Micro flow bio-molecular computation

Ashish Gehani *, John Reif

Department of Computer Science, Duke University, Durham, NC 27708 0129, USA

Abstract

In this paper we provide a model for micro-flow based bio-molecular computation (MF-BMC). It provides an abstraction for the design of algorithms which account for the constraints of the model. Our MF-BMC model uses abstractions of both the recombinant DNA (RDNA) technology as well as of the micro-flow technology and takes into account both of their limitations. For example, when considering the efficiency of the recombinant DNA operation of annealing, we take into account the limitation imposed by the concentration of the reactants. The fabrication technology used to construct MEMS is limited to constructing relatively thin 3D structures. We abstract this by limiting the model to a small constant number of layers (as is done with VLSI models). Besides our contribution of the MF-BMC model, the paper contains two other classes of results. The main result is the volume and time efficient algorithm for message routing in the MF-BMC model, specifically useful for PA-Match. We will show that routing of strands between chambers will occur in time

$$O\left(\frac{N \cdot D}{m \cdot n}\right),$$

where N is the number of strands in the MF-BMC, n is the number of chambers where RDNA operations are occurring, D is the diameter of the topology of the layout of the chambers, and m is proportional to the channel width. Operations that need annealing, such as PA-Match, are shown feasible in

$$O\left(\frac{N^2 \log \frac{N}{n}}{n}\right)$$

volume instead of the previous use of $\Omega(N^2)$ volume, with reasonable time constraints. Applications of the volume efficient algorithm include the use of the Join operation for databases, logarithmic depth solutions to SAT (Boolean formula satisfiability) problems and parallel algorithms that execute on a PRAM. Existent algorithms can be mapped to ones that work efficiently in the MF-BMC model, whereas previous methods for applications such as PRAM simulation in BMC were not both time and volume efficient. Our other class of results are theoretical lower bounds on the quantities of DNA and the time needed to solve a problem in the MF-BMC model, analogous to lower bounds in VLSI. We bound the product BT from below, and further show that BT^2 has a stronger lower bound of I^2. Here B is the maximum amount of information encoded in the MF-BMC system at a time. T is the time for an algorithm to complete, and I is the information content of a problem.

* Corresponding author.
E-mail address: {gehani,reif}@cs.duke.edu (A. Gehani)

Preprint of paper appearing in a special issue of Biosystems, Journal of Biological and Informational Processing Sciences, Vol. 52, Nos. 1-3, (Edited By L. Kari, H. Rubin, and D. H. Wood), pp 197-216, (October 1999).

Keywords: MEMS; DNA computing; Parallel computation

1. Introduction

A significant aspect of chemical reactions is the randomness with which they occur at the molecular level even though there is a discernible pattern at a higher level as described in kinetic models (SantaLucia et al., 1996; Wetmur, 1997; Winfree, 1998). One of the results is that even though we can predict that a reaction between two molecules will eventually occur, the larger the volume in which the reactants exist, the less likely it is and so the longer it will take. This is a situation where we wish particular reactants to interact with each other but not with other incidental complexes as that merely increases the overall time before the desired reaction is completed. In such a situation if we were able to place molecules between which an interaction was desired, into a smaller volume then they would interact much more quickly with high likelihood.

Conversely, there are certain reactants that we would prefer to isolate. While the ability to perform operations in parallel on the entire data set has advantages, it brings with it the drawback that reagents added to execute an operation necessarily interact with all the bio-molecular material present, even that which it does not need to. In addition, the presence of extraneous material increases the chance of partial matching and other such incomplete interactions occurring, causing avoidable errors. In such cases, we would like to be able to separate the data set where possible.

Living systems have evolved a simple method to effect some control in such situations. We refer here to the division of an organism into its many constituent cells within which independent biochemical reactions occur. Through the use of a circulatory system, materials are routed to particular cells which perform different operations in parallel on the material. In addition the material can be processed and then sent along to another destination through the infrastructure of the circulatory system. By having limited volumes in each cell, the reactants that need to interact can do so efficiently. Certain highly reactive chemicals which have positive uses but can have a detrimental effect on other reactants and reactions can be isolated within particular cells, where they can be productively used without affecting other material (Fig. 1).

Analogously, we propose a framework in which BMC can occur in a large number of isolated regions of limited volume. As with a living system, there will be an associated infrastructure that

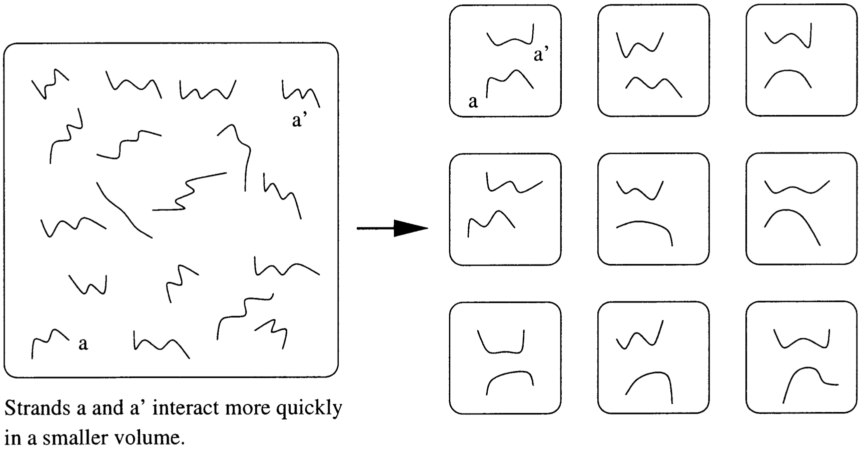

Fig. 1. Strand a and a′ interact more quickly in a smaller volume.

will allow reactants, reagents and associated material to be transported from one region to another. Since our method will make use of micro-flow device technology, we refer to the framework as micro-flow bio-molecular computation (MF-BMC). A 3D MEMS fluid system has been proposed in a different context (Ikuta 1997) but does not examine the BMC potential in any detail.

All the operations used for DNA computing as described by Adleman (1996), Lipton (1996), Reif (1997, 1998a) will be supported in the MF-BMC framework. The details of the operations can be found in their respective papers. One of the early concerns regarding the scalability of BMC was the issue of limiting the number of steps that must be performed in the laboratory (Adleman, 1996; Reif, 1998a). We note that the issue has dual roots. The first is the long periods of time needed for operations such as anneal. As outlined in Section 1, we propose to alleviate this with smaller volumes. The other reason is that it is not practical to expect a human laboratory technician to be able to execute a large number of steps. Three different approaches arose to deal with the latter. One was to attempt to effect molecular self-assembly that would obviate the need for human intervention as described by Winfree (1995). Another was the reformulation of BMC models so that the complexity analysis incorporated a notion of the operations with the restriction of utilizing a single laboratory technician (Amos et al., 1997). A third possibility was the introduction of automation to perform the necessary tasks, as described in the paper introducing the idea of a *stickers machine* based on the *sticker model* (Roweis et al., 1996). Our model builds further on this approach. We assume the mechanical equivalent of an army of technicians acting in parallel and controlled by a conventional microprocessor, in contrast to the first and second approaches. Algorithms, such as Ogihara's solution to SAT (Ogihara and Ray, 1997), which have significantly higher complexity in the 'strong model' than in the 'weak model' and are thus deemed impractical (Amos et al., 1997), can be implemented practically in the MF-BMC since distinct operations can be executed at distinct locations simultaneously. The MF-BMC framework can be differentiated functionally from the stickers machine in two respects. The first is the scale on which the total volume is expected to be divided into smaller volumes. Since the reduction in total volume needed will be seen to be proportional to the scale of this division, this scale will necessarily be large. Secondly, since the topology envisioned is a two dimensional array, there exist, and we describe, methods to efficiently route strands between chambers. All MEMS devices are assumed to be controlled by a micro-processor.

2. Motivation

2.1. Overview and assumptions

The model described will have a fixed number of layers. In each layer there will be a grid along whose lines will run channels for liquid transport. At the grid points will be located chambers in which biochemical reactions can occur.

Architectural limits much like those of VLSI models give rise to some of the model's assumptions. We can summarize these as follows. The first limit is the small constant number of planar layers due to the current limits of MEMS fabrication technology. The next is the lower bound on the size of any component of the system. If the volume of a chamber is too small then meaningful computation can not be effected in it. If the area of cross-section of a channel is too small then practical movement of fluid volumes will not be possible. Also, the rate of transport of fluids between chambers must be sufficient for routing between chambers to be practical. Further constraints analogous to the Meade-Conway design rules of VLSI may be added to carry the model from a theoretical proposition to one useful in practice.

