DNA Computing in Microreactors

Danny van Noort, Frank-Ulrich Gast, and John S. McCaskill

BioMolecular Information Processing, GMD
53754 Sankt Augustin, Germany
tel.: +49 2241 14 1521/15xx/1526, fax: +49 2241 141511
{danny.van-noort, frank-ulrich.gast, mccaskill}@gmd.de
http://www.gmd.de/BIOMIP

Abstract. The goal of this research is to improve the programmability of DNA-based computers. Novel clockable microreactors can be connected in various ways to solve combinatorial optimisation problems, such as Maximum Clique or 3-SAT. This work demonstrates by construction how one micro-reactor design can be programmed optically to solve any instance of Maximum Clique up to its given maximum size (N). It reports on an implementation of the concept proposed previously [1]. The advantage of this design is that it is generically programmable. This contrasts with conventional DNA computing where the individual sequence of biochemical operations depends on the specific problem. Presently, in ongoing research, we are solving a graph for the Maximum Clique problem with $N = 6$ nodes and have completed the design of a microreactor for $N = 20$. Furthermore, the design of the DNA solution space will be presented, with solutions encoded in customised word-structured sequences.

1 Introduction

DNA computing involves a multidisciplinary interplay between molecular biology, information science, microsystem technology, physical detection methods and evolution. Since the first practical example by Adleman [2] there has been intensive research into the use of DNA molecules as a tool for calculations, simulating the digital information processing procedures in conventional computers. In the short term, however, the main application of DNA computing technology will be rather to perform complex molecular constructions, diagnostics and evolutionary tasks. However, in order to assess the limits of this technology, we are investigating a benchmark computational problem: Maximum Clique, with an NP-complete associated decision problem, chosen because of its limited input information [3]. The step from batch processing in test tubes to pipelined processing in integrated micro-flow reactor networks [1], gives us complete control over the process of information flow and allows operations much faster than in conventional systems. More importantly, it allows the extension to optical programming. Moreover, the proposal differs radically from the surface based DNA computing approach [4] in requiring no problem dependent manual or robotic operations, programming of specific problem instances being completely under

N. Jonoska and N.C. Seeman (Eds.): DNA7, LNCS 2340, pp. 33–45, 2002.

light controlled immobilisation techniques. While other publications report on micro-flow reactor networks [5,6,7], this paper describes the first steps towards practical DNA computing in micro-reactors.

2 Benchmark Problem

2.1 Maximum Clique

The decision problem associated with the maximum clique problem becomes rapidly harder to solve as the problem size increases (it is NP-hard). Maximum clique requires finding the largest subset of fully interconnected nodes in the given graph (Fig. 1). To obtain the set of cliques and then determine its largest member using a micro-flow system, an algorithm was devised consisting of a series of selection steps containing three parallel selection decisions.

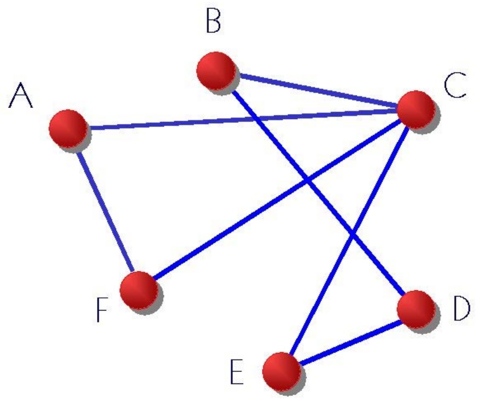

Fig. 1. An $N = 6$ instance of the clique problem. The maximum clique is given by ACF, represented by 101001.

The problem can be divided into two parts: (i) find all the subsets of nodes which correspond to cliques in the graph and (ii) find the largest one. The basic algorithm is simple [1]: for each node i ($i \geq 1$) in the graph retain only subsets either not containing node i or having only other nodes j such that the edges (i, j) are in the graph. This can be implemented in two nested loops (over i and j), each step involving two selectors in parallel. A third selector has been introduced to allow the selector sequences to be fixed independently of the graph instance. Thus the graph dependence is programmed not by which but by whether a sub-sequence selection in the third selector is performed (see Fig. 2). The above procedure is described in more detail by McCaskill [1]. It is important to note that only positive selection for sequences with the desired property is performed, not subtractive selection.

