

Research Article

A Hybrid DNA-computer's Structure Proposal

Alexander S. Polenov

Perm National Research Polytechnic University, Komsomolsky Avenue, 29, Perm, 614990, Russia

Abstract: This study describes the possible structure of hybrid DNA-computer. Hybrid DNA-computer is a hybrid of traditional binary electronic computer and DNA-computer. DNA-computer is a computer that uses DNA-molecules to perform the computations instead of silicon chips. Traditional part of hybrid computer is supposed to control the DNA-part. The hybrid DNA-computer combines the power of modern electronic and DNA-computers. This computer provides the convenience and capabilities in solving problems that requires the brute-force search. The suggested structure of DNA-part is scalable and reusable. DNA-part is controlled by special electronic microcontrollers, which are controlled by traditional electronic computer. Electronic computer can keep some intermediate data while DNA-part is operating. The hybrid computer is a universal machine and it can compute a wide variety of problems. Another advantage of hybrid computer is a software that could be developed in familiar way. Hybrid computer can provide an API, which could be used in regular programming languages. It will be sufficient to import some library into a project and the apps will be able to use advantages of both electronic and DNA-computers. Of course, creating of such computers could give additional difficulties, but the model suggested in this study could be a good starting point for hybrid computer systems. Here are the structure and the common principles of DNA-part in this study. The suggested structure is scalable and could be modified to work with DNA-memory complexes of any length, memory unit also could be extended.

Keywords: Bio computer, DNA-computations, DNA-logic, hybrid computer, molecular computer

INTRODUCTION

DNA-computations is an interdisciplinary field of knowledge. The subject of this field is creating the computers that uses Deoxyribonucleic Acid (DNA) molecules instead of silicon chips. There is no common way how to use DNA-molecules for computations. Different researchers proposed different models and each model has own pros and cons.

Adleman (1994) performed the first experiment in 1994. He solved the problem of search the Hamiltonian path in graph using the DNA molecules.

Later another researcher Lipton (1995) has extended the Adleman's approach to solve Boolean Satisfiability Problem using DNA-molecules.

A sticker-based model for DNA-computations (Roweis *et al.*, 1996) was proposed in 1996. It was one of the most promising model.

Another notable model is autonomous computing machines made of biomolecules proposed by Benenson *et al.* (2001).

DNA-molecules also were used to build self-assembly structures, such as Sierpinski Triangles (Rothemund *et al.*, 2004). Later even 3-Dimensional structures were built by self-assembly of DNA-bricks (Ke *et al.*, 2012).

The important step is appearance of DNA computing circuits. Some of them uses enzymes (Elbas *et al.*, 2010) and some of them are enzyme-free (Qian and Winfree, 2011).

To date most of researchers efforts are directed to create DNA-computers where all computations must be performed by DNA-molecules. It is acceptable to call these DNA-machines as pure DNA-computers. However, the combining of DNA-computers with electronic computers may provide additional advantages.

In this study, a hybrid DNA-computer idea are explained. A sticker-based model mentioned above was extended to improve the precision of operations. Principles of this model are used in the DNA-part of hybrid computer. The aim of this study is developing of the structure of DNA-part and explaining of the principles of operating of every unit in DNA-subsystem. Another important thing is discussing of the advantages of hybrid system in comparison with stand-alone DNA-computing systems, which were developed earlier.

A hybrid computer idea: A hybrid DNA-computer is a computer that combines both DNA-computer and traditional binary electronic computer. The DNA-part is used to compute tasks via brute-force search. The

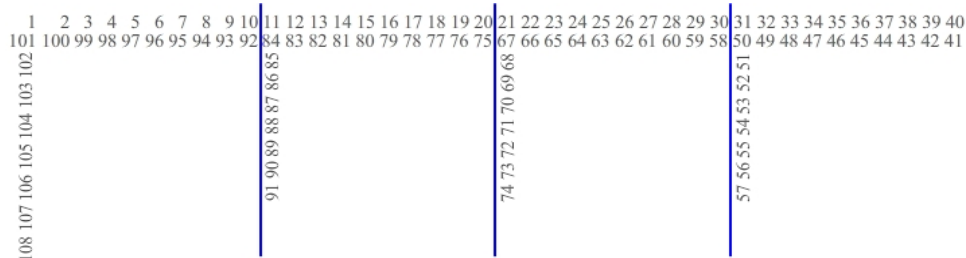


Fig. 1: Design of memory complex of extended sticker-based model with 4 stickers

traditional part is used to control the DNA-part, to keep constants and to perform additional computations.

The offered structure of DNA-part uses the sticker-based model for DNA-computation (Roweis *et al.*, 1996). However, it was improved by adding the mechanics of toehold-mediated strand displacement.

Extended sticker-based model for DNA-computations: Extended model can perform the same operations as the standard sticker-based model does: Combine, Separate, Set, Clear, Initialization, Detection.

The design of memory complexes is almost the same, but the stickers are slightly longer than their pads on the memory strand (Fig. 1).