Additionally, we assume that we can detect hybridizations that are exact, and we assume that these occur without error. Further we do not consider kinetics asymptotically as we expect to run them for only constant periods of time with the focus on interaction between local pairs of strands.

2.2. Strand routing efficiency

The process by which a strand in a chamber reaches another strand with which it will combine using some RDNA operation such as anneal or ligate, is essentially probabilistic. Heuristics exist that allow the manipulation of quantities such as the melting temperature and the rates of reactions by altering the salt concentration, metal ion concentration, the temperature of the solution, the hydrogen potential, and the ratio of purine bases to pyrimidine bases (SantaLucia et al., 1996; Hartemink and Gifford, 1997; Rose et al., 1997; Wetmur, 1997). For significant yields of a particular product, the reactant concentrations must necessarily be adjusted appropriately (Kurtz et al., 1996; Winfree, 1998). This is where the need for quadratic volume for guaranteed reactions between strands enters, as explained below.

For the purpose of analysis in this section. we assume that strand concentration remains relatively invariant. That is, a reference to $O(N)$ volume has implicit in it that there are $O(N)$ times as many strands as in a unit volume. Additionally, since the model being developed is for effecting computation, rather than effecting biochemical reactions till they reach their thermodynamic equilibrium, the creation of even single copies of expected products in reactions is the desired result in a chamber.

Specifically, if m_i copies of strand s_i interact with m_j copies of strand s_j resulting in m_h copies of the heteroduplex s_h, then $m_\mathrm{h} > 0$ is considered a successful routing between the two strand types s_i and s_j. In N volume the initial rate of production of s_h will be

$$\frac{\mathrm{d}c_\mathrm{h}}{\mathrm{d}t} = k_\mathrm{f} \cdot c_i \cdot c_j,$$

where c_h is the concentration of s_h, k_f is the constant associated with the forward reaction, c_i is the concentration of s_i, which is

$$\frac{m_i}{kN_\mathrm{A}},$$

and c_j *is* the concentration of s_j, which is

$$\frac{m_j}{kN_\mathrm{A}}.$$

N_A is the Avogadro constant and k is a fixed scaling factor such that the total number of strands in N volume is kN_A. We expect a single copy of s_h in the N volume would be produced immediately with probability proportional to the rate

$$\frac{\mathrm{d}c_\mathrm{h}}{\mathrm{d}t}.$$

Therefore, in a volume:

$$\frac{1}{\frac{\mathrm{d}c_\mathrm{h}}{\mathrm{d}t}},$$

we would expect a single copy of s_h to be formed immediately with constant probability independent of N. Simplifying this by considering the case where $m_i = m_j = 1$, then

$$\frac{1}{\frac{\mathrm{d}c_\mathrm{h}}{\mathrm{d}t}} = \frac{N^2}{k_f}.$$

Thus, by using a total $O(N^2)$ volume, that is one with $O(N)$ copies of s_i and $O(N)$ copies of s_j among a total of $O(N^2)$ strands, one can expect the immediate production of a copy of s_k with fixed probability.

This result is borne out by considering the strand interaction from a different point of view as well. Suppose we assume that the strands are moving with Brownian motion in the chamber. In a chamber with N copies of strands s_i denoted by $s_{i1},\ldots, s_{iN}$ and N copies of s_j, denoted by $s_{j1},\ldots, s_{jN}$, among a total of N^2 strands, the probability that s_{i1} interacts immediately with s_{j1} is

$$\frac{1}{N^2}.$$

The probability that s_{i1} interacts immediately with any s_{jx} where $\mathrm{x} \in \{1,\ldots, N\}$ is

$$N \cdot \frac{1}{N^2} = \frac{1}{N}.$$

The probability that s_{i1} does not interact with any s_{jx} is

$$1-\frac{1}{N}.$$

Thus, the probability that all s_{iw} where $w \in \{1,\ldots, N\}$, do not interact with any s_{jx} is

$$\left(1-\frac{1}{N}\right)^{N},$$

which is $1/e$ where e is the natural logarithm base, a constant. Thus, the probability that one of the strands of type s_i interacts immediately with one of the strands of type s_j is

$$1-\frac{1}{e},$$

which is a constant.

Further, by using $r.N \log N$ volume, where there are $r.\log N$ copies of each of the N different types of strands. the probability of hybridization not occurring is

$$\left(1-\frac{1}{N}\right)^{rN\cdot \log N}=\frac{1}{e^{rN\cdot \log N}}=\frac{1}{N^{r}}.$$

The probability of each of the different N strands being routed to one of the copies that we desire it to interact with is, therefore, $1-N^{-r}$, which is close to unity for reasonable r. After T time steps, we would expect that the computation would achieve the desired result with probability

$$\left(1-\frac{1}{N^{r}}\right)^{T}.$$

In particular, by increasing the volume redundancy r linearly, we can increase the number of time steps T exponentially without decreasing the probability of a successful computation.

3. MEMS micro-flow devices

In order to replace the laboratory technician, we must create a system to effect the same actions that the technician would perform. This is achieved by the use of MEMS pumps, valves, and sensors.

We only refer to the devices as abstractions even though there is a wealth of available MEMS technology. The reason is that the choice is an engineering issue that depends on many constraints. In general, the scale of the MF-BMC and the budget available for building the MF-BMC will affect the choice of the implementation. In addition, other factors specific to the device will play a role. For example, consider the use of a micro-pump. In the event that a larger budget is available, several pumps of different technology could be used. One would be suited to maximize flow when the strands are expected to be linear. Another would minimize turbulence when the data strands are expected to be more complicated structures. A third would be used when the strands are being pumped vertically.

3.1. Actuators

These are the devices that translate a control signal into some form of mechanical motion. There are numerous available implementations (Manz et al., 1992; Shoji and Esashi, 1992; Verpoorte et al., 1992; Adey et al., 1995). In the description of the model, we will use the notation $A(i)$ to refer to the operation of an actuator in the component i, where i can be either a chamber or a channel as described in Section 4. An actuator can either be on or off and this is denoted by 1 or 0 respectively when specifying this in an algorithm running on the controlling microprocessor.

3.2. Micro-valves

One of the critical components of a system that controls the flow of DNA strands and complexes is necessarily a valve or regulator of some sort. Details of the constructions can be found in the literature (Manz et al., 1992; Verpoorte et al., 1992; Adey et al., 1995). In the description of the MF-BMC model, we will use the notation $V(i)$ to refer to the operation of a micro-valve in the channel i as described in Section 4. A micro-valve can either be open or closed and this is denoted by 1 or 0, respectively, when specifying this in an algorithm running on the controlling microprocessor.

Depending on what type of actuator is used, the micro-valve will have different characteristics such as how quickly it can open and close, whether it is better suited to situations where it is open most of the time or to situations where it is closed most of the time, whether it can control flow across a channel with a large area of cross-section, and what pressure it can withstand. In many situations, it will be appropriate to place several micro-valves based on different technologies at a given point where flow control is required, along with micro-sensors that will activate the micro-valve most suited to the immediate need, and deactivate the remaining ones.

Three-way flow control has been implemented (Shoji and Esashi, 1992) and provides the abstract basis for performing selection between multiple routing paths. It is easily generalized to allowing regulators at a node to connect a chosen source channel to a chosen destination channel.

3.3. Micro-pumps

In order to transport the information around the MF-BMC, micro-pumps are employed to move the solution and the DNA strands contained in it, that are in a particular chamber to another chamber, as described in Section 4. Refer to the literature (Shoji and Esashi, 1992) for details. For the purpose of the model, the abstracted representation $P(i)$ will be used. It refers to the appropriately chosen micro-pump operating along the channel i as described in Section 4. A micropump can either be operating in one of two directions or it can be off and this is denoted by setting it to 1, -1 or 0, respectively, when specifying its use in an algorithm to the controlling micro-processor. The 1 refers to pumping in the direction of the chamber with the larger subscript, and -1 refers to the pumping from the chamber with the larger subscript. (Note that the reference i to a channel implicitly includes two chambers as parameters. It is the subscript of these that we refer to here). As with micro-valves, depending on the underlying technology used for the actuator, the micro-pumps have various characteristics such as the rate at which volumes can be pumped, the pressure that can be built up by the micro-pump without side-effects, and the ability to reduce or contain aspects such as rippling or turbulence, which are especially important in the context of DNA computing.

3.4. Flow sensors

In the later description of the model, the notation $Sen(i)$ will refer to a sensor that is located in the channel i or the chamber i. A flow-sensor reports 1 to the controlling microprocessor if the flow is above a certain threshold, and 0 otherwise. The implementation of the abstraction will be based on a *liquid flow sensor*, the type most suited to our needs, which can be of several varieties. Refer to Manz et al. (1992), Shoji and Esashi (1992), Verpoorte et al. (1992)) for more details.

