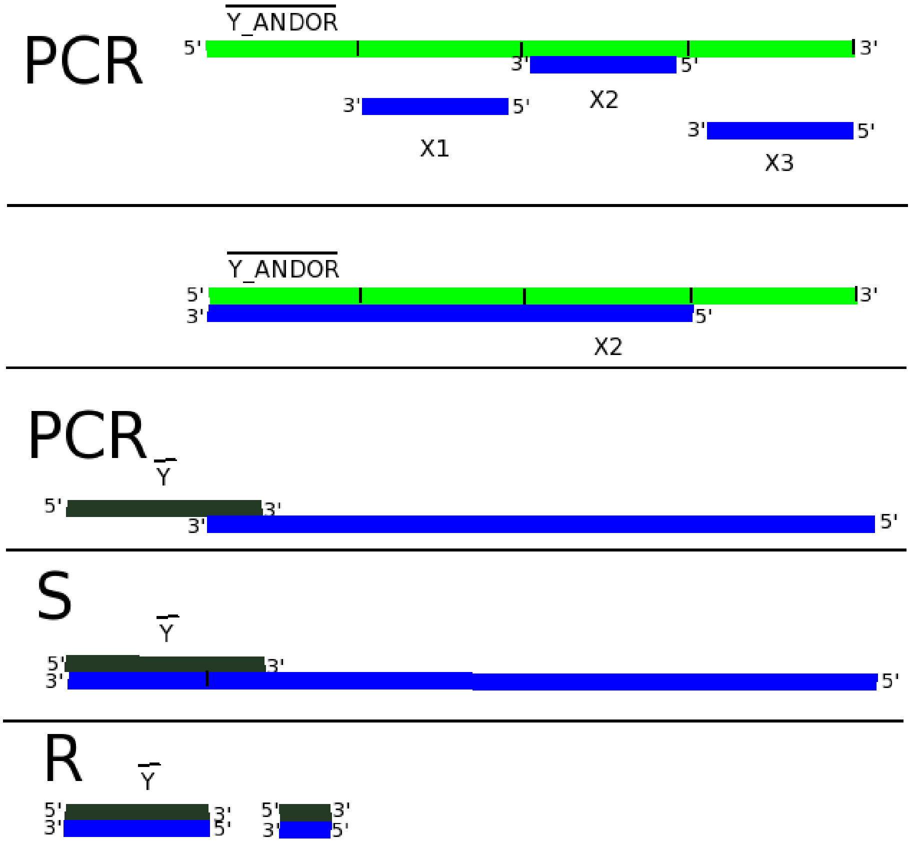

Figure 5. Molecular OR gate $X_1 \vee X_2 \vee X_3 \Rightarrow Y$

Surface-based methods were presented by Liu, Smith and their research group [29–31]. 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.

4. DNA MOLECULES AND OPERATIONS ON THEM

A double helix of DNA is made from two single strands of DNA oriented in opposite directions e.g. $T = ATGC$ and $\overline{T} = TACG$, each of which has two different ends 5', 3' and is a chain of four nucleotides Adenine, Thymine, Cytosine, Guanine denoted by the symbols A, T, C, and G, respectively, due to hybridization (annealing) reaction, because A is complementary with T, and C is complementary with G.

Oligonucleotides may connect with each other during concatenation process called ligation L to form longer DNA chains [42] as is described in Fig. 1 on the left. In order to amplify a target between predefined sequences (primers) a cycle of annealing-melting-extension operations, called Polymerase Chain Reaction (PCR), is utilized with the help of free nucleotides and DNA polymerase which duplicates DNA by adding complementary nucleotides to 3' ends of primers as is depicted in Fig. 1 on the right. One class of enzymes, called restriction endonucleases, recognize a specific short sequence of DNA, known as a restriction site and cut any double-stranded DNA at that location (an operation R). Exonucleases can degrade DNA molecules from the ends in (operation S).