Each sticker has 10 nucleotides that are used for pairing with memory strands and 7 free nucleotides. These 7 free nucleotides is necessary for implementation of toehold-mediated strand displacement during Clear operation.

To perform the Clear operation is sufficient to add to solution the DNA-molecules, which are completely complementary to one of the stickers. One can call these molecules as clearers or anti-stickers. Anti-sticker has 17 nucleotides, which are complementary to corresponding nucleotides of the sticker. So the sticker will be moved away from memory complex by toehold-mediated strand displacement (Cheng *et al.*, 2013).

Free nucleotides of the sticker are also used to perform the Separate operation. Memory complexes that have certain stickers can be captured by DNA-probes, which are complementary to the last 7 free nucleotides of this sticker. DNA-probes can be connected to DNA-microarray and this DNA-microarray can be moved from solution to another tube when complexes were attached to DNA-probes. The complexes, which haven't this certain sticker, still will be into initial solution. The separation completed.

Other operations in extended sticker-based model work the same way like in standard sticker-based model for DNA-computations.

MATERIALS

Structure scheme of DNA-part: Physically the DNA-part is a set of tanks connected with pipes. Tanks and pipes could be made of acrylic glass. All the system is filled with PBS buffer.

Structure scheme is shown at Fig. 2.

DNA-molecules (memory complexes, stickers) can only move in the directions shown by arrows. Solution can move in any direction. The system has a pump that can move the solution in both directions. The current solution route (circle) depends on opened and closed valves. Each pipe's and tank's connection has a special valve.

Every unit of the system and its operating principles will be described below.

Valve system: Each unit's input and output has a valve. Pipes crossroads also have a valve. An electronic microcontroller controls the valve system.

Valve could be into one of three states:

- **Completely closed:** Valve does not let anything to flow. Both solution and molecules are blocked.
- **Opened for solution:** Valve lets the solution to flow but any molecules can't pass this valve.
- **Opened:** Solution and molecules can flow through this valve.

Some units have special valves that let only stickers to flow in opened state. Memory complexes cannot pass these valves at all.

Good option for valves is a solenoid valve. System has a controller that opens and closes certain valves by changing the voltage. At every moment of computation only the one route for solution exists, that's why there is a few amount of opened valves at one certain moment. All valves should be completely closed when the system is shut down.

Separators: DNA-part has a separators unit to perform *Separate* operation. This unit could include one or more individual separators depending on DNA-part scale. One of separators is shown at Fig. 3.

One separator consists of two tanks. The first one has input for memory complexes. When the memory complexes are inside, then DNA-microarray with corresponding DNA-probes goes down inside the tank and catches memory complexes with corresponding stickers. DNA-probes will be paired with last 7 free nucleotides of stickers as was described in below section. When the memory complexes are connected with DNA-microarray, it goes up from the first tank,