3.5. Flow control systems

By coupling a flow sensor's output to a micro-valve's actuator, and to a micro-pump's actuator if necessary, a feedback system is created whereby micro-flow in a system can be largely automatically controlled.

3.6. Digital to analog conversion

For the purpose of the micro-processor effecting control upon various MEMS components, we note that this is easily implemented by the use of micro-flow devices that use electrical circuits in them, such as the solenoid of an electro-magnet or the resistor of a heater. This is because the micro-processor need only use an amplifier to increase or decrease the voltage and current as necessary, and then let the output be connected through wires to the relevant micro-flow devices.

4. Architecture

The architecture of the model envisioned consists of a small constant number, $\Pi_{\max}$, of planar layers (Fig. 2). We denote these planes by $\Pi(i)$, where i ranges from 1 to $\Pi_{\max}$. The composition of each one is described below. We note here that the planar nature is due to the inherent constraints of current MEMS fabrication. In addition, although we limit our model to a fixed number of layers, in principle if the micro-flow device technology were to allow arbitrary vertical scaling, then our model and the associated analy-

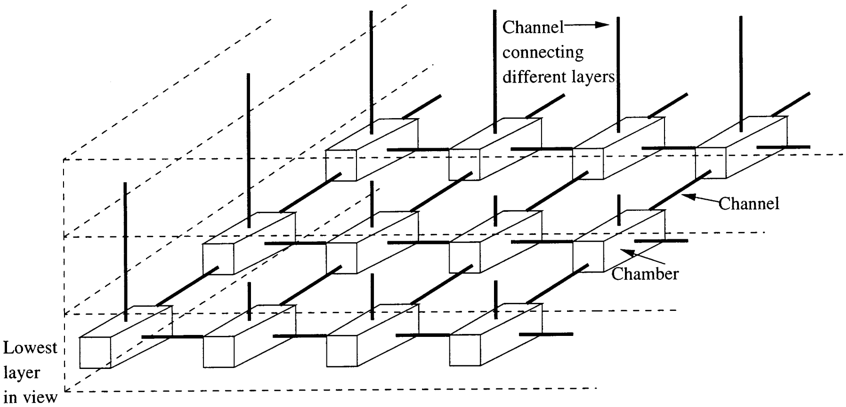
Channel connecting different layers
Channel
Chamber
Lowest layer in view
Proposed architecture for Micro-Flow Bio-Molecular Computation consists of several planar layers, in which an array of chambers connected by channels exists. Channels from layers above bring in data strands, reagents needed for reactions.

Fig. 2. Proposed architecture for Micro-Flow Bio-molecular Computation consists of several planar layers, in which an array of chambers connected by channels exists. Channels from layers above bring in data strands, reagents needed for reactions.

sis could be scaled to 3D. The intuition for this can be seen in the literature on parallel architecture design (Thomson Leighton, 1992).

The architecture relies on three orthogonal technologies, namely DNA computation in a test-tube as previously envisioned (Reif, 1998b), packet routing in networks, and micro-flow devices. In addition, the various aspects are tied together using a conventional silicon based micro-processor. As we proceed to describe the model, we refer to particular implementations of the technologies mentioned. We note, however, that in all three fields there are numerous competing alternatives that could be used in place of the specific instances that we have chosen to use.

4.1. Composition of a layer

A typical layer is solid all through with four exceptions. These are (1) the *chambers* in which RDNA operations occur, (2) the *channels* which connect the chambers, (3) the *control lines* which allow a single silicon-based micro-processor to synchronize the RDNA and MEMS operations, and finally (4) the MEMS devices that effect the routing.

4.1.1. Chambers

4.1.1.1. Layout. Chambers are cavities in which the RDNA model computations can occur. We denote these regions by $R(i)$, where i ranges from 1 to $R_{\max}$, the number of chambers in the MF-BMC.

Some planes will use a regular layout in the form of a grid of chambers. Typically, this would be the case for plane $\Pi(1)$, where the chambers in which the RDNA operations are executed. Other planes, where reservoirs that hold pools of strands at intermediate points during their transfer from one chamber to another, would also have such structure. Such planes have h chambers in each row and w chambers in each column, giving us a value of $h \times w$ for $R_{\max}$. When such a layout is implicit, we refer to the chamber $R(i \cdot w + j)$ with the notation $R(i, j)$, where $1 \leq i \leq h$, and $1 \leq j \leq w$. When there is ambiguity regarding which plane a chamber is in, we use the notation $(\Pi(i), R(j))$ to disambiguate the reference.

4.1.1.2. Structure. A chamber $R(i)$'s height, width and depth respectively are denoted by $h_c(R(i))$, $w_c(R(i))$ and $d_c(R(i))$. The maximum volume that can be stored in the chamber is therefore $h_c(R(i)) \times w_c(R(i)) \times d_c(R(i))$. The *aspect ratio* is the quotient of the height and the width. If it is not within a certain factor from unity, then the chamber's design could introduce effects such as uneven diffusion of reactants.

4.1.1.3. Micro-flow enablers. Each chamber $R(i)$ has an inlet and outlet on opposing sides. The inlet flow is controlled by a micro-valve $V_{\text{inlet}}(R(i))$, and the outlet flow by a micro-valve $V_{\text{outlet}}(R(i))$, both of which are of the type described in Section 3. These are for DNA strands to be brought in and removed. We note here that each inlet and outlet have the ends of channels from up to four other chambers ending at them if the architecture is a regular grid of chambers in the plane. From an opening in the ceiling of the chamber, controlled by a micro-valve $V_{\text{reagent}}(R(i))$ reagents may be introduced. Through the use of this micro-valve's interface to the layer above and the additional power provided by a micro-pump denoted by $P_{\text{reagent}}(R(\text{i}))$, reagents may be flushed out. In addition, this is the interface that will be used to connect to layers above and below. Below each chamber, a *reservoir* of dimensions equal to that of the chamber, is present for temporary storage. It is denoted $R'(i)$, and is connected to the chamber via a micro-valve, denoted $V_{\text{reservoir}}(R(i))$. Contents can be pumped to and from it through this valve by the activation of a two-way micro-pump, denoted by $P_{\text{reservoir}}(R(i))$.

4.1.1.4. RDNA enablers. Near the inlet and outlet will be located two electrodes, through which current is passed when gel electrophoresis is performed in the chamber (Fig. 3). The cathode, denoted $E^{-}(R(i))$ is placed near the inlet, and the anode, denoted $E^{-}(R(i))$ is placed near the outlet.

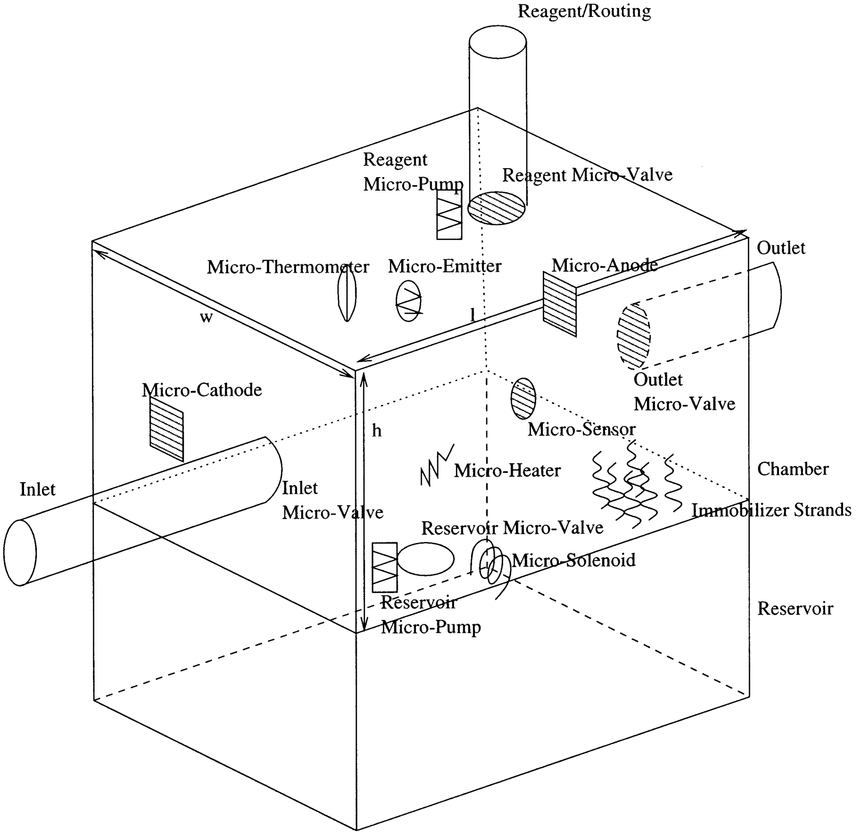

Fig. 3.