Genetic operations driven by enzymes, heating and cooling, DNA sequence and a model make computation possible.

5. MOLECULAR-SYSTEMS-ON-A-CHIP

Significant progress has recently been made in microfluid devices utilizing nano membrane valves, pumps and microreactors. These devices are fabricated on silicon wafers using conventional photolithography. By combining lab-on-a-chips

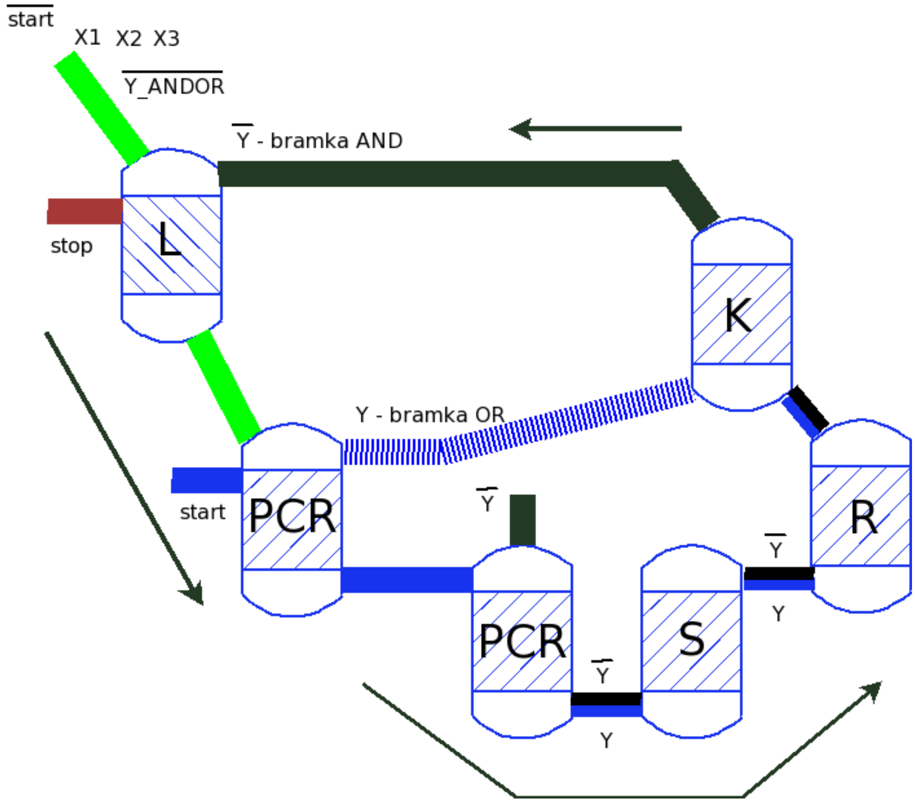

Figure 6. A lab-on-a-chip scheme

with DNA computing evolves a new area of research so called molecular systems-on-a-chip [1, 2, 32, 33].

A McCaskill's on-chip system for serial capture and release of DNA made of H-shaped channel junctions called "selective transfer modules" was a prototype [1] of actual microluidic DNA computers. One of two neighboring channels contains input DNA in hybridization buffer and the other contains an alkaline solution with the high pH. Despite the DNA and alkaline streams contacting each other the laminar flow profile limits cross-contamination between the streams. Magnetic beads derivatized with capture oligonucleotides are held by an external magnet in DNA input stream. After capturing DNA strings with correct, perfectly complementary sequence of nucleotides, the beads are transferred magnetically into the alkaline stream, where the high pH destabilizes the DNA duplexes and releases the captured DNA into the alkaline stream downstream mixing with fresh hybridization buffer neutralizes the alkaline and makes selected DNA ready for additional capture/release steps. After a series of capture/release steps the remaining DNA represents the correct solution. Initial population encoding all possible answers to the given problem was processed in a single pass through the transfer modules. The most complex DNA computation performed to date was solved using 300 base pair input DNA containing constant 15-base sequences for each of 20-binary bits [4].