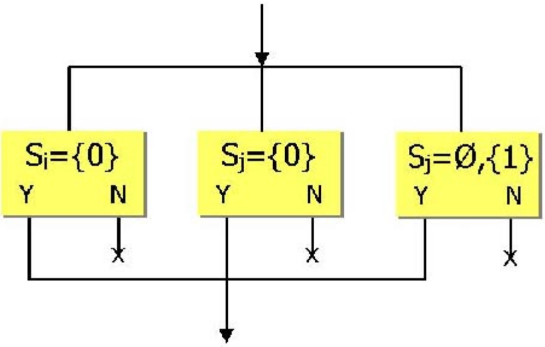

Fig. 2. A flow diagram showing the selection step for node subsets regarding 'cliqueness' at (i, j). The three modules reflect that either node i or node j is absent or the edge (i, j) must be present in the graph.

The edges of the graph, i.e. the connections between the nodes, can be represented by a so called connectivity matrix. The connectivity matrix for the 6-node example shown in Fig. 1 is the 6×6 matrix in Table 1. As Table 1 shows, the matrix is symmetrical over the diagonal, while the diagonal is trivially one, reducing the number of necessary selections from N^2 to $1/2N(N-1)$.

Table 1. The connectivity matrix for the 6-node graph as shown in Fig. 1. The shaded numbers are trivial selections and don't have to be included in the selection procedure to obtain all the cliques. The boxed positions also do not influence the selection, but are included to allow optical programmability.

2.2 Selection Procedure

Each DNA sequence encodes a binary sequence corresponding to a particular subset of nodes in the graph. Different DNA sub-sequences are used to represent presence (1) or absence (0) at each node. As shown in Fig. 2, each selection step consists of 3 selection modules connected in parallel. After each selection step, the sub-population is passed on to the next selection step. Each selection module is coded with short selection-DNA strand (ssDNA) from a finite set of $2N$ predefined sequences. Following the selection algorithm as described above, the final population of DNA-sequences will consist of all the possible cliques represented in the given graph. For the example given graph in Fig. 1, Table 2 shows all the possible cliques.

Table 2. All the possible cliques from the graph shown in Fig. 1. Note that the nodes themselves are cliques as well.

To determine the maximum clique, a sorting procedure has to follow, to select the DNA-sequence with the largest number of bits with the value 1. As shown in the example, Table 2, this would be 101001 (ACF), which with 3 bits of value 1 represents the maximum clique.

Any graph instance up to the maximum size N can be programmed using the same one microreactor design.

3 DNA Library

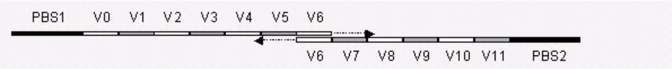

Fig. 3. Design of the DNA used for computation.

3.1 Word Design

The DNA consists of 12 bits (V0–V11) and two primer binding sites (PBS), which contain restriction sites for *Bam*HI ($G \downarrow GATCC$) and *Eco*RI ($G \downarrow AATTC$), respectively. (Top) The complete DNA was assembled by hybridization of two oligodeoxynucleotides in bit V6 and polymerization. Only the upper strand (spanning V0–V6) was randomized as described by Faulhammer et al. (2000); for the overlap assembly, two different lower strands (spanning V11–V6) with fixed sequence were used. Cloning and sequencing gave no indication of a nonrandom composition of the library. (Bottom) The sequence of the DNA words used. The bit value (0 or 1) is given in the suffix after the bit position. The sequences do not show significant similarities

The DNA molecules used consist of a series of 12 words which assume a value of either 0 or 1 (Fig. 3). This gives us a certain flexibility in the choice of a optimal set for the case $N = 6$. The word design is a compromise between maximal specificity of the pairing of the DNA words with complementary probes immobilised on magnetic beads and minimal secondary structure of the single-stranded DNA analysed. Since every word not only represents a bit value, but also the bit position, all DNA words must be unique.