This is due to the fact that DNA is negatively charged on account of its its sugar–phosphate backbone which contains PO_4^- an ions. Using solenoids, denoted $Sol(R(i))$, which are embedded above and below the chamber, magnetic fields may be introduced as required. By placing a suitably protected resistor based micro-heater, denoted $H(R(i))$, in the bottom of the chamber, heating of the contents may be performed, with convection helping circulate the materials. By placing an optical emitter and sensor, denoted $OE(R(i))$ and $OS(R(i))$, respectively, at the two ends of the tube, required measurements may be taken. The cathode, anode, solenoids, heater and emitter can all be either on or off, and when specifying this to the controlling micro-processor this is done by setting the respective abstraction to 1 or 0, respectively. The optical sensor, as in the case of the flow sensor of Section 3, will provide feedback to the micro-processor according to whether its reading crosses a threshold. Each chamber is equipped with a micro-thermometer, denoted $T(R(i))$ which can be read by the micro-processor in real-time and provides continuous valued feedback.

A final type of RDNA enabler present in each chamber is a set of known nucleotide sequences that have been anchored to the walls of the chamber. Various operations can take advantage of these as they are always known ahead of time and are distinctly encoded for each chamber. For example, consider the problem of immobilizing all strands of a certain class that are in the chamber $R(i)$. Assume that the strands anchored in the chamber are composed of the single stranded sequence $s_{R(i)}$ called the *anchor strands*, and the single stranded DNA to be immobilized are of sequence types $t_1,\ldots, t_m$, where m is the count of the number of different strands to be immobilized. We assume without loss of generality that $s_{R(i)}$ and all the t_j, where $1 \leq j \leq m$, are oriented in the same direction. The complementary *immobilizer strands* $s'_{R(i)}\mathrm{t}'_j$ for all j, where $1 \leq j \leq m$, can now be introduced. Annealing can be induced and the strands t_j will now be anchored down as they will anneal to the immobilizer strands which are annealed to the anchor strands. By choosing sequences for $s_{R(i)}$ which bind more strongly to their complement than the various t_j do, denaturing can be selectively performed such that only the t_j strands break away.

4.1.2. Channels

The second type of exception to the solidness of a layer consists of the *channels*, which are conduits connecting the various chambers as required. A channel is fully specified by its two endpoints, each of which must be a chamber. Thus a channel connecting two topographically adjacent chambers $R(i)$ and $R(i+1)$, would be denoted by $C(R(i), R(i+1))$. In addition, each channel has a micro-pump as described in Section 3 placed in it so as to facilitate the flow of the contents from one chamber to another. The micro-pump for the above example channel would be denoted by $P(C(R(i), R(i+1)))$. Similarly, the channel's length and radius of cross-section would be denoted by $l_c(C(R(i), R(i+1)))$ and $r_c(C(R(i), R(i+1)))$, respectively. The area of cross-section of the channel is referred to as its width, $m(C(R(i), R(i+1)))$.

The micro-pump's capacity would be such that the unit of volume it could pump in a unit of time would be proportional to the area of cross-section, $m=\pi r_c^2$, of the channel in which it is pumping. To complete a full operation of the RDNA model would require moving the contents of a chamber to another one. The time taken for this would thus be inversely proportional to the channel width, proportional to the volume of the chamber and would have a latency introduced to get the channel filled and emptied. We could thus formulate the time to transfer the contents from one chamber to another as

$$T=\frac{h_c \times w_c \times d_c}{c \times m}+c'l_c.$$

As in VLSI models, the factor c' is initially treated as 0 since the term is miniscule relative to the other one.

Note that a channel can connect two reagent interfaces of two chambers vertically over each other, and this is what allows transmission between layers. As in the case of VLSI design, channels transporting the raw material of the computation's encoding and reagents, the analogue of power lines in a VLSI chip, may eventually be the most significant constituent as judged by area or volume as a metric.

4.1.3. Control lines

Since all the MEMS devices are controlled by a single micro-processor, there must be lines of control from each of them. These pass through conduits bored into the solid. Laying these out should not pose a significant problem in early implementations as there are three degrees of freedom for a path from each device. It may become a constraining factor as in the case of VLSI when significant miniaturization of the chambers occurs.

4.2. Specialized layers

Will now describe certain layers that have a specialized design.

4.2.1. Material input and output

One type of layer specially adapted would be the *material layer*. It would consist of the large reservoirs containing the various fluids needed for the operation of the MF-BMC. These would be located at the surface so that they may be easily replenished. Reservoirs containing various constituents such as the building blocks for the data in solution form, the reagents for the RDNA operations, the liquid used to flush a chamber or channel to clean it out after it has been used, and the other agents such as gel for electrophoresis, are all stored in this layer. By placing them in the topmost layer, the implicitly higher potential energy that they will have. can be exploited for micro-pumping and flushing purposes. There will be a finite number of reservoirs accessible from this layer. Controlling each one's content's flow will be a micro-pump and a micro-valve, with the expected notation. For example the flushing liquid's reservoir would be controlled by the micro-valve $V_{\text{flush}}(\Pi(\Pi_{\max}), R(i)$ and the micro-pump $\Pi_{\text{flush}}(\Pi(\Pi_{\max}), R(i)$ on the assumption that the reservoir is in the topmost plane and in chamber i in this plane.

4.2.2. Routing layer

While the layer described above is completely general and allows for routing of strands between any two chambers, we propose special form *routing layers* which can be used for optimizing routing. We will not exploit the properties of such layers in the remainder of the paper, but their utility in a practical MF-BMC machine is still valid. Such layers would be solid with three types of exceptions.

The first consists of channels laid out for the purpose of routing reagents, reactants and other chemicals needed in the general RDNA layers. The layout would be a grid with multi-way microvalves at the crossing points of the various conduits. Such a system would allow very efficient transfer of material along a path since it does not have to pass through the ordinary chamber at inter-connection points. It would have the disadvantage of not allowing the concurrent transport of materials along two paths that cross at any interconnect since the absence of chambers means the contents can not be intelligently sorted out.

The second exception is the association with each inter-connect point of a fixed number of *buffers*, each of which is a cavity, that are connected to the crossing point via a multiway microvalve that allows flow to be redirected into a buffer for temporary storage. These would act as analogues of scratch registers in a silicon based computer, as they would provide fast intermediate storage.

The final provision would be for vertical columns that would allow the passage of control lines from the MEMS devices as present in the general form layers.

4.3. Using the MF-BMC

The two aspects of the MF-BMC that must be controlled by the programmer are the data input and output and the sequence of operations to be performed upon it. In the sections below we describe how this is to be done.

4.3.1. Data input

The user of the MF-BMC will be able to provide it with data using three different methods. These will be summarized below.

4.3.1.1. 2D strand arrays. The first method is the use of DNA-on-a-chip technology (Suyama, 1998) if copies of particular strand encodings are required as input. The single stranded template for these strands can be replicated using a DNA-chip as follows. The template strands are biotinlylated. The chip will have a *solid support* base populated with fixed DNA sequences. DNA strands which will serve as the templates for the data being constructed and with prefixes that are Watson–Crick complementary to the fixed solid support sequences will be introduced. The complementary regions are allowed to anneal. Now there exist a population of templates held down to the chip. Many small sub-sequences of all possible combinations are introduced and appropriate primers are introduced. Annealing that makes the single stranded templates double stranded is allowed to occur. After flushing extra sub-sequences out, ligase is introduced. Next, by denaturing and then washing over magnetic beads with avidin, the template strands can be immobilized. The desired data is then routed for use and the templates can be recovered from the beads. By introducing suitable template strands from a library of stored strands, data strands can be created. Additional techniques using specific enzymes to extract only desired strands from those grown as well as the use of only certain primers allows finer grain control over the production of data strands.

4.3.1.2. Synthesizers. The above technique's limitation is that strand length must be limited and complex structures can not easily be created. The next level is the incorporation of a unit that is able to build an arbitrary strand from an abstract specification. Synthesizers of this sort are widely commercially available. They work by chemically adding single nucleotides to the strand being synthesized.

4.3.1.3. External fabrication. The above methods can not deal with arbitrary structure creation when complexes such as the 2D tiles made of DNA are needed. This leads to the last option which is the possibility for external DNA data manufacturing, followed by its incorporation into the MF-BMC by its introduction at any one of the unused reservoirs in the material layer described in Section 4.2.