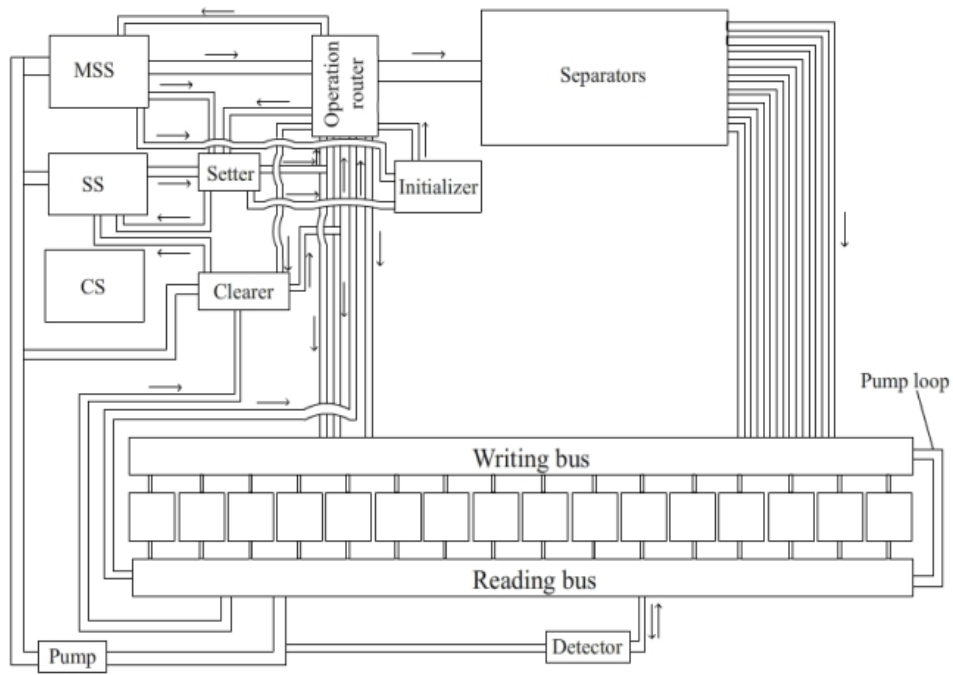


Fig. 2: Structure of DNA-part of hybrid computer

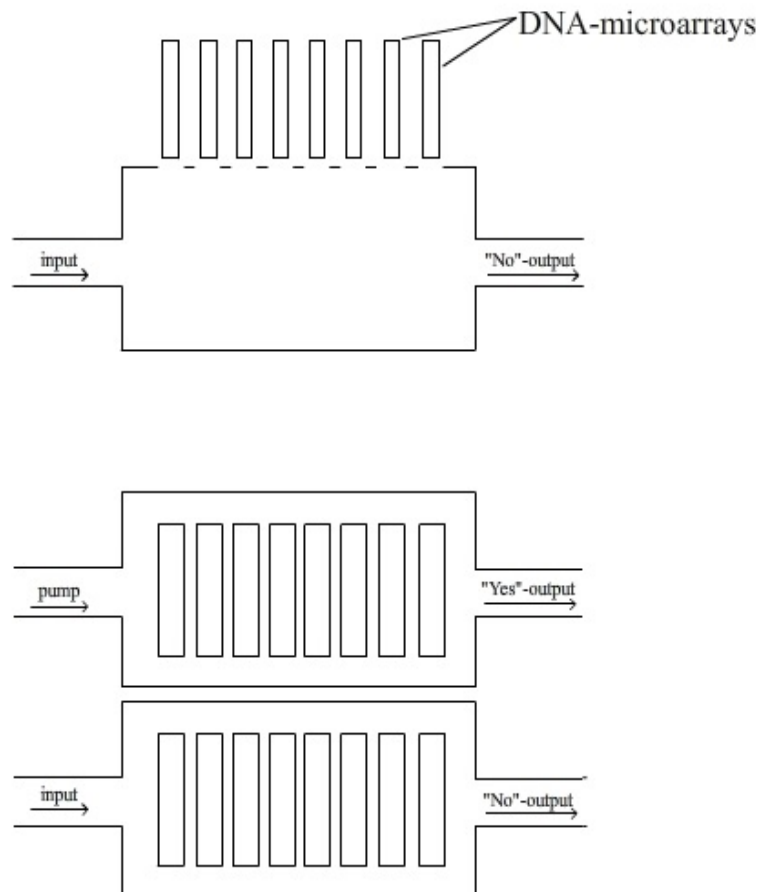


Fig. 3: Separator. A view from side (at the top) and a view from top (at the bottom)

moves and goes down to the second tank. The second tank is heated and complexes become able to leave the DNA-probes. It is important to not overheat the solution in this tank, because the excess heating could disturb the stickers and make them move away from memory complexes. It is highly recommended to use heatable DNA-microarrays to help DNA-probes leave the stickers without significant heating of solution.

At this point we have memory complexes that haven't specific sticker inside the first tank and complexes that have this specific sticker inside the second tank. Separation completed. Now the memory complexes with or without conceived sticker could be removed from separator independently through "Yes"-output and "No"-output. Pumping should be applied to the input of first or second tank respectively to make complexes move.

Only one separator could be used to perform any separation. However, if the length of memory complexes is large enough then it will be good idea to have several separators to work with specific stickers. For example, the first separator can perform separation only by one of first 8 stickers, the second separator can perform separation only by one of second 8 stickers and etc. This approach could save a time, because corresponding DNA-microarrays will be near the corresponding separators and their ways will not be too long.

Separators unit has twice as many outputs than the number of individual separators. If the number of separators is big enough, then all "Yes" and "No" outputs could be combined in two pipes. Anyway, only one *Separate* operation can be performed inside separators unit at the one moment of time.

The separators unit at Fig. 2 has no combining of "Yes" and "No" outputs.

Setter: Setter unit performs Set operation. Setter unit is pretty simple. It has two inputs: one for memory complexes and one for stickers. Resulting memory complexes can escape the setter through output. Of course, some stickers might not be paired with memory strands and it is necessary to bring them back to their storage through stickers exit. Stickers exit valve has a sticker filter, so only stickers can move through this valve when it is in Opened state. Memory complexes should not be able to pass this valve in any situation.

Only one type of stickers might be set inside setter at one certain moment of time. It is necessary to move the rest of stickers to their own cell in the storage. Setter unit scheme is shown at Fig. 4.

Clearer: Clearer unit performs the *Clear* operation. It works a little bit similar with separators unit. Clearer unit consists of two tanks (Fig. 5).

Both tanks have a pump input. The first tank accepts memory complexes through complexes input. Afterwards, DNA-microarray with anti-stickers as DNA-probes goes down into the tank. Each anti-sticker are completely complementary to all 17 nucleotides of specific sticker. Anti-sticker can remove the specific sticker via toehold-mediated strand displacement.