In the original work on the Maximal Clique Problem, the length of the 0 and 1 words were different, which helped to identify the maximal clique from a mixture of all cliques by chromatography [3]. A fixed word length with an identical $G+C$ content (50%) was chosen in order to obtain comparable melting points (in contrast, Ouyang et al. [3] used restriction cleavage for selection). A word length of 16 nt is long enough to ensure specific hybridisation and short enough to minimise secondary structure. Unlike the design used by Faulhammer et al. [8] we did not create constant boundary regions between the bits but checked all the overlap regions for non-specific binding. Furthermore, the purification of the library is easier if one uses a fixed length. Since the sorting of the maximal clique occurs in the DNA reactor, no additional molecular properties are needed.

To perform a word design from random sequences, the following criteria had to be fulfilled:

- the difference in base sequence between different words (Hamming distance) including the primer binding sites (PBS), should be maximal in all different registers. Gaps in the sequence alignment were not considered at this time;
- the distance between the sequences should be valid in all possible frames, i.e., the neighbouring sequences (all combinations) should be included in the analysis;
- in the words or at the boundary between words the restriction sites used in the primer binding regions should be present.

3.2 Synthesis of the DNA Library

The DNA synthesis, performed at NAPS (Göttingen, Germany), followed the "mix and split" strategy of Faulhammer et al. The randomised DNA, consisting of PBS1 and bits V0–V6, was then combined by overlap assembly [3,8] with a non-random molecule, consisting of bits V6–V11 (i.e., the overlap was at V6) and a second PBS2. Following PCR amplification and purification, single-stranded DNA molecules could be prepared by linear amplification and purification on non-denaturing gels.

Unambiguous amplification patterns (for two $N = 6$ individual sequences) were detected when a hot start polymerase was used and the reaction was limited to 20 cycles, indicating that the hybridisation of complementary bits is specific and that the DNA words were properly chosen (see Fig. 4). A unique pattern was even obtained in the case of a 10% contamination with a wrong solution. The PCR readout was monitored in real-time using intercalating fluorescent dyes (iCycler, Bio-Rad, CA; SYBR Green I, Molecular Probes, OR). More detailed results of the biochemical analysis will be published elsewhere.

4 DNA-computer Set-up

4.1 Microreactor Structures

The above procedure can be implemented in a network of micro-reactors. To this end we have developed a module which is able to make positive selections from a population of specific DNA sequences. To actively transfer the selected DNA sequences to the appropriate output, they are transferred from one flow to another by moving paramagnetic beads on which single stranded selector-DNA complementary to a nodal sub-sequence is immobilised. The DNA strands in solution hybridise to the selector-strands and are thus transferred to another channel in the micro-flow reactor where they are de-hybridised and passed on to the next selection procedure. To optimise the transfer of only the appropriate sequences, a washing step has to be performed so as to rinse off the non-specific bound DNA sequences. See Penchovsky and McCaskill [9] for an experimental

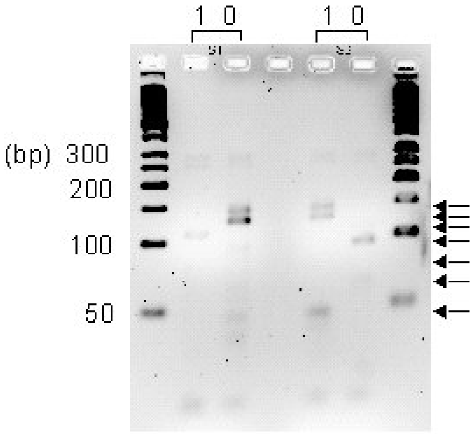

Fig. 4. Typical example of a multiplex PCR analysis. The readout follows Adleman's (1994) strategy, but with a mixture of primers in a single tube, as introduced by Faulhammer et al. (2000). DNAs containing the sequences (bits V0–V6) 0010100 (two left lanes) and 1010011 (two right lanes) were amplified using the constant primer binding to PBS1 (1 μM) and a mixture of reverse primers (hybridizing to V0–V6) with a value of either 0 or 1 (AmpliTaq Gold; 95° C, 10 min, followed by 10 cycles of 95°/47°/72°, 30 s each) and analyzed on a 4% agarose gel (inverted for better readability). If only bands of correct molecular weight are counted (arrows), the correct band pattern (cf. Fig. 3) can be read.

investigation of this process in a separate microreactor design. A typical selection module is shown in Fig. 5.