4.3.2. Data output

We will assume that at the end of the computation the results will be encoded in what we term *output strands*, and that these will be routed to a *data output chamber* that could be one of the reservoirs in the material layer, or any other location designated and equipped as described below. In general, bioanalytical sensors are implemented as the coupling of chemical transduction, the process of selectively responding to particular compounds, to energy transduction, the resulting of a defined detectable pattern. The chemical transduction processes fall into one of four categories, depending on whether they are based on adsorption, entrapment, cross-linking, or covalent bonding (Cunningham, 1998).

4.3.2.1. 2D DNA chips. As with data input, the first method of reading data output is based on the use of DNA-on-a-chip technology. There are several different implementations available from Affymetrix, HySeq and other companies, as well as work done by research groups at universities such as the one at the University of Wisconsin. A standard technique is to have all data strands tagged with phosphorescent material. A 2D grid of different sequence strands are attached to known locations. The output data strands are washed over and when a complementary strand binds, the optical readout at the point where the strand attaches is read and processed by the microprocessor. Since it knows the mapping from sequences to locations on the chip it can deduce the complementary strand's sequence. While there is a vast literature available on the field, what is of particular interest are the techniques that can handle high throughput, as described in various studies (Pease et al., 1994; Blanchard et al., 1996; Chee et al., 1996; Fodor et al., 1991).

4.3.2.2. External analysis. Some strands washing through may not have their complements encoded in the library of the DNA strands available, or the strand may be far beyond the length encoded in the library. For such strands, we can have a sequencing unit connected to an output chamber, which is capable of analyzing, nucleotide by nucleotide, the content of a sequence. Since the output may be non-linear in some cases, such as tiles of DNA, and novel methods may be needed for readout, there will be a fallback option routing to a reservoir where external access is easy and the contents may be extracted and further analyzed with other machines.

4.3.3. Control

A micro-processor is located at a convenient location in the MF-BMC, with all the various control lines from the embedded MEMS in the various layers connected to it.

The micro-processor can control the opening and closing of the valves and the activation and deactivation of the actuators and pumps used for routing the pools of strands from one chamber to another. It can adjust the temperature of a chamber through a heater, the magnetic field through solenoids, and the electric field through electrodes. It can read the temperature of a chamber, the presence of flow in a channel, and the optical density of the DNA in a chamber. It can cause data strands to be generated, and it can read the sequence of an output strand. It can perform any of the other tasks previously described. It is therefore at the micro-processor that a programmer of the MF-BMC needs to implement an algorithm. The various primitives and the notation needed for the BMC and the micro-flow operations have all been provided as the MF-BMC was described. What remains to be stated is that the micro-processor has the capability to perform conditional branching depending on the readings it can make as has been described thus far.

4.3.4. Pipelining

A great strength of our model is that parallel transport between various volumes in which RDNA operations are occurring is feasible practically. An immediate result is that it is possible as well as desirable to carry out *pipelining* of the computation. The idea was first described in the context of the stickers machine (Roweis et al., 1996). For example during the implementation of the graph three coloring problem (Adleman, 1996), Adleman uses several separate operations on the results of previous separate operations, removing strand encodings of graphs that do not have the three coloring property. We can split all the strands encoding the graphs and their coloring, introducing only a small fraction into a chamber making the rate at which the operations can be performed in the chamber much higher due to the limited volume. As soon as the result of one separate is available from one chamber, it can be passed to the next chamber where it can undergo the next separate operation, while in the previous chamber the next batch of encodings is undergoing the same operation. This method allows the data set to be split and operated on with lower volumes and thus better DNA computation efficiency, yet no asymptotic increase in complexity of the algorithm. The only loss is the latency involved in filling and emptying the pipeline.

Finally, we note that, even though our model suggests that valves and pumps have a binary nature and can be only open or closed, and active or inactive, it is easy to modify it to allow for intermediate states whereby peristaltic transport of the strands along a chain of chambers can occur. This can be seen to have immediate application in the problem of having a large data set encoded with the destination chamber's address as a prefix on each strand. In this case, by the use of complementary strands affixed by magnetic beads or their equivalent in each chamber with the chamber's address on them, the data set can be systolically routed through all the chambers and the data for each chamber will be extracted from the solution as it flows by since it will anneal to the address encoding laid out for it.

5. Recombinant DNA operations in a chamber

The goal of the RDNA model was to provide computer scientists with an abstract interface to some of the possible operations available in a laboratory setting. The power and limitation of such models has been explored (Winfree et al., 1996) in connection with the various proposed operations. The operations that have been chosen are those that can be easily effected empirically based on established techniques and tools of the molecular biologist. We adapt them here for use in the context of chambers instead of test-tubes. The details of some operations, such as the specific primers needed in a PCR, are dependent on the specific encoding of the data, and are therefore to be specified by the algorithm designer.

A set of operations was proposed bv Adleman (1996) and Lipton (1996). These were extended by Reif (1998a) based on feasible operations available at the work bench in a molecular biologist's laboratory. We describe them along with possible implementations in our MF-BMC framework below. Instead of the operations acting upon test-tubes, the ones used in an algorithm effected on a MF-BMC computer operate on chambers. Instead of output being in designated test-tubes, they are in designated chambers.

Since all the operations that will be used below and the domains upon which they operate have been stated explicitly in Section 4, we shall refer to them by the generic terms rather than the formalisms introduced.

5.1. Execution of Adleman model operations with MF-BMC

Merge operations represent mixing the contents of two test-tubes in the laboratory into one tube in the Adleman model. Temperature and solution concentration is assumed to be such that the contents do not interact chemically. This is easily adapted to mixing the contents of two chambers bv routing the pool in the first chamber to the second chamber, where after emptying in it is caused to mix by the slight application of movement effected by a micro-actuator of the form described in Section 3. The operation's semantics are extended to allow merging into an empty chamber and to an output interface for various purposes such as external storage.

Copy operations in the Adleman model represent the replication of the contents of a test-tube, as would be performed using a polymerase chain reaction (PCR) in the laboratory (Campbell, 1993). The assumption is made that the protocol does not introduce significant mutation errors in the contents during replication, and that the relative concentrations remain the same-that is that the PCR replicates all strands equitably. We adapt this as follows. From the reservoirs in the material layer, the salt buffer, the DNTPs (deoxynucleotide-tri-phosphates) that are the free nucleotide building blocks, the two primers, the ligase, and the polymerase are micro-pumped into the chamber where the copy operation is occurring. Using a micro-heater created with a calibrated resistor and a micro-sensor together setup in a feedback loop as described in Section 3, the temperature of the chamber is regulated to a desired level. The sensor's sensitivity is set by the microprocessor in the control layer. By cycling repeatedly through temperatures, around 95°C to denature the strands, around 55°C for the primers to bind, and around 75°C for the polymerase to build along the template strands, the PCR can be effected in the chamber.

Detect operations in the Adleman model represent checking whether there are any DNA complexes present in a given test-tube. In the lab, a PCR would be run to amplify trace quantities of any interesting poly-nucleotide sequences that may potentially be present (Adleman, 1996). In the chamber, a PCR is performed as described above. After that, a micro-emitter on one side of the chamber coupled with a micro-sensor on the other side, and controlled by the micro-processor, is used to obtain an OD (optical density) reading that reveals whether any significant quantity of DNA strands are present in the chamber.

Separate operations in the Adleman model correspond to using an appropriate reagent in the laboratory to detect the presence of a sequence of nucleotides in the strands in a test-tube, and then separate the resulting two classes of strands into two separate test-tubes, those containing the subsequence and those devoid of it. This operation is effected by the introduction of micro-scale magnetic beads which have the complementary encoding embedded on their surface. Strands in solution that contain the target subsequence will adhere to the beads. By the creation of the appropriate magnetic field in the chamber the beads can be made to sink while the remaining solution can be pumped out. By increasing the temperature as needed and deactivating the magnetic field, the target strands can be denatured from the beads. By application of the magnetic field the beads can be immobilized again, and the target strands can be routed to the next chamber. The magnetic beads can be removed by deactivating the field and then washing them out in solution.

5.2. Extended model operations

We now discuss the RDNA model, which includes all the above operations of the Adleman model, as well as the operations discussed below.