When the stickers have been removed from memory complexes, DNA-microarray with sticker/anti-sticker pairs goes up from the first tank. Resulting memory complexes now can leave the clearer tank through output. Then DNA-microarray goes down to the second tank of clearer. The solution in the second

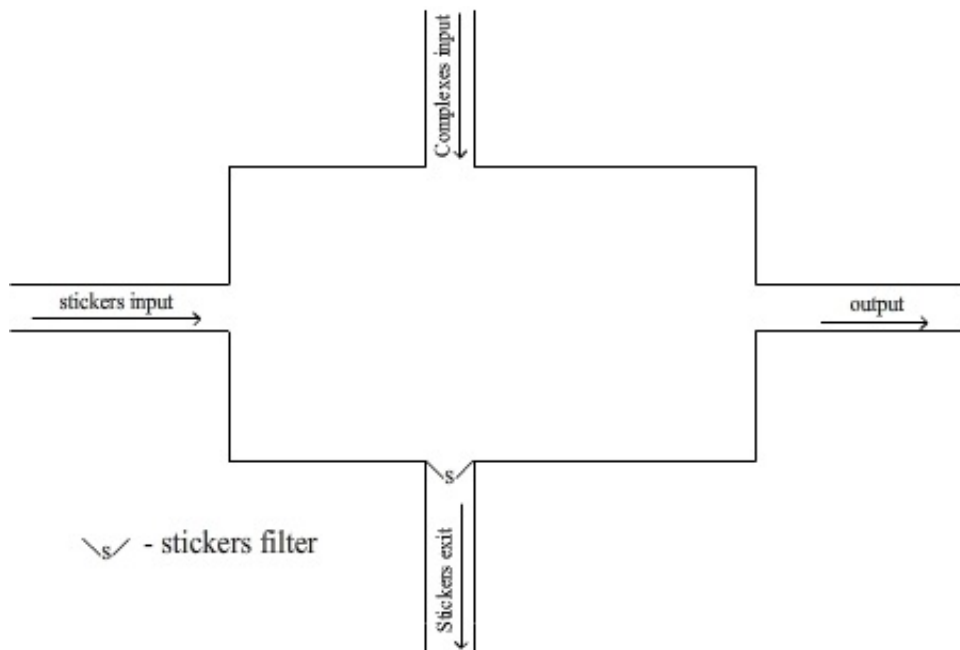


Fig. 4: Setter unit scheme

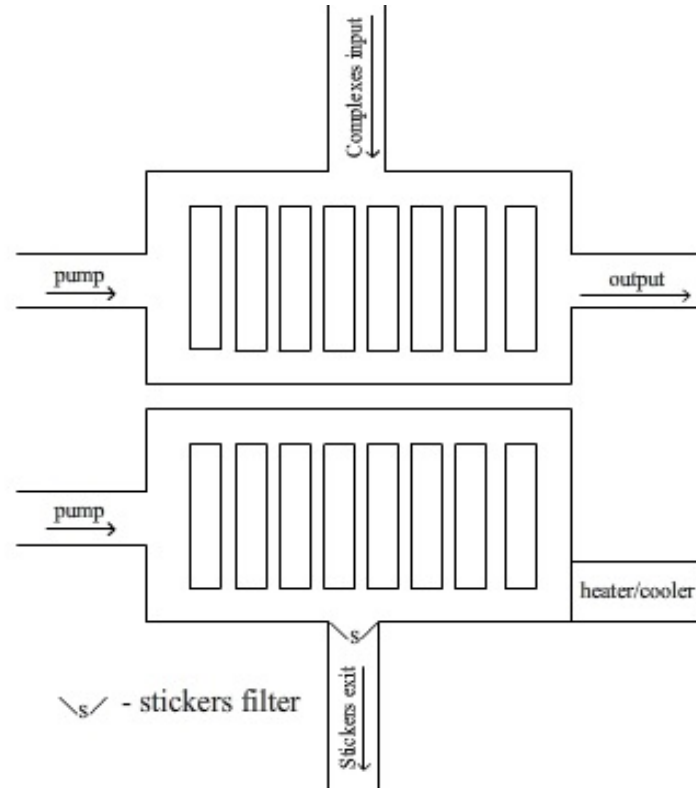


Fig. 5: Clearer view from top; Both tanks have a pump input

tank is heated and stickers become free from anti-stickers DNA-probes. The DNA-microarray with free probes leaves the tank. Solution is cooled down and stickers must be returned to storage through stickers exit in second tank. Only stickers can move through stickers exit as it was in setter.

As it works in setter, only one type of stickers can be cleared at one certain moment of time in the clearer.

Operation router: Operation router is a unit that reroutes memory complexes to other units of DNA-part. Operation router is a central unit of DNA-part. It has inputs from Memory Strands Storage (MSS), from setter, from clearer and from initializer. It also has access to direct writing and reading to/from DNA memory unit. Operation router has outputs to separators, to setter, to clearer and to MSS.

To reroute memory complexes, one of inputs of operation router should be *Opened for solution* and one of outputs should be in *Opened* state. Memory complexes will be washed away to the target unit of DNA-part. Operation router is shown at Fig. 6.

Random access memory: DNA-part has a memory unit that is a little bit similar with traditional Random Access Memory (RAM) from electronic binary computers.

Memory unit consists of two tanks named writing bus and reading bus respectively and memory cells that are situated between these two tanks (Fig. 7).

