DROPLET ROUTING IN NANO-BIO-CHIPS

Satish Kumar Singh
M.Tech(CS), 2nd YEAR, MTC0714

Under The Supervision
of

Prof. B. B. Bhattacharya
Advanced Computing and Microelectronics Unit
INDIAN STATISTICAL INSTITUTE, KOLKATA

July 20, 2009

Contents

Acknowledgement

I would like to express my deepest gratitute to my supervisor Prof. B. B. Bhattacharya for his constant guidance and support throghout my dissertation work. I would also like to thank all the members of Advanced Computing and Microelectronics Unit for their constant support and motivation.
I am thankful to all of my family members for their constant support and encouragement without which it would have never been possible for me to persue my work. I would like to thank all of my colleagues and friends for their support.
I have no words to express my thanks to God who, I feel, always gives me strength and energy to accomplish any task.

Satish Kumar Singh
$16^{th} July$ 2009

Abstract

In this report we present an approach to coordinate the motions of droplets in digital microfluidic systems, a new class of lab-on-a-chip systems for biochemical analysis. A digital microfluidic system typically consists of a planar array of cells with electrodes that control the droplets. Basically this control is done with the help of ***ACTUATION SEQUENCES***[1]*. The primary challenge in using droplet-based systems is that they require the simultaneous coordination of a potentially large number of droplets on the array as the droplets move, mix, and split[2]. In this report we describe a method to route the droplets from their initial positions to their final positions. This is the scheme based on heuristic approach. Firstly, we try to move the droplet with minimum* ***MANHATTAN DISTANCES*** *between their source to destination position first, along with the subset of other movable droplets simultaneously. In effect it is* ***SHORTEST MANHATTAN DISTANCE FIRST***[2] *algorithm. The algorithm is greedy one. The elements of the subset are updated after each clock cycle. Sometimes the algorithm may lead to a configuration which is a* ***DEADLOCK*** *situation. In that case we backtack. For this reason, in this algorithm, there is provision of* ***BACKTRACKING****. This may, on one hand, increases the time of excution but on the other hand we can reach to solution if it exists. The algorithm have been able to successfully coordinate a good number of droplets simultaneously.*

[1]This is preprogrammed sequence of voltages to be applied on the electrodes.
[2]Discussed in Section 7.

1 Introduction

Microfluidic chips or Microelectromechanical sytems (MEMS) consists of *submillimetre* scale components such as channel, valves, pumps, microlitre of fluids, etc on a small $2D$ array of electrodes. One of the most advanced technologies to build a biochip[3] is based on microfluidics where micro/nano-liter droplets are controlled or manipulated to perform intended biochemical operations on a miniatured lab, so called lab-on-a-chip (LOC) [14]. The old generation of microfluidic biochip consists of several micrometer scale components including channels, valves, actuators, sensors, pumps, and so on. Even though this generation shows successful applications [20] like DNA probing, it is unsuitable to build a large and complex biochip, because it uses continuous liquid flows, as like continues voltages in analog VLSI design. The new generation of microfluidic biochip has been proposed based on a recent technology breakthrough where the continuous liquid flow is sliced or digitized into droplets. Such droplets are manipulated independently by an electric field. This new generation is referred to as a DMFB[4] (Figure 2). Due to such a digital nature of a DMFB, any operation on droplets can be accomplished with a set of library operations like VLSI standard library, controlling a droplet by applying a sequence of preprogrammed electric signals (e.g, Actuation sequences) [10]. Therefore, a hierarchical cell-based design methodology can be applied to a DMFB. Under this circumstance, we can easily envision that a large scale complex DMFB can be designed as done in VLSI, once strong CAD frameworks are ready. However, CAD research for DMFB design has started very recently. In [16], the first top down methodology for a DMFB is proposed, which mainly consists of architecture level synthesis and geometry-level synthesis.

Geometry-level synthesis can be further divided into module placement and droplet routing. During module placement, the location and time interval of each module are determined to minimize chip area or response time. Since different modules can be on the same spot during different time intervals based on reconfigurability [8, 11], module placement is equivalent to a $3D$ packing problem [32]. There are testing issues also involved with the entire techninique [5, 6, 7]. Meanwhile, in droplet routing, the path of each droplet is found to transport it without any unexpected mixture under design requirements. Similarly to module placement, a spot can be used to transport different droplets during different time intervals (simply

[3]Popularily known as Bio-MEMS.
[4]Digital Microfluidic Bio-chip.

in a time-multiplexed manner), which increases the complexity of routing. The most critical goal of droplet routing is routability as in VLSI [19], while satisfying timing constraint and maximizing fault-tolerance [14].