Select operations in the RDNA model correspond to finding all strands that contain more than a certain number of base pairs. This is done in the lab by using gel electrophoresis to separate them by weight, since heavier sequences move less in a fixed time period and electric field than do lighter ones. If the length of the chamber permits the separation of the DNA complexes, then a procedure that is the analogue of the above can be executed. We emphasize that there exist significant technical hurdles to effect this. By introducing enough heated liquid agarose electrophoresis gel into the chamber to completely disconnect the buffer near the inlet of the chamber from the buffer near the outlet of the chamber, and allowing the gel to cool to the required temperature, a regular gel electrophoresis setup is effected. We note here that a chamber in which such an operation needs to be performed will have certain lower bounds on its length. This is due to the fact that the migration of the DNA complexes is inversely proportional to the logarithm of its molecular weight. By pumping, with a micro-pump as described in Section 3, the DNA strands to be operated upon into the buffer solution near the inlet of the chamber, then proceeding to activate the current through the electrodes near the inlet and the outlet so as to cause the strands to migrate through the matrix of the gel at a rate inversely proportional to their size, and stopping the current at a time that corresponds to the cut-off in size that is required, the set of DNA strands in the buffer solution near the outlet of the chamber correspond to all those below the specified threshold. These may be micro-pumped to their required destination. The routing can be done as described in Section 6. The current is then reactivated for a period of time sufficient to allow the remaining strands to travel through the matrix. These are also collected in the buffer at the outlet. They may then be micro-pumped to their selected destination. Finally, the electrophoresis gel is removed by raising the temperature in the chamber with a micro-heater to a point where the gel is fluid enough and then pumping it out.

If the chamber length is restricted, then alternative techniques such as ultra-filtration (Blatt et al., 1968) can be used. Here, the difference in molecular weight of the DNA, coupled with the viscosity of the solution, would result in differential rate of flow of the complexes when transported using a micropump. The smaller complexes would be moved more swiftly. This procedure would have lower reliability and accuracy, but could be executed within constrained dimensions. The membranes used as filters would be capable of differentiating between complexes which are different by an order of magnitude. Alternatively, other recent developments in the form of capillary electrophoresis, micro-dialysis, and miniaturized ultra-filtration can be used (Robinson and Justice, 1991; Baker, 1996; Schneiderheinze and Hogan, 1996).

Cleave operations in the RDNA model correspond to using an enzyme that recognizes a nucleotide sub-sequence and makes a nick in a known way (Reif, 1998a). We work with a simplified model instead of the more complex space of possibilities that can occur in a laboratory setting by limiting the types of enzymes used. This operation is effected in the chamber easily by merely pumping in the required enzyme.

Denature in the RDNA model refers to the process of causing double stranded nucleotide sequences to separate into their respective single strands. This has been extensively studied in the laboratory (Hartemink and Gifford, 1997; Rose et al., 1997; Wetmur, 1997). The change in the Gibbs Free Energy is the change in the enthalpy of the system less the change in the Standard Free Energy of the system scaled by the absolute temperature. Thus, by regulating the conditions such as the temperature and concentration of the reactants in solution, the rate at which the process of denaturing occurs can be controlled. By the application of heat with resistor based micro-heaters and micro-sensors to regulate the temperature, denaturing is effected.

Anneal in the RDNA model refers to the process of single strands which have complementary base pairs combining by the formation of hydrogen bonds between the respective complementary base pairs. It has been modeled in the literature (SantaLucia et al., 1996; Hartemink and Gifford, 1997; Wetmur, 1997) with the energetics explained above in the denature operation. It is effected by the control of various conditions such as temperature, concentrations, and reagents. By introducing the required buffers and reagents, and by altering the temperature, annealing can be effected in a chamber. The control of the parameters is necessary since potentially there are many competing strands which may anneal but we wish only the ones designed to match exactly to do so (Winfree, 1998).

Ligation in the RDNA model is the closing up along the back-bone of open bonds between consecutive bases in a strand. It is effected by pumping ligase into chamber where the operation is occurring.

6. Strand routing

'Routing' between strands is said to occur when the two strands that need to interact biochemically have done so. Routing in a chamber becomes a concern when operations that need strings to interact are used, such as anneal or ligate in the RDNA model. Reif added the parallel associative match (PA-Match) operation to Adleman's original model. The Parallel Associative Match (PA-Match) operates on sets of strings encoded in DNA (Reif, 1998a). Assume that a string from the first set using the sets' encoding is represented as $E(\alpha)E(\beta)$. Assume another string from the second set using the sets' encoding is represented as $E'(\beta')E'(\gamma)$. Then the PA-Match operation transforms all such pairs where β and β' are Watson–Crick complementary sub-sequences. The result of any such pair undergoing the operation would be a new sequence of the form $E''(\alpha)E''(\gamma)$. If in a set of N strands there was only one copy of each of two strands, then obtaining the desired result with fixed probability would require $O(N^2 \log N)$ total volume with $O(N \log N)$ copies of each of the strands) and $O(1)$ time. (This is explained in Section 2.) We will see how this is improved in the MF-BMC later.

Therefore, if there are N strands in a chamber, and two particular strands need to interact with each other. then since the initial rate of interaction is proportional to the product of the concentrations of the two strands, guaranteeing with high probability that the two strands will interact in fixed time can only be achieved by the use of $O(N \log N)$ volume. By limiting the total number of distinct strands in the chamber in which the reactants are to N/n, the chance of two particular reactants interacting is thereby increased. Now only

$$O\left(\left(\frac{N}{n}\right)^2 \cdot \log \frac{N}{n}\right)$$

volume is needed. When there exists a method by which N strands can be separated into n chambers, then the total volume needed drops from the original $N^2 \log N$ to

$$O\left(\frac{N^2}{n} \cdot \log \frac{N}{n}\right)$$

for a fixed time operation.

If we have an operation that previously used 10^{10} strands of length 10^3, since it would take $O(N^2 \log N)$ volume or roughly on the order of $10^{2(10)} \times \log \quad 10^{10} \times 10^3 = 2.3 \times 10^{24}$ nucleotides, this would not be feasible. However, if we envision the use of the MF-BMC framework with 10^6 chambers, then the required volume drops to

$$O\left(\frac{N^2}{n} \cdot \log \frac{N}{n}\right)$$

which would be on the order of

$$\frac{10^{2(10)}}{10^{6}} \times \log 10^{(10-6)} \times 10^{3} = 10^{18},$$

which is a feasible volume. In general, the use of n chambers will give a drop in cumulative volume by a factor of n.

A key issue would be what size of n would be optimal. A reasonable approach to solving this would be to parametrically estimate the amount of time it would take to pump the contents of a chamber across the diameter of the topology of chambers, say $T(D)$. Then we could parametrically estimate the amount of time it would take for annealing to occur in a chamber of volume v^3, say $T(v)$. By equating these two we could gain a parametric solution which would specify the number of chambers given individual chamber volume or vice versa. The idea is based on the intuition that if this equality held then operations in the MF-BMC framework could operate independently, without concern for bottlenecks caused by either RDNA operations or routing operations since each would take the same amount of time. This strategy is used below in the reduction of PRAM simulation.

6.1. Routing in the MF-BMC model

While synchronized movement of pools of DNA strands extends the possible algorithms that can be implemented, further independence of data relocation, at the strand level, would be even more useful. We thus introduce the notion of using *chamber addresses* which are prefixes attached to the DNA strands that can be created and altered by the steps in a BMC algorithm. On the assumption that all strands used in the course of the computation do contain such prefixes, we proceed to introduce the idea of independent strand routing along the lines of independent packet routing in a computer network.

At each stage of the algorithm, the contents that need to be routed from one chamber to another are sent by pumping them through the channels, using the routing protocol outlined. We note that as in the case of chamber operations, the programmer might need to flush the channels to be used such that residues are minimized.

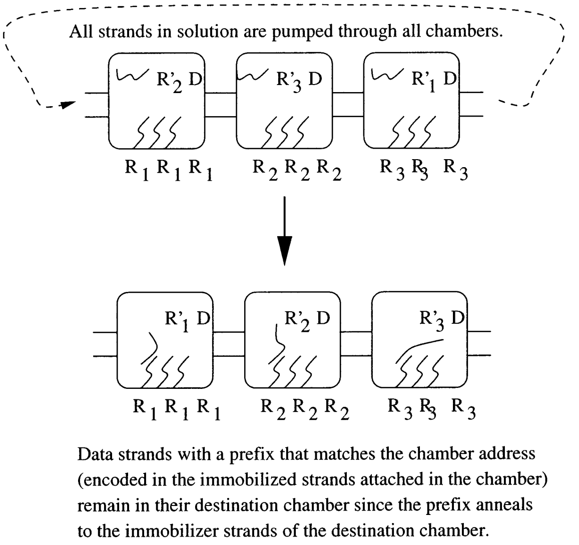
All strands in solution are pumped through all chambers.
R'2 D
R'3 D
R'1 D
R1 R1 R1
R2 R2 R2
R3 R3 R3
R'1 D
R'2 D
R'3 D
R1 R1 R1
R2 R2 R2
R3 R3 R3
Data strands with a prefix that matches the chamber address (encoded in the immobilized strands attached in the chamber) remain in their destination chamber since the prefix anneals to the immobilizer strands of the destination chamber.

Fig. 4. Data strands with a prefix that matches the chamber address (encoded in the immobilized strands attached in the chamber) remain in ther destination chamber since the prefix anneals to the immobilizer strands of the destination chamber.