Although new solutions of not very good efficiency were proposed e.g. "negative selection", where the alkaline streams are removed, only an invention of monolithic elastomer membrane valves and pumps suitable for large-scale integration into glass microfluidic analysis devices gave a chance to improve capture efficiency, to reduce hybridization time from typical 4 hours to minutes and replace constant multi-base capture sequences with single nucleotides. The air or fluid driven pumps and valves enables programmable autonomous solution flow and mixing in closed channel sectors [32, 43]. Now is it possible to reuse the same channel several times utilizing different reactions every time. Solution with information can be cycled, processed and stored in separated channels.

An exemplary microfluidic processor [32] is described in Fig. 2. It has eight reaction chambers containing magnetic beads with capture oligos. Chambers can be open and closed at both sides with the help of microvalves. Pumps in the middle makes solution flow. An input DNA flow is shown on the left in Fig. 3. After capturing of input signals in the

G chamber, valves near B chamber are closed and solution can flow from the G chamber to the B chamber, where it is captured by other DNA oligos on the beads. After a series of capture/release operatons the correct solution remains in the final chamber. All pumps and valves can be controlled by a computer program [43].

6. THE ON-A-CHIP INFERENCE SYSTEM

A signal is represented by a DNA string called an oligo, an oligonucleotide, a strand, a DNA fragment. In [41] a signal equal to 0 is also a DNA string 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 absent. It simplifies process of encoding gates (one molecule per one gate, not all input signal value permutations in many one gate copies, therefore much more material to encode many input signals), but it enables only AND gates and OR gates.

Signals can be formed in a line [44]. In the process of concatenation L all input signals $\overline{X_1}, \overline{X_2}, \overline{X_3}$ have to be present this means equal to one in order to create a single string containing $\overline{start}$ and $\overline{Y_A NDOR}$ DNA fragments. In the next PCR process of extending an added in surplus primer *start* this line string is necessary for amplification to proceed up to its end as is depicted in Fig. 4. In the process of selection G e.g. in electrophoresis or by magnetic beads the all new generated strings with the builtin *start* primer are separated. The resulting strings can hybridize with a string $\overline{Y}$ and in the next PCR operation the same line string from previous operations is extended, but the string $\overline{Y}$ remains the same e.g. it can be achieved by encoding a line string with only three nucleotides and making PCR only in the presence of these three nucleotides. The S operation degrades single strings and after a restriction enzyme process cutting between sectors Y and $Y_A NDOR$ the output signal Y is obtained.

The mentioned line string can be an OR gate molecule waiting for any input signal as a primer (Fig. 5). There is no *start* primer in the OR gate and there is no standard signal representation in the AND gate and this is the end of differences between molecular OR and AND gates as is shown in a microfluidic processor scheme (Fig. 6), where every microreactor is connected with one molecular operation. The single output string Y flows to OR gates and the $\overline{Y}$ flows to the AND gates. Execution is possible for only one type of gates at one cycle in this MSOC. This is the inference process of rules made of molecular premises and conclusions. After some cycles molecular solution will be obtained in a final chamber.

7. SUMMARY

From fundamental self-assembly of basic elements, new logic systems could emerge through implementing information techniques among others inference systems, neural structures, binary operations [19–21] on nano scale. A gap between implementation and theory of computer science should be omitted in the mentioned way leading to new architectures of based on polimers or peptides specialized processors computing huge data amounts.

In future, in one reaction chamber the modified polymerase can lengthen appropriate primers for many cycles and nanodevices can be created one layer over another one without unwanted polymerase side effects.

With latest achievements it would be possible to invent new paradigms, more effective algorithms and with a help of all previous methods to make autonomous biomachines, which can contain even living cells as processing units with implemented selfmodification.

Enzymes drive molecular computation processes and these processes can change their properties. Information data can circulate much longer. Of course, quicker process should be achieved in vivo or in synthesized living cells, but it is beyond present technological limits.