2 Biochip and Microfluidic Technology

A biochip is a collection of miniaturized test sites (microarrays) arranged on a solid substrate that permits many tests to be performed at the same time in order to achieve higher throughput and speed.There are two classes of biochips, till date, which are described as under :-

2.1 Static Biochips

The first generation biochips were based on the concepts of DNA microarray. The DNA microarray is a piece of glass, plastic or silicon substrate on which pieces of DNA are affixed in a microscopic array (Figure 1). These DNA segments act as probes. DNA probes helps in detecting genetic sequences of a biological sample simultaneously. Similar to the concepts of DNA micro array, a protein array is a very small scaled array, where large quantities of capture agents (like monoclonal antibodies) are affixed on the chip surface. These capture agents act as detectors and helps in determining the presence and/or amount of proteins in biological samples, e.g., blood. GeneChip DNA array from Affymetrix, DNA microarray from Infineon AG, NanoChip microarray from Nanogen are few DNA microarray technique based biochips available in the market. A major disadvantage of DNA and protein arrays is that once these chips are synthesized they are neither configurable nor scalable. Moreover, there is no facility to carry out sample preparation in this kind of biochips.

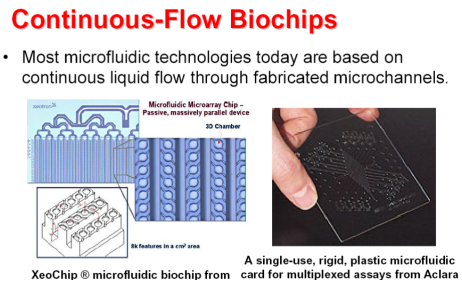

Figure 1: Continious Microfluidic Systems

2.2 Dynamic Biochips

Then comes the next generation biochips based on microfluidics. Microfluidics technology can be used to integrate all necessary functions for biochemical analysis onto one chip, i.e., microfluidic assay operations, detection and also the sample pretreatment and preparation can be done using the same chip. There are two kinds of microfluidic biochips available; continuous flow biochips and droplet based microfluidic biochips (Figure 3) .

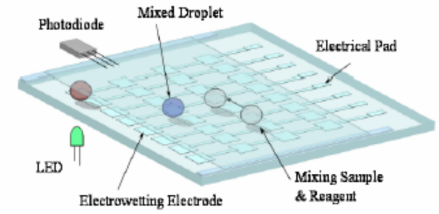

Figure 2: Discrete Microfluidic Biochip Systems

2.2.1 Continuous-Flow Microfluidics

As the name suggests, these technologies are based on the manipulation of continuous liquid flow through micro-fabricated channels. External pressure sources, integrated mechanical micropumps , integrated mechanical micropumps, or electrokinetic mechanisms are used for the actuation of liquid flow. Continuous flow chips are useful in carrying out simple biochemical applications which require less complicated fluid manipulation. But when the applications are more complex and require high degree of flexibility and complicated fluid manipulation, continuous flow chips become unsuitable. The fluid flow in a micro channel is governed by parameters like pressure, fluid resistance and electric field. These parameters vary along the flow path making the fluid flow at any one location dependent on the properties of the entire system. So it becomes very difficult to integrate and scale these kinds of microfluidic chips.The re-configurability in this type of chips is very poor because of the permanent etching of the microstructures. Because of the low reconfigurability the fault tolerance capability is very low.

2.2.2 Digital (Droplet-based) Microfluidics

Instead of continuous flow of liquid, here the liquids are divided into discrete and independently controllable droplets. The droplet based microfluidic chips are open systems in contrast to closed-channel continuous flow systems.There are two kinds of electrical methods; Dielectrophoresis (DEP) and electro-wetting-on-dielectric (EWOD) (Figure 12) are used for droplet actuation. DEP uses high frequency ac voltages while EWOD uses dc voltages to carry out droplet actuation. With the help of electro-hydrodynamics forces both the techniques provide high droplet speeds with relatively simple geometries. Liquid DEP actuation is defined as the attraction of polarizable liquid masses into the regions of higher electric-field intensity, as shown in Figure 6 . In DEP based microfluidics, patterned electrodes are placed on a planar substrate. The electrodes are coated with a thin dielectric layer. The electrodes are charged by applying ac voltage ($230 - 300V_{rms}$ at $50 - 200kHz$). EWOD technique uses dc voltage to control the interfacial energy between liquid and solid substrate. In comparison to DEP, there is almost negligible Joule heating occurs in EWOD because the dielectric layer covering the electrodes blocks dc electric current.