Memory has 3 modes:

- **Writing:** Set of memory complexes appears in writing bus. Then top valve of memory cell is switched to *Opened* state and bottom valve is switched to *Opened for solution* state. Therefore, memory complexes will flow into the cell. One set of memory complexes must be completely written to one cell, that's why only one cell can be opened to writing at any particular moment of time.
- **Reading:** Top valve of memory cell is switched to *Opened for solution* state and bottom valve is switched to *Opened* state. Memory complexes will be washed away from memory cell to reading bus. Reading only from one memory cell is allowed of any particular moment of time.
- **Bypass mode:** Memory unit is always on the solution's way. Therefore, when there is nothing to write or read in the memory and all memory cells are closed, solution must still flow through memory unit. To let this happens there is a pump loop in the memory unit. In bypass mode both valves of pump loop are in *Open for solution* state. Pump loop valves are always in *Closed* state while writing or reading.

A set of memory complexes is often need to be saved after *Separate* operation. That is why separators unit has direct access to writing bus of memory. Setter

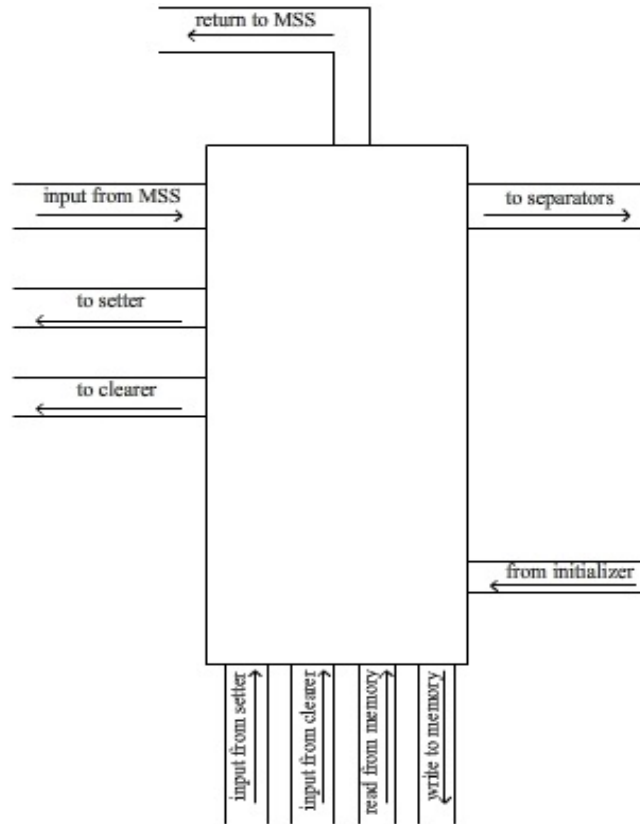


Fig. 6: Operation router

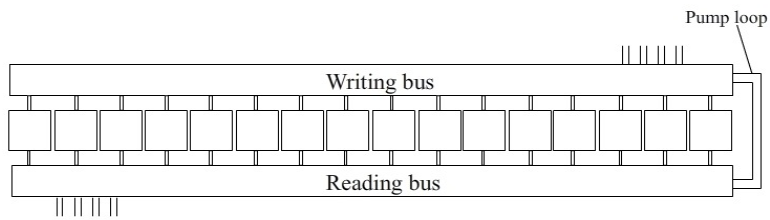


Fig. 7: Memory unit

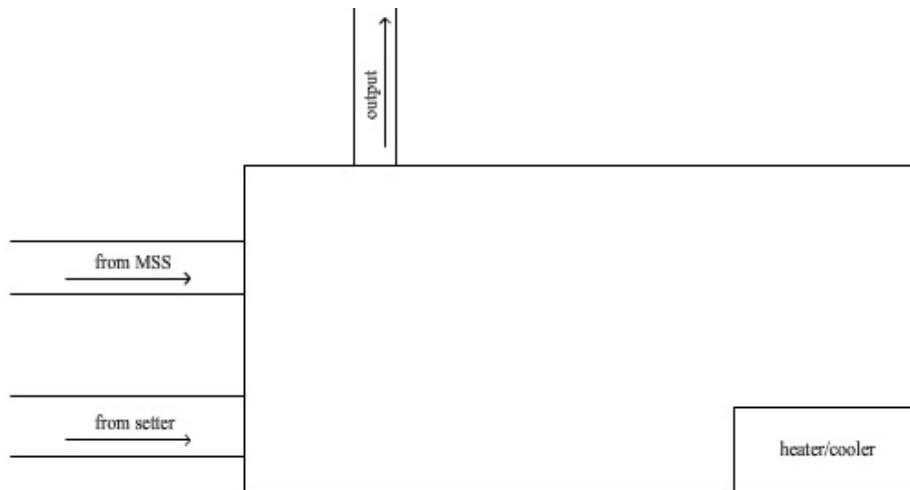


Fig. 8: Initializer

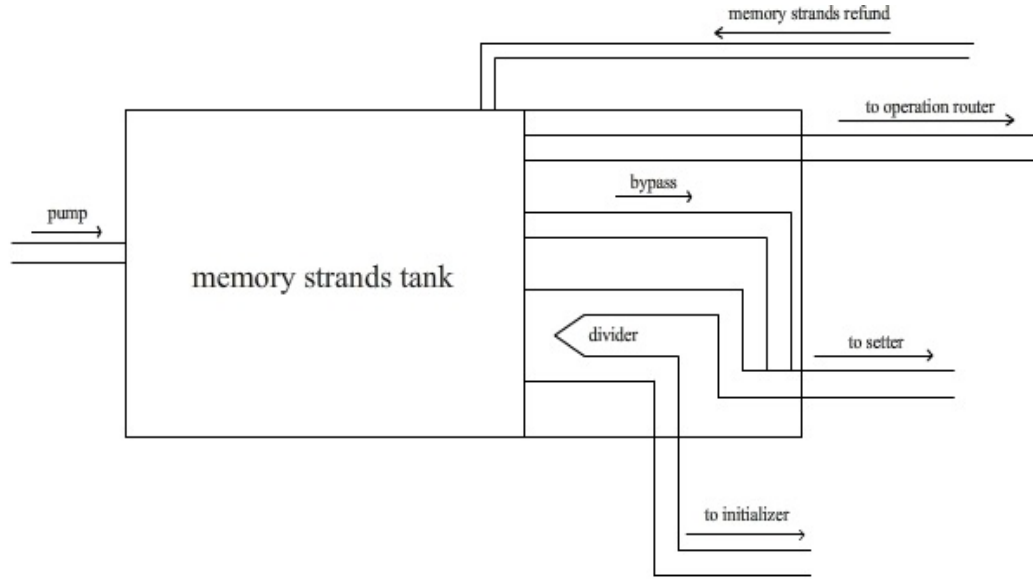


Fig. 9: Memory Strands Storage (MSS)

and clearer also have direct outputs to writing bus of memory. Direct access to writing bus allows to reduce the path for memory complexes that helps to reduce losses of memory complexes during computations.

Reading bus has outputs to operation router, to detector and to clearer. To restore the DNA-system to its initial state, all memory complexes should be cleared from stickers sequentially, that's why the reading bus of memory has direct link to clearer.

Initializer: Initializer is unit in DNA-part where random data filling occurs. For example, it is necessary to randomly fill first 4 bits. In this case, to get all possible sets of data with length equal 4, we should divide all memory complexes into two sets. The one half of these complexes should approach from memory strands storage without any stickers at these 4 positions and the second half of these complexes should approach from setter with a sticker in all positions within this 4 bits. Afterwards, the solution with memory complexes will be heated and stickers will become free from memory strands. After that, the solution will be cooled again and stickers will be paired with memory complexes completely randomly. Ideally, it should produce every possible set of data only once. However, in real world, of course, there might be some duplicates and missing sets of data, but the amount of these mistakes should not be catastrophically large (Roweis *et al.*, 1996).