To prevent the beads from flowing to other STMs, a bead-barrier has been added to the design, which stops the beads from disappearing down the channels. This also provides a straight ledge to move the beads along. Furthermore, by extending the barrier into the channels, a back-flow from the next selection stage is prevented. Experiments and simulations have shown that this design is necessary to ensure a correct flow. More results will be published elsewhere.

Presently we have constructed microflow reactors for $N = 6$ and $N = 20$ nodes and are testing performance firstly with the $N = 6$ version requiring 15 $(= N(N-1)/2)$ STMs and 14 $(= 2N - 1 + 3)$ inlets. The number of inlets scales linearly and the number of modules quadratically in the problem size.

There are $2N$ selection-strands needed, the 0 and 1 bit representation of the N nodes. This number determines the architecture of the DNA-computer microstructures, while these selection-strands have to be transported to the designated selection and sorting modules. In the $N = 6$ case, there will be 11 programming channels, 2 supply channels and 1 template channel. The programming channels will be used for the de-hybridisation solution during operation.

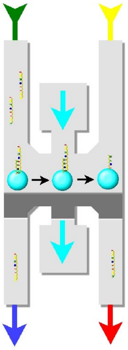

Fig. 5. The Selection Transfer Module (STM). The DNA template enters through channel A. Some of the DNA-strands will hybridise to the selector-strand, which is immobilised on the paramagnetic beads. The beads are moved with a magnet through a wash channel B, to rinse off the unbound strands, into the de-hybridisation channel C from where the selected strands will be transported after a continuos flow neutralisation step, to the next STM. No fluidic switching of flows is required.

Table 3 shows all the ssDNAs necessary to perform the clique selection for a graph of $N = 6$. The selection-strand A_1 is not present due to the fact that it would only be needed to solve the trivial case when comparing node A to itself. The shaded area indicates the positions which must be programmed according to the connectivity matrix. If there is no connection, this field stays empty ($\emptyset$). For the example presented in Fig. 1, the first column, checking the connectivity with node A, would contain beads labelled with ($\emptyset\, C_1\, \emptyset\, \emptyset\, F_1$) successively. It is clear from this table, that the design of the DNA-computer is determined by the problem type and algorithm choice, but not by the problem instance. The design for a 3-SAT problem, for example, would be different. Figure 6 shows the total lay-out for the clique and subset size selection microflow reactor for the case of problems up to $N = 6$.

Table 3. All the necessary selection steps, for a graph of up to $N = 6$ nodes, needed to determine the existence of edges between node i and j. The letter indicates the nodes while the indices denote the bit value (0 or 1). $\emptyset$ is an empty STM. The shaded area is programmable and is determined by the edge between node i and j.

4.2 Sorting Module

The sorting section consist of a parallel selection method to determine the number of bits of value 1 in the DNA-sequence. The algorithm employed is similar in structure to the proposal in Bach et al. [10]. In this design, positive selection for the 1-bits is employed. In the final version of the design (not shown) the 0-bits

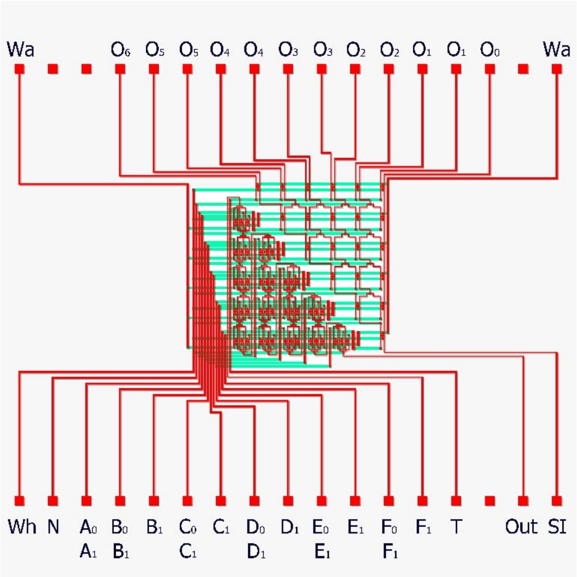

Fig. 6. The complete design of the DNA-computer for graphs with up to 6 nodes. The top right triangle does the length selection, while the other half does the clique selection. The square pads are the input and outputs of the DNA-computer, while the rest of the channels are supply channels. These are connected on the back of the wafer with the reactors (the light grey, horizontal channels). The abbreviations are as follows: Wh is the wash inlet; N is the neutralisation inlet; A_0–F_1 are the ssDNAs programming inlets; T is the input template; Out is the output from the clique sorting modules; SI is the sorting module input channel; Wa is the waste output; Oi are the sorted outputs, with i the clique length. The second row of ssDNAs is for an alternative sorting programming scheme with increasing number of nodes.