3 Digital Microfluidic Systems

More recently, there has been increased interest in microfluidic devices that handle discrete droplets, with volumes usually in the sub-microliter range. In these digital microfluidic systems DMFS[5], droplets are generated,transported, merged, analyzed, and disposed on planar arrays of addressable cells. This architecture for microfluidic systems is attractive because of (1) greater flexibility - analyte handling may be reconfigured simply by re-programming rather than by changing the physical layout of the microfluidic components;(2) high droplet speeds[2]; (3) no dlution and cross-contamination due to diffision and shearflow; and (4) the possibility for massively parallel microfluidic circuits.
Instead of continuous flow of liquid, here the liquids are divided into discrete and independently controllable droplets. The droplet based microfluidic chips are open systems in contrast to closed-channel continuous flow systems (Figure 4). There are two kinds of electrical methods; Dielectrophoresis (DEP) and electro-wetting-on-dielectric (EWOD) are used for droplet actuation. DEP uses high frequency

[5]Digital Microfluidic Systems

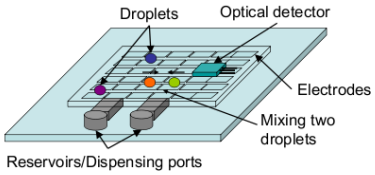

Figure 3: Dynamic Biochips

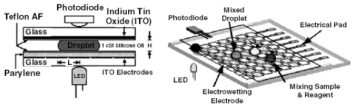

Figure 4: Schematic of a DMFB

ac voltages while EWOD uses dc voltages to carry out droplet actuation. With the help of electro-hydrodynamics forces both the techniques provide high droplet speeds with relatively simple geometries. Liquid DEP actuation is defined as the attraction of polarizable liquid masses into the regions of higher electric-field intensity, as shown in Figure 8. In DEP based microfluidics, patterned electrodes are placed on a planar substrate. The electrodes are coated with a thin dielectric layer. The electrodes are charged by applying ac voltage ($230 - 300V_{rms}$ at $50 - 200kHz$). EWOD technique uses dc voltage to control the interfacial energy between liquid and solid substrate. In comparison to DEP, there is almost negligible Joule heating occurs in EWOD because the dielectric layer covering the electrodes blocks dc electric current.

3.1 Droplet Transport

Small droplets can be moved across a planar surface effectively with a variety of techniques, for example with electric fields (e.g., [8]), the thermocapillary effect (e.g., [30]), light driven [12] electrochemical surface modulation or conformational changes in molecular surface layer (e.g., [16]). For the work in this report, droplet transport with electric fields is most suitable; hence we briefly discuss the two main techniques in this realm.

3.1.1 Dielecrophrosis

In dielcctrophoresis (DEP), neutrally charged objects are first polarized by an electric field, and then experience 3 net force due to the field. This force can only be non-zero if a field gradient

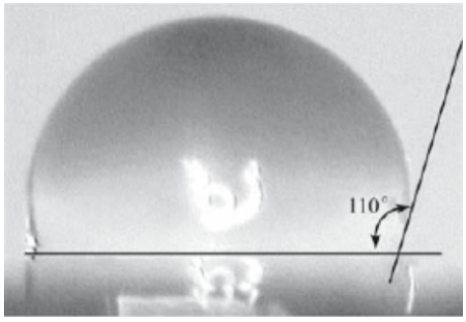

Figure 5: Contact Angle

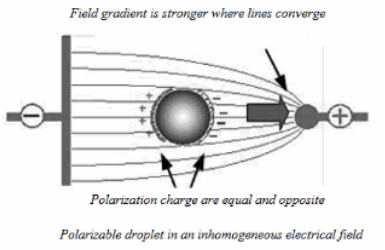

Figure 6: Polarization Of Droplets

exists, i.e., the positively and negatively polarized regions of the object occupy areas of different field strengths. If the object has stronger polarization than the surrounding medium then it is pulled towards the areas of higher field strength (this is called positive DEP), but if the surrounding medium has higher polarization, then the object is pushed towards areas of lower field strength (negative DEP). DEP can be considered the electrostatic analogy of induced magnetism[13]. Common examples for DEP are charged clothes that attract neutral lint particles.

3.1.2 Electrowetting

Electrowetting on dielectric (EWOD) exploits the decrease of contact angle that an aqueous droplet on a dielectric surface expriences when exposed to an electric field. If the field is localized at only one side of the droplet, then the difference in contact angle causes a pressure differential in the droplet, which drives it towards the region of higher field strength.

Electrowetting is a method for modifying the wetting properties of a surface. An electrostatic field is created when we apply voltage. Due to the creation of electrostatic field the interfacial tension between liquid-gas and liquid-substrate is changed. Because of the