It is easy to assume that initializer should have two inputs: one from setter and one from memory strand storage. It also should have an output to operation router and a heating/cooling system (Fig. 8).

DNA-part is designed to solve brute-force search tasks, that is why the initialization should always takes place at very beginning of computations, therefore,

there are no stickers at any position, except positions that we want to initialize. No data will be corrupted while heating and cooling inside initializer, because there is no data saved in memory complexes yet.

Memory Strands Storage (MSS): Memory Strands Storage (MSS) is a place where memory strands stored before and after computations. All memory strands have no stickers while stored here.

MSS consists of memory strands tank and pipe system (Fig. 9).

Memory strands tank has only a pumping input and an input for empty memory strands returning.

Most of computations start with initializing, that is why all available memory strands should be divided into two equal groups. The first group will go to setter and the second one will go directly to initializer. The first group will meet the second one inside initializer after all required stickers will be set inside setter.

To divide all memory strands into two groups there is a divider inside pipe-system of MSS. The divider works like groyne (breakwater) and breaks the flow from memory strands tank into two approximately equal flows.

All memory strands are also might be delivered directly to setter (through bypass) or operation router, but most of the tasks will not require this.

Stickers Storage (SS): Stickers Storage (SS) is basically an array of cells. Each cell contains stickers of only one type. Amount of cells (types of stickers) should be equal memory strands length. Amount of stickers of one type inside one cell should be equal or more than memory strands count. Therefore, each existing memory strand inside DNA-part should be able to get one sticker of each type.

Stickers storage also has a pumping fork guiding to each cell. Only one type of stickers might be forwarded to storage's output at one certain moment of time.

Stickers storage has inputs from setter and from clearer for stickers returning. These inputs are also connected with front end of pumping fork inside the stickers storage. Setter and clearer must return a set only of one type of stickers each time, therefore this set could be correctly stored in the corresponding cell inside stickers storage.

Clearers Storage (CS): Clearers Storage (CS) is just a set of DNA-microarrays with anti-stickers of each type. The corresponding DNA-microarray moves to clearer when it is needed and returns back when clearing is completed.