will also be actively selected. There are 12 outputs from the selection in which the population is classified (only seven are needed). Table 4 shows the ssDNA needed to perform the sorting procedure as shown in Fig. 6. After step 1, all the strands with A_0 in the sequence will flow down to the step 2, while all the other strands with A_1 move one column to the right and to step 2. This process continues until the output layer is reached.

Table 4. All the selection-strands needed for the length sorting procedure. The output gives the number of bits with value 1.

It can be seen that the path through the sorting module gives the sequence's code. This means that if the DNA strands are fluorescence labelled, their path can be followed optically (e.g. Mathis et al. [11]) and no gel electrophoresis is needed to analyse the sequence.

4.3 Clique Selection Module

The STMs have three supply channels, for the DNA-solution template, the washing solution and de-hybridisation buffer respectively. De-hybridisation is performed by using an alkali solution (NaOH), adjusted in concentration for the common "melting" temperature of the hybridised DNA strands. Because of the change in pH at the de-hybridisation step, a subsequent neutralisation step is necessary after each selection stage before flowing the selected DNA into the next module. This procedure has been successfully applied as shown by Penchovsky et al. [6]. In order to allow the magnetic beads to move uniformly from left to right in all the STMs over the entire microflow reactor, alternate stages are mirrored (see Fig. 7). When the DNA-sequences de-hybridises from the beads, the beads in the next stage will be in the correct place for hybridisation to take place.

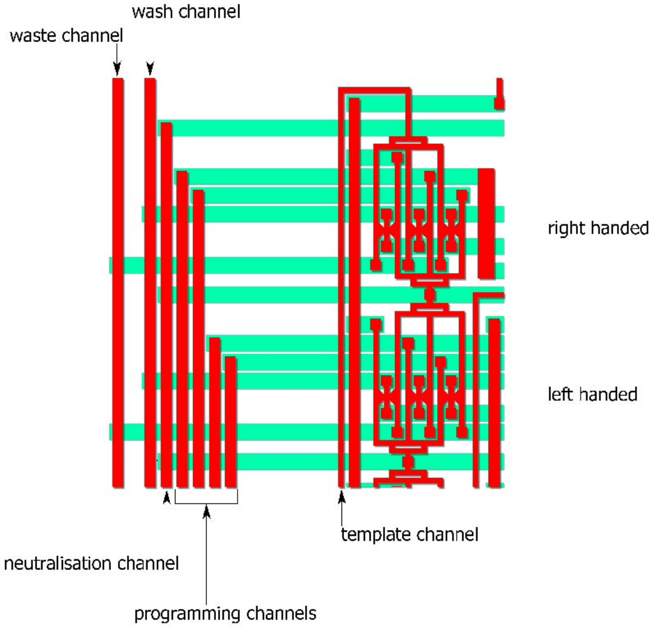

Fig. 7. Two selection steps in sequence as found in the microflow reactor design. Each selection step has 3 selection modules for a node connectivity decision (dark grey) with the template, wash, programming/de-hybridisation, neutralisation and waste channels. The supply channels are connected to the modules by horizontal channels etched on the back of the silicon wafer (light grey).

4.4 Programmability

These Strand Transfer Modules (STM) can be optically programmed as outlined in [1], by means of photo-immobilisation [12], thus creating a programmable micro-fluidic computer. Unlabeled beads are delivered in parallel at the appropriate locations to each STM. UV laser light then determines whether or not a sequence is immobilised on the beads at each STM. If there is no connectivity between a pair of nodes, the corresponding third STM is not immobilised. The other STMs can be preloaded with immobilised beads since they will never have to be changed when re-programming the problem. The immobilisation pattern is directly related to the connectivity matrix, from which a programming mask can be derived. The information flow can be tracked using a sensitive CCD detection system to detect laser-induced fluorescence with intercalating dyes or labelled DNA Because of the fluorescent information from each STM in which a correct DNA strand transfer occurs, it is possible to monitor the solution of the algorithm to its conclusion over time.