In further research original nanoconstruction systems with the help of modified enzymes should be proposed.

References

[1] D. van Noort, F. Gast, J. S. McCaskill. DNA Computing in Microreactors. LNCS 2340, 33-45.

[2] Wagler P. F., Tangen U., Maeke T., Chemnitz S., Juenger M., McCaskill J. S., Varadan V. K.: Molecular systems on-chip (MSoC) steps forward for programmable biosystems, *Proc. of SPIE. Smart Structures and Materials 2004: Smart Electronics, MEMS, BioMEMS, and Nanotechnology* 5389, 2004, 298-305.

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

[4] Braich R. S., Johnson C., Rothemund P. W. K., Chelyapov N., and Adleman L. M.: Solution of a 20-variable 3-SAT problem on a DNA computer. Science, 296(5567), 2002, 499-502.

[5] Ouyang Q., Kaplan P. D., Liu S., Libchaber A.: DNA solution of the maximal clique problem. *Science*, 278, 1997.

[6] Amos M., Dunne P. E.: DNA simulation of boolean circuits. Technical Report CTAG-97009, Department of Computer Science, University of Liverpool, UK, 1997. *http://www.csc.liv.ac.uk/ ctag/archive/t/CTAG-97009.ps.gz.*

[7] Kobayashi S., Yokomori T., Sampei G., and Mizobuchi K.: DNA implementation of simple Horn clause computation. *IEEE International Conference on Evolutionary Computation*, 1997.

[8] Mulawka J.J., Borsuk P., and Weglenski P.: Implementation of the inference engine based on molecular computing technique. In *Proc. IEEE Int. Conf. on Evolutionary Computation, Anchorage, USA*, 1998.

[9] Ogihara M., Ray A.: Simulating boolean circuits on a DNA computer. Technical Report TR 631, University of Rochester, Computer Science Department, August 1996.

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

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

[12] Wąsiewicz P., Janczak T., Mulawka J.J., Plucienniczak A.: The Inference Based on Molecular Computing. Int. Journal of Cybernetics and Systems 31/3, 2000, 283-315.

[13] P. Wąsiewicz, J.J. Mulawka. Molecular Genetic Programming. Soft Computing, Springer-Verlag 5(2), 2001:106-113.

[14] Barua R., Misra J.: Binary arithmetic for DNA computers. Poster paper at 8th International Workshop on DNA-Based Computers, DNA 2002, Sapporo, Japan, 2002.

[15] Guarnieri F., Fliss M., and Bancroft C.: Making DNA add. *Science*, 273, 1996.

[16] Gupta V., Parthasarathy S., and Zaki M. J.: Arithmetic and logic operations with DNA. In *Proceedings of the 3rd DIMACS Workshop on DNA Based Computers, held at the University of Pennsylvania*, 1997.

[17] Hug H., Schuler R.: DNA-based Parallel Computation of Simple Arithmetic. LNCS 2340, 321-328.

[18] Oliver J. S.: Computation with DNA-matrix multiplication. In *Proceedings of the Second Annual Meeting on DNA Based Computers, held at Princeton University*, 1996.

[19] P. Wąsiewicz, R. Rudnicki, J.J. Mulawka, and B. Lesyng. Adding numbers with DNA. In *Proceedings 2000 IEEE International Conference on Systems*, 2000.

[20] P. Wąsiewicz, Subtraction of binary numbers using biochemical reactions on DNA molecules, Polish Patent no 212785, Patent Office of the Republic of Poland, 2012.

[21] P. Wąsiewicz, Addition of binary numbers using biochemical reactions on DNA molecules, Polish Patent no 214232, Patent Office of the Republic of Poland, 2013 .

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

[23] J.J. Mulawka R. Nowak, P. Wąsiewicz and A. Plucienniczak. Processing DNA tokens in parallel computing. In *Proc. International Parallel and Distributed Processing Symposium*, 2001.