6.2. Strand routing on a linear array

First we describe routing strands in a one dimensional array of chambers (Fig. 4). The essential strategy to perform routing of strands of DNA will be to assume that each is prefixed with the complement of the destination chamber address in the form of a nucleotide sequence. Then the strands are circulated through all the chambers.

If the destination chamber is $R(i)$, then the chamber has many immobilizer strands of type $s_{R(i)}$ as outlined in Section 4.1.1 and each strand has prefix $s'_{R(i)}$. If these encodings are different and long enough, and there are enough immobilizer strands in each chamber, then after circulating all the strands through all the chambers several times the strand's prefixes will have annealed to the immobilizer strands and they will be in the correct chambers. The chambers can then be isolated using the micro-valves between them. If there are n chambers then routing occurs in time $O(n)$.

6.3. Strand routing on a grid

Since we are using a two dimensional array of n chambers, we now address the issue of routing in this framework. It is possible to have a distinct address for each chamber and to then circulate the strands through the chambers. However, we can improve upon this by exploiting the topology. If the destination chamber is $R(i, j)$, then we assume that the prefix encoded is of the form s_i, s_j, where all chambers in row i will have the base of the immobilizer strand encoded with the sequence s_i and all chambers in the column j will have the sequence immediately contiguous to the base sequence encoded with the sequence s_j. By first introducing the entire volume of strands and fluid to be routed into the MF-BMC, so that all the chambers are filled, we can effect movement of all strands to random nodes. Next, the rows of chambers are isolated by closing all the micro-valves along the channels that connect chambers in different rows. By routing strands through a single row of chambers and letting their prefixes anneal to the immobilizer strands, we can effect routing of strands to the correct chamber in the correct column. This is analogous to routing in a linear array of chambers. Next all the micro-valves in channels connecting chambers in different columns are closed, and the micro-valves in channels connecting chambers in different rows are opened. The strands are circulated along the columns of chambers. Since the prefixes of all strands match the prefix at the base of the anchor strands, the next segment is the portion that is required to match now. By raising the temperature marginally, only strands where the entire prefix s_is_j match will remain annealed while strands with only s_i will break away and be circulated through the various chambers till they anneal to the anchor strands in the chamber that matches both parts of the prefix encoded on them. Since there are $\sqrt{n}$ chambers along each stage of routing, routing occurs in time $O(\sqrt{n})$.

7. Applications

This MF-BMC framework is particularly useful on account of the fact that it lends itself to direct application of known algorithms developed in several other fields. Examples are described below.

We should note that the strategy of creating all possible solution encodings in vitro and then performing various laboratory operations on all the strands of DNA necessarily has limited scope on its own, since the volume of DNA required grows exponentially in the size of the problem. The obvious refinement needed is to develop strategies that operate on a fixed size section of the solution space at any given point in time. This can be achieved in one of two ways. *Solution segmentation* refers to the idea of devising an algorithm to solve fixed size sub-parts of the solution at a time, thereby obtaining segments of the ultimate solution at each stage. For certain hard problems, such as certain *NP-complete* ones that have been chosen and attacked in the laboratory (Adleman, 1994; Ouyang et al., 1997), this is impractical as the inter-dependence between segments of the final solution may preclude this. The alternative approach is to attack with *solution space segmentation*, which is the division of the solution space into independent sections that are then attacked separately.

7.1. Database operations

By exploiting the ability to store large amounts of information in limited volumes using DNA, it is conceivable that databases would be stored and need to be operated upon and accessed. One of the operations used on databases is the Join operation and its implementation using PA-Match was described by Reif (1998a). The Join operation is used to combine multiple databases with different relations defined into a single database with a new set of relations defined wherever possible. Since Join was effected using PA-Match which relies on annealing, using the MF-BMC would allow significant volume reduction as described.

7.2. PRAM algorithms

By treating chambers in the model as processors, we can apply any of the algorithms in the literature (Jaja 1992; Reif, 1993) developed for the parallel random access machine (PRAM) model. Further, systolic algorithms originally developed for grid based frameworks in the context of VLSI circuits, can be applied within the framework of the model we described, resulting in better performances than with the regular RDNA model.

7.3. PRAM simulation

Reif showed how to simulate a CREW PRAM, which is a machine with P processors that can all concurrently read a large shared memory of size M and each processor is free to exclusively write a memory location as long as no other processor attempts to write the same location at the same time in the PAM model (Reif, 1998a). If the PRAM had time bound U and used strings of length $s = O(\log(PM))$, then it was shown that simulation used at most $O(U + s)$ PA-Match operations and $O(s \log s)$ PAM model operations. This was done by modeling the state of the PRAM as a strand encoding and the state transitions that applied arithmetic operations on pairs of memory locations, accesses of memory locations for reads, and accesses for writes, as a sequence of operations in the PAM model. Reif also showed how PA-Match operations could be implemented in the RDNA model with a slowdown of $O(s)$ (Reif, 1998a). Although time efficient, the simulation had the drawback that the volumes required increased quadratically with the number of PA-Matches needed.

We can further improve on this model by simulating a CRCW PRAM, one which has the added power of being able to concurrently write a single memory location, with conflicts arbitrated according to some arbitrary priorities. This can be done with slowdown of a factor of $\log P$. This can be effected using a parallel pipelined merge sort to resolve the mapping between source and destination memory locations and processors as described by Jaja (1992) for example.

In the MF-BMC model, operations that occur between chambers require a linear increase in volume since the routing algorithms can effect an exact relocation, while operations that occur within a single chamber will still require the quadratic volume to achieve reasonable probabilities of occurrence, but with the volume now being that of an individual chamber rather than of the entire MF-BMC. Thus if the original model dealt with N different strands, the volume required would be $O(N^2 \log N)$, while if there are n chambers, then the volume requirement would drop to

$$O\left(\frac{N^2}{n} \log \frac{N}{n}\right).$$

Since the time for access to a strand anywhere in the MF-BMC would be $O(D)$, where D *is* the diameter of the chamber layout, it is reasonable to allow reactions to run for time proportional to $O(D)$, which might reduce the strand replication needed in each chamber. Optimally, that is when the reactions that run to equilibrium result in as much of the product as the reactants, this would result in a volume requirement of

$$O\left(\frac{n}{D} \times \left(\frac{N}{n}\right)^2 \log \frac{N}{n}\right).$$

Thus PRAM simulations could then be executed in the MF-BMC with lower and practical volumes. In the case of a grid based topology, the requisite volume would be

$$O\left(\frac{N^2}{n^{3/2}} \log \frac{N}{n}\right).$$

operating in $O(\sqrt{n})$ time. (If we restricted operations to $O(1)$ time, then we would need volume

$$O\left(\frac{N^2}{n} \log \frac{N}{n}\right).$$

We can compare this to the previous result by Reif (1998a) of $O(N^2 \log N)$ volume for $O(1)$ time. The result of Ogihara and Ray (1997) for satisfiability of Boolean circuits. from which PRAM simulation can be deduced, was independently arrived at and in agreement with Reif's. Finally, we note that recent work (Amos et al., 1998) that reduces the volume needed for PRAM simulation by the use of a representation of gates in DNA still relies on annealing and can benefit from using our model to reduce the cumulative volume required as with the above methods.

8. Lower bounds on space and time

Thompson (1980) developed lower bounds on area and time in the context of very large scale integration (VLSI) design. We can adapt these to gain further insight into the scale of the MF-BMC constructs. We can use them to help determine how many chambers we would need, what these chambers' individual volumes should be, what the channel width must be, and the amount of time that is required to execute a particular computational task.

Several lower bounds are developed in the manuscript (Gehani and Reif, 1998). We summarize them here. Here *problem* and *information content* have technical definitions from VLSI lower bounds (Ullman, 1984). We show, with certain assumptions, the memory capacity B of the MF-BMC. We lower bound this by, the log d, where d is the number of distinct configurations in the MF-BMC. If T is the time to execute an algorithm, then BT is bounded from below by the largest of the input/output representations used in solving a problem. Finally, the product BT^2 is bounded from below by I^2, where I is the informtion content of the problem being solved.

Acknowledgements

Supported by grants NSF CCR-9725021 and CCR-96-33567 and Army Research Office Contract DAAH-04-96-1-0448. Prepared for DNA4 Proceedings.

References

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

Adleman, L.M., 1996. On constructing a molecular computer. In: Baum, E.B., Lipton, R.J. (Eds.), DNA Based Computers, volume 27 of DIMACS: Series in Dascrete Mathematics and Theoretical Computer Science. American Mathematical Society, 1996 (based on Manuscript, Computer Science Department, University of Southern California, January 11, 1995).