Detector: Detector is a unit that designed to find out if there any memory complexes in the set. It is based on carbon nanotube field-effect transistors (Sorgenfrei *et al.*, 2011).

This kind of detector is high-sensitive, a single DNA-molecule might be detected by one transistor if the molecule will be close enough to transistor. To increase the probability of detection the amount of transistor might be slightly increased. Several transistors are able to detect the fact that there are several DNA-complexes (more than one) in a set.

Carbon nanotube field-effect transistors has a DNA-probe connected to the gate of transistor. This DNA-probe should be completely complementary to empty memory strand. The memory complexes that we want to detect do not have to be free from stickers completely. If memory complex have approximately 3 bit positions one next to other without stickers, then this memory complex is likely to be able to be partially paired with DNA-probe. Even a partially pairing will cause the reduction of conductivity, that could be detected.

If memory complexes have a sticker in every bit position, then it is necessary to remove some of them in clearer before. After the detection, these stickers should be restored if we want to continue the computation. If our memory complexes are large enough, then the chance to get that situation is small.

Pumping system: The pumping system of DNA-part has a pump that makes the solution to circulate in the DNA-part. The pump can work in both directions.

The flow of solution always starts and ends in the pump. If the flow starts at left side of pump and ends at right side of pump, then this flow is called *direct flow*. If the flow starts at right side of pump and ends at left side of pump, then this flow is called *reverse flow*. If one input valve and one output valve of any unit in DNA-system are *Opened for solution*, then the solution can flow through this unit without catching of DNA-molecules.

At any moment of computation, there is only one route for solution exists. Valves should block all other routes.

All possible routes are described below.

- Memory strands go from MSS to operation router. *Direct flow* through pump, MSS, operation router, separators, writing bus, pump loop, reading bus, pump.
- Memory strands go from MSS to setter. *Direct flow* through pump, MSS, setter, writing bus, pump loop, reading bus, pump.
- Stickers go from SS to setter. *Direct flow* through pump, SS, setter, writing bus, pump loop, reading bus, pump.
- Memory complexes go from operation router to setter. *Direct flow* through pump, MSS, operation router, setter, writing bus, pump loop, reading bus, pump.
- Memory complexes go from operation router to clearer. *Direct flow* through pump, MSS, operation router, clearer, reading bus, pump.
- Memory complexes go from setter to RAM. *Direct flow* through pump, SS, setter, writing bus, memory cell, reading bus, pump.
- Memory complexes go from clearer to RAM. *Direct flow* through pump, clearer, writing bus, memory cell, reading bus, pump.
- Memory complexes go from setter to operation router. *Direct flow* through pump, SS, setter, operation router, separators, writing bus, pump loop, reading bus, pump.
- Memory complexes go from clearer to operation router. *Direct flow* through pump, clearer, operation router, separators, writing bus, pump loop, reading bus, pump.
- Memory strands go from operation router back to MSS. *Reverse flow* through pump, reading bus, operation router, MSS, pump.
- The remaining stickers go from setter back to SS. *Reverse flow* through pump, reading bus, setter, SS, pump.
- Free stickers go from clearer back to SS. *Reverse flow* through pump, reading bus, clearer, SS, pump.
- Memory complexes go from operation router to separators. *Direct flow* through pump, MSS, operation router, separators, writing bus, pump loop, reading bus, pump.
- Memory complexes go from separators to RAM. *Direct flow* through pump, MSS, operation router, separators, writing bus, memory cell, reading bus, pump.
- Memory complexes go from operation router to RAM. *Direct flow* through pump, MSS, operation router, writing bus, memory cell, reading bus, pump.
- Memory complexes go from memory cell to reading bus. *Direct flow* through pump, clearer writing bus, memory cell, reading bus, pump.

- Memory complexes go from RAM to operation router. This takes two steps. The first one: memory complexes go from memory cell to reading bus. The second one: Memory complexes go from reading bus to operation router. The first one is described in 16th item above. The second one is *reverse flow* through pump, reading bus, operation router, MSS, pump.
- Memory complexes go from memory to detector. *Direct flow* through pump, clearer, writing bus, memory cell, reading bus, detector, pump.
- Memory complexes go from detector to operation router. *Reverse flow* through pump, detector, reading bus, operation router, MSS, pump.
- Initialization. *Direct flow* through pump, MSS, bipartition: MSS-setter-initializer and MSS-initializer, reunion of flows, then operation router, separators, writing bus, pump loop, reading bus, pump.

Restoration of the DNA-part. All memory complexes from every memory cell go to reading bus like it was described in 16th item of this list. Then all memory complexes go from reading bus to clearer. *Reverse flow* through pump, reading bus, clearer, pump.

RESULTS AND DISCUSSION

The suggested structure of DNA-part hasn't got a dosage system, that's why it always throws all available memory strands from MSS into the further computation. All initialized memory complexes might be divided into subsets only after *Separate* operation. If it is necessary to attach stickers to one subset of memory complexes in specific bit position, then all available stickers for this bit position will be sent to the setter. Afterwards, excess stickers will be returned back to their cell in SS. It doesn't matter, how much bits are required to solve current computation problem, all available DNA-resources will be involved. This approach has advantages, it provides additional memory complexes that can carry the right solution, so this feature increases the chance that the answer will be found, especially for small tasks that don't require the full length of memory complexes.