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

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

[26] Rothemund P.W.K.: Theory and experiments in algorithmic self-assembly. *Dissertation for PhD degree*, University of Southern California, 2001.

[27] P. Sa-Ardyen, N. Jonoska, N.S. Seeman. Self-assembling DNA Graphs LNCS 2568, 1-9.

[28] P. Wąsiewicz, A. Dydynski, G. Tomczuk, J.J. Mulawka, A. Plucienniczak. Molecular Neuron Realization. WSEAS Transactions Journal on Biology and Biomedicine, 1(1):73-75.

[29] Liu Q., Frutos A. G., Wang L., Thiel A. J., Gillmor S. D., Strother T., Condon A. E., Corn R. M., Lagally M. G., and Smith L. M.: Progress toward demonstration of a surface based DNA computation: a one word approach to solve a model satisfiability problem. In *Proceedings 4th DIMACS Workshop on DNA Based Computers, held at the University of Pennysylvania*, 1998.

[30] Liu Q., Thiel A. J., Frutos A. G., Corn R. M., and Smith L. M.. Surface-based DNA computation : Hybridization and destruction. In *Proceedings of the 3rd DIMACS Workshop on DNA Based Computers, held at the University of Pennsylvania*, 1997.

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

[32] Grover W.H.: Microfluidic Molecular Processors for Computation and Analysis. PhD thesis, University of California, Berkeley, 2006.

[33] Wang Y.: Modeling and Simulation of Lab-on-a-Chip Systems. PhD thesis, Carnege Mellon University, Pittsburgh, 2005.

[34] S. Kashiwamura, M. Yamamoto, A. Kameda, T. Shiba, and A. Ohuchi. Hierarchical DNA memory based on nested PCR. Poster paper at 8th International Workshop on DNA-Based Computers, DNA 2002, Sapporo, Japan, 2002.

[35] J. H. Reif, T. H. LaBean, M. Pirrug, V. S. Rana, B. Guo, C. Kingsford, and G. S. Wickham. Experimental construction of a very large scale DNA database with associatice search capability. In *DNA Computing, 7th international Workshop on DNA-Based Computers, Tampa, U.S.A.*, 2001.

[36] A. P. Mills Jr. and B. Yurke and P. M. Platzman. DNA Analog Vector Algebra and Physical Constraints on Large-Scale DNA-Based Neural Network Computation. In *Proceedings 5th DIMACS Workshop on DNA Based Computers, held at the Massachusetts Institute of Technology*, 1999.

[37] J. Amenyo. Mesoscopic computer engineering: Automating DNA-based molecular computing via traditional practices of parallel computer architecture design. In *Proceedings of the Second Annual Meeting on DNA Based Computers, held at Princeton University*, 1996.

[38] Y. Benenson, T. Paz-Elizur, R. Adar, E. Keinan, Z. Livneh, and E. Shapiro. Programmable and autonomous computing machine made of biomolecules. *Nature*, 414, 2001.

[39] K. Komiya, K. Sakamoto, H. Gouzo, S. Yokoyama, M. Arita, A. Nishikawa, and M. Hagiya. Successive state transitions with i/o interface by molecules. In *DNA Computing, 6th international Workshop on DNA-Based Computers, Leiden, The Netherlands*, 2000.

[40] M. Garzon, Y. Gao, J. A. Rose, R. C. Murphy, R. Deaton, D. R. Franceschetti, and S. E. Stevens Jr. In-vitro implementation of finite-state machines. in *Proc. 2nd Intl. Workshop on Implementing Automata WIA'97*, 1998.

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

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

[43] Urbanski J.P., Thies W., Rhodes C., Amarasinghe S., Thorsen T.: Digital microfluidics using soft lithography. *Lab Chip*, 2006, 6, 96-104.

[44] Wąsiewicz P., Plucienniczak A.: Molecular Inference Network Experimental Approximation. In *Elektronika* 156, Evolutionary Computation and Global Optimization 2006, 405-412.