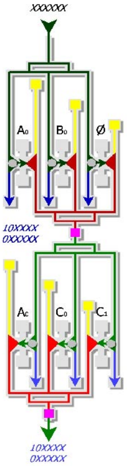

Fig. 8. A close up is shown of the first two clique selection steps. The flows and transfer are shown with the coloured lines. The binary sequences (where $X = 0$ or 1) indicate what selection has taken place.

4.5 Etching Procedure

The whole micro-reactor configuration is photolitho-graphically etched on 4” (100) silicon substrates. The etched wafer is sealed with an anodically bonded pyrex glass wafer. Capillary tubing (0.8 mm diameter) is attached through ultrasound drilled holes in the pyrex wafer. The distance between the holes is 3.5 mm. The supply channels width on the front side are 200 _m, while those on the back are 300 _m. The relatively large width is to reduce the flow resistance and pressure drops in the structure. The channel width in the STMs is 100 _m. To etch through the 220 _m thick silicon wafer, etching pads on the front (200 × 200 _m) and on the back (300 × 300 _m) are made as to obtain holes of 100 × 100 _m. These then have the same width as the channels in the STM.

4.6 Set-up Overview

To set up a DNA-computer in a micro-flow system puts high demands on the control system, an overview of which is presented in Fig. 9.

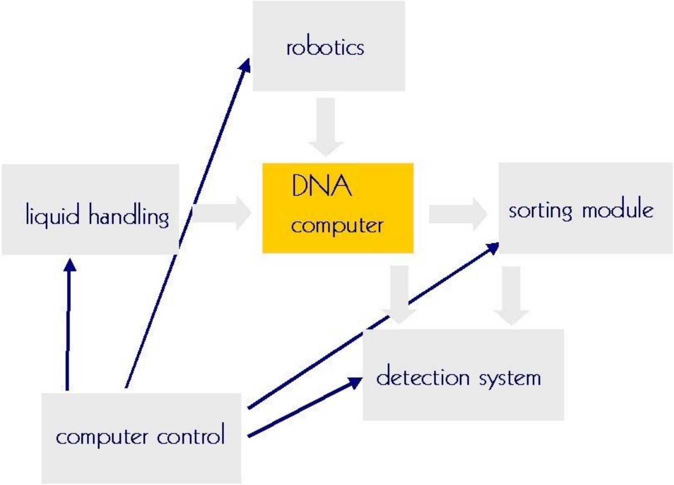

Fig. 9. An overview of the operation of a DNA computer.

To distribute the DNA template and buffer solutions to the wafer, a liquid-handling system is connected. It consists of a pipetting robot and a series of multi-position valves which control the solution distribution. Flow rates will be smaller then 1 _l/min. One of the multi-position valves, together with the micro-structure architecture, makes it possible to address the individual immobilisation sites in the STMs for the selection strand. This can readily be made parallel for full scalability. To programme the computer optically laser light from an EXCIMER laser (308 nm) is used to project either serially trough a microscope or through a static (later dynamic) mask. The wafer is mounted on an *xy*-translation stage so as to address all the STMs individually. A detection system consisting of a CCD camera for a general overview and a microscope for detailed pictures is in place.

The beads are moved by a magnet which sweeps over the DNA-computer's surface. This sweep clocks the serial steps of the computation, from one stage to the next, which is pipelined to increase throughput.

5 The Next Step: A 20-node DNA-computer

Presently an initial design for a DNA-computer has been constructed to solve a Maximum Clique problem for $N = 20$. The principle layout is the same as for the $N = 6$ case, although it will require a tighter control over the flows. This design has 190 STMs, 210 sorting modules, 43 inlets and 43 outlets. Figure 10 shows the top channel layer for the 20-node computer. The first version has been etched on a 4" wafer in the same fashion as the 6-node computer described above.

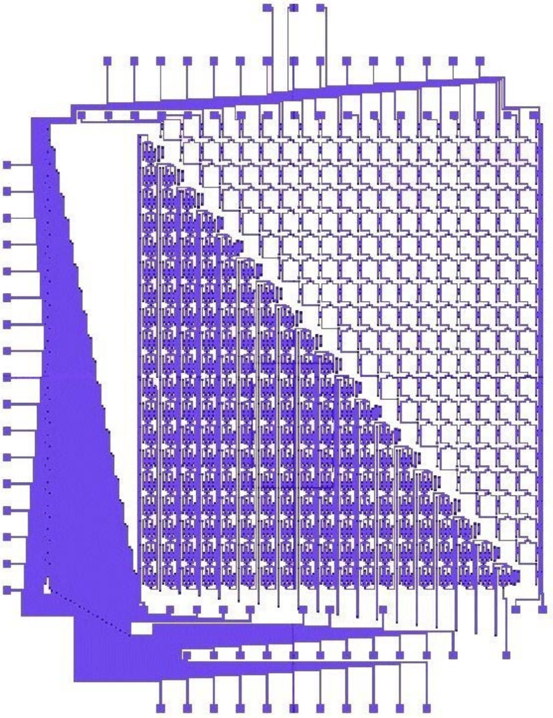

Fig. 10. The complete top layer of the DNA-computer for a 20-node graph. The top right triangle does the length selection, while the other half does the clique selection. The square pads are the input and outputs of the DNA-computer, while the rest of the channels are supply channels.

6 Conclusion

Microflow reactors will prove to be a powerful tool to construct a programmable parallel DNA computer. The main advantage is in the light programmable, integrated system (operating under steady flow) with no problem dependent pipetting steps. Future reactor configurations can be made re-configurable with evolving DNA-populations to obtain a universal programmable DNA-computer [13].

7 Acknowledgement

The authors wish to acknowledge the support of the GMD for DNA computing and the German Ministry of Science (BMBF) for BioMIP's start-up grant (#01SF9952). We would like to thank Marlies Gohlke, Harald Mathis, Robert Penchovsky, Thomas Rücker and Patrick Wagler for their assistance with the word design, biochemical tests, microreactor implementation and set-up.

References

1. McCaskill, J. S. (2001) Optically programming DNA computing in microflow reactors. Biosystems **59** (2), 125–138.
2. Adleman, L. M. (1994) Molecular computation of solutions to combinatorial problems. Science **266**, 1021–1024.
3. Ouyang, Q., Kaplan, P. D., Liu, S. and Libacher, A. (1997) DNA solution to the maximal clique problem. Science **278**, 446–449.
4. Liu, Q., Wang, L., Frutos, A. G., Condon, A. E., Corn, R. M. and Smith, L. M. (2000) DNA computing on surfaces. Nature **403**, 175–179.
5. Gehani, A. and Reif, J. (1999) Micro flow bio-molecular computation. BioSystems **52**, 197–216.
6. Ikuta, K., Maruo, S., Fujisawa, T. and Hasegawa, T. (1999) Biochemical IC family for general micro chemical systems. Transducers '99, June 7–10, 1999, Sendai, Japan, 1046–1049.
7. Suyama, A. (1997) DNA Chips — Integrated Chemical Circuits for DNA Diagnosis and DNA Computers. Proc. of Third International Micromachine Symposium, 7–12.
8. Faulhammer, D., Cukras, A. R., Lipton, R. J. and Landweber, L. F. (2000) RNA solutions to chess problems. Proc. Natl. Acad. Sci. USA **15**, 1385–1389.
9. Penchovsky, R. and McCaskill, J. S. (2001) Cascadable hybridization transfer of specific DNA between microreactor selection modules. Submitted.
10. Bach, E., Condon, A., Glaser, E. and Tanguay, C. (1998) DNA models and algorithms for NP-complete problems. J. of Comp. and Sys. Sci. **57**, 172–186.
11. Mathis, H., Kalusche, G., Wagner, B. and McCaskill, J. S. (1997). Steps towards spatially resolved single molecule detection in solution. Bioimaging **5**, 116–128.
12. Penchovsky, R., Birch-Hirschfeld, E. and McCaskill, J. S. (2000) End-specific covalent photo-dependent immobilisation of synthetic DNA to paramagnetic beads. Nucleic Acids Research **28**, e98, 1–6.
13. McCaskill, J. S. and Wagler, P. (2000) From reconfigurability to evolution in construction systems: spanning the electronic, microfuidic and biomolecular domains. In R. W. Hartenstein and H. Grünbacher (Eds.) FPL 2000, LNCS **1896**, pp. 286-299, Springer-Verlag, Berlin Heidelberg.