Amos, M., Gibbons, A., Dunne, P.E., 1997. The complexity and viability of DNA computations. In: Olsson, B., Lundh, D., Narayanan, A. (Eds.), Bio-Computing and Emergent Computation. World Scientific, 1997.

Amos, M., Dunne, P.E., Gibbons, A. Efficient time and volume DNA simulation of CREW PRAM algorithms. Technical Report CTAG-98006, Department of Computer Science, University of Liverpool, UK, Alay 1998.

Adey, R.A., Lahrmann, A., LeBmollmann, C. 1995. (Eds.), Simulation and Design of Micro systems and Microstructures. Computational Mechanics Publication.

Baker, D.R., 1996. Capillary Electrophoresis. Wiley, New York.

Blanchard, A.P., Kaiser, R.J., Hood, L.E., 1996. High-density oligonucleotide arrays. Biosensor Bioelec. 11, 687–690.

Blatt, W.F., Robinson, S.M., Bixler, H.J., 1968. Membrane ultrafiltration: The diafiltration technique and its application to microsolute exchange and binding phenomena. Anal. Biochem.

Campbell, 1993. Biology. Benjamin/Cummings Publishing Company.

Cunningham, A., 1998. Introduction to Bioanalytical Sensors. Wiley, New York.

Chee, M., Yang, R., Hubbell, E., Berno, A., Huang, X.C., Stern, D., Winkler, J., Lockhart, D.J., Morris, M.S., Fodor, S.P.A., 1996. Accessing genetic information with high-density DNA arrays. Science 274, 610–614.

Fodor, S.P.A., Read, J.L., Pirrung, C., Stryer, L., Lu, A.T., Solas, D., 1991. Light-directed spatially addressable parallel chemical synthesis. Science 251, 767–773.

Gehani, A., Reif, J., 1998. Micro-flow bio-molecular computation-extended manuscript (available upon request from gehani@cs.duke.edu).

Hartemink, A.J., Gifford, D.K., 1997. Thermodynamic simulation of deoxyoligonucleotide hybridization for DNA computation. In: Wood, D. (Ed.), Proc. 3rd DIMACS workshop on DNA Based Computers, University of Pennsylvania, June 23–25, 1997, DIMACS: Series in Discrete Mathematics and Theoretical Computer Science, Providence, RI, June, 1997, American Mathematical Society, pp. 15–25.

Ikuta, K. 1997. 3D-micro integrated fluidic system toward living LSI. In: Langton, C.G., Shimohara, K. (Eds.), Proc. 5th International Workshop on Artificial Life: Synthesis and Simulation of Living Systems (ALIFE-96), Cambridge, May 1997. MIT Press, Cambridge, MA, pp. 17–24.

Jaja, J., 1992. An Introduction to Parallel Algorithms. Addison-Wesley, Reading, MA.

Kurtz, S.A., Mahaney, S.R., Royer, J.S., Simon, J. Active transport in biological computing. In: Proc. 2nd Annu. Meet. DNA Based Computers, Princeton University, June 10–12, 1996., Vol. 44, DIMACS: Series in Discrete Mathematies and Theoretical Computer Science. American Mathematical Society, pp. 171–180.

Thomson Leighton, F., 1992. Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes. Morgan Kaufmann, San Mateo, CA.

Lipton, R.J., 1996. Speeding up computations via molecular biology. In: Baum, E.B., Lipton, R.J. (Eds.), DNA Based Computers, Vol. 27, DIMACS: Series in Discrete Mathematics and Theoretical Computer Science. American MathematicalSociety, pp. 67–74.

Manz, A., Effenhauser, C.S., Burggraf, N., Harrison, D.J., Seiler, K., Fluri, K., 1992. Electroosmotic pumping and electro-osmotic pumping and electro-phoretic separations for miniaturized chemical analysis systems. J. Micro-mech. Micro-eng. 4, 257–265.

Ouyang, Q., Kaplan, P.D., Liu, S., Libchaber, A., 1997. DNA solution of the maximal clique problem. Science 278, 446–449.

Ogihara, M., Ray, A. 1997. Simulating Boolean circuits on a DNA computer. In: Proc. 1st Int. Conf. Computational Molecular Biology, New York, January, 1997. ACM Press, pp. 226–231.

Pease, A.C., Solas, D., Sullivan, E.J., Cronin, M.T., Holmes, C.P., Fodor, S.P., 1994. Light generated oligonucleotide arrays for rapid DNA sequence analysis. Proc. Natl Acad. Sci. USA 91, 5022–5026.

Rose, J.A., Deaton, R., Garzon, M., Murphy, R.C., Franceschetti, D.R., Stevens, S.E. Jr., 1997. The effect of uniform melting temperatures on the efficiency of DNA computing. In: Wood, D., (Ed.), Proc. 3rd DIMACS Workshop on DNA Based Computers, University of Pennsylvania, June 28-25, 1997, DIMACS: Series in Discrete Mathematics and Theoretical Computer Science, Providence, RI, June 1997. American Mathematical Society, pp. 35–42.

Reif, J., 1993. (Ed.), Synthesis of Parallel Algorithms. Morgan Kaufmann, 1993.

Reif, J., 1997. Local parallel biomolecular computing. In: Wood, D., (Ed.), Proc. 3rd DIMACS Workshop DNA Based Computers, University of Pennsylva nia, June 23–25, 1997, DIMACS: Series in Discrete Mathematics and Theoretical Computer Science, Providence, RI, June, American Mathematical Society, pp. 243–264.

Reif, J., 1998a. Parallel molecular computation: Models and simulations., Algorithmica, Special Issue on Computational Biology.

Reif, J., 1998b. Paradigms for biomolecular computation. In: Calude, C.S., Casti, J., Dinneen, M.J. (Eds.), Unconventional Models of Computation. Springer, Berlin.

Robinson, T.E., Justice, J.B., Jr. 1991. Microdialysis in Neuroscience. Elsevier, Amsterdam.

Roweis, S., Winfree, E., Burgoyne, R., Chelyapov, N.V., Goodman, M.F., Rothemund, P.W.K., Adleman, L.M. 1996. A sticker based architecture for DNA computation. In: Proc: 2nd Annu. Meet. DNA Based Computers, Princeton University, June 10–12, 1996, Vol. 44, DIMACS: Series in Discrete Mathematics and Theoretical Computer Science. American Mathematical Society, May 1996, p. 1–30.

SantaLucia, J. Jr., Allawi, H., Seneviratne, P.A., 1996. Improved nearest-neighbour parameters for predicting DI\-A duplex stability. Biochemistry, 35: 3555–3562.

Shoji, S., Esashi, M., 1992. Micro-flow devices and systems. J. Micro-mech. Mcro-eng. 4, 157–171.

Schneiderheinze, J.M., Hogan, B.L., 1996. Selective in vivo and in vitro sampling of proteins using miniature ultrafiltration sampling probes. Anal. Chem. 68, 3758–3762.

Suyama, A. 1998. DNA chips-integrated chemical circuits for DNA diagnosis and DNA computers (in press).

Thompson, C.D., 1980. A complexity theory for VLSI. PhD thesis, Carnegie Mellon University, 1980.

Ullman, J.D., 1984. Computational Aspects of VLSI. Computer Science Press, Rockville, MD.

Verpoorte, E.M.J., vanderSchoot, B.H., Jeanneret, S., Manz, A., Widmer, H.M., deRooij, N.F., 1992. Three-dimensional micro-flow manifolds for miniaturized chemical analysis systems. J. Micro-mech. Micro-eng. 4, 246–256.

Wetmur, J.G., 1997. Physical chemistry of nucleic acid hybridization. In: Wood, D., (Ed.), Proc. 3rd DIMACS Workshop on DNA Based Computers, University of Pennsylvania, June 23–25, 1997, DIMACS: Series in Discrete Mathematics and Theoretical Computer Science, Providence, RI, June 1997, American Mathematical Society, pp. 1–14.

Winfree, E. 1995. On the computational power of DNA annealing and ligation. In: Baum, E.B., Lipton, R.J. (Eds.), DNA Based Computers, Vol. 27 DIMACS: Series in Discrete Mathematics and Theoretical Computer Science. American Mathematical Society, May 1995.

Winfree, E. 1998. Simulations of computing by self-assembly. In: Proc. 4th DIMACS Meet. DNA Based Computinig, May 1998, pp. 213–242.

Winfree, E., Yang, X., Seeman, N.C. 1996. Universal computation via self-assembly of DNA: Some theory and experiments. In: Proc. 2nd Annu. Meet. DNA Based Computers, Princeton University, June 10–12, 1996, Vol. 41 DIMACS: Series in Discrete Mathematics and Theoretical Computer Science, AmericanMathematical Society, May 1996, pp. 191–214.