When the computation is completed the DNA-part should be restored to its initial state. All memory complexes from every unit of DNA-part should be delivered to the clearer. Then every bit position will be cleared one by one. When the next position has been cleared the free stickers go to the corresponding cell in SS.

The DNA-part inside a hybrid computer is an exhaustible system. By the nature of things, some DNA-molecules will be stuck inside pipes and valves. Some of them might go into wrong place. However, sometimes they will be able to return back on track.

The reversible flow of solution can release some molecules that are stuck. Unfortunately, after some cycles of computation the system will become ineffective. In this case, the DNA-part cannot be restored automatically. It needs full manual restoration. Fortunately, most of DNA-molecules from system might be reused.

This study is demonstrated that DNA-computing system might be integrated into traditional binary electronic computer. The detector unit is a link between DNA and traditional parts of hybrid computer, because it allows the traditional part to find out is there a specific data set that was supposed to be there. Depending on result, the traditional part might take a decision what to do with DNA-part at the next step. The unity of DNA and traditional parts allows to develop the software in familiar way. It is necessary to write a device driver which allows traditional part to work with DNA-part. Afterwards a libraries for programming languages might be written. Then a library with DNA-part API might be imported into a programming project. The program that will work with DNA-part could be written like any other software for traditional binary electronic computer.

CONCLUSION

The structure of DNA-part were developed. The suggested system is scalable and reusable. DNA-part is Turing complete, that is why it can be used to compute everything that modern computers can do, but the right way is a balancing of the task between DNA-part and traditional binary electronic part in the hybrid computer. Hybrid computers will be useful in solving problems that requires brute-force search: Boolean Satisfiability Problem, breaking cipher by brute-force search, travelling salesman problem, clique problem and any other NP-hard problems.

The hybrid computers might be useful in scientific and cybersecurity fields.

ACKNOWLEDGMENT

The author would like to thank Erik Winfree for his advice about NUPACK and Niles Pierce for his help and advices via E-mail. Without that help this study could not be done.

REFERENCES

- Adleman, L.M., 1994. Molecular computation of solutions to combinatorial problems. *Science*, 226(5187): 1021-1024. <http://www.sciencemag.org/content/266/5187/1021>
- Benenson, Y., T. Paz-Elizur, R. Adar, E. Keinan, Z. Livneh and E. Shapiro, 2001. Programmable and autonomous computing machine made of biomolecules. *Nature*, 414: 430-434.

- <http://www.nature.com/nature/journal/v414/n6862/abs/414430a0.html>.
- Cheng, Z., M. Lina, D. Yafei, Y. Jing and X. Jin, 2013. Molecular logic computing model based on DNA self-assembly strand branch migration. *Chinese Sci. Bull.*, 58(1): 32-38.
- Elbas, J., O. Liobashevski, F. Wang, F. Remacle, R. Levine and I. Willner, 2010. DNA computing circuits using libraries of DNAzyme subunits. *Nat. Nanotechnol.*, 5: 417-422. <http://www.nature.com/nnano/journal/v5/n6/abs/nnano.2010.88.html>.
- Ke, Y.G., L.L. Ong, W.M. Shih and P. Yin, 2012. Three-dimensional structures self-assembled from DNA bricks. *Science*, 338(6111): 1177-1183. <https://www.sciencemag.org/content/338/6111/1177.short>.
- Lipton, R.J., 1995. Using DNA to solve NP-complete problems. *Science*, 268: 542-545.
- Qian, L. and E. Winfree, 2011. Scaling up digital circuit computation with DNA strand displacement cascades. *Science*, 332(6034): 1196-1201. <http://www.sciencemag.org/content/332/6034/1196.abstract>.
- Rothemund, P.W.K., N. Papadakis and E. Winfree, 2004. Algorithmic self-assembly of DNA sierpinski triangles. *PLOS Biol.*, 2(12). http://www.dna.caltech.edu/Papers/SierpinskiDNA_PLoS2004.pdf.
- Roweis, S., E. Winfree, R. Burgoyne, N.V. Chelyapov, M.F. Goodman, P.W. Rothemund and L.M. Adleman, 1996. A sticker-based model for DNA computation. *J. Comput. Biol.*, 5(4): 615-629. <http://www.ncbi.nlm.nih.gov/pubmed/10072080>.
- Sorgenfrei, S., C.Y. Chiu, R.L. Gonzalez Jr., Y.J. Yu, P. Kim, C. Nuckolls and K.L. Shepard, 2011. Label-free single-molecule detection of DNA-hybridization kinetics with a carbon nanotube field-effect transistor. *Nat. Nanotechnol.*, 6: 126-132. <http://www.nature.com/nnano/journal/v6/n2/full/nnano.2010.275.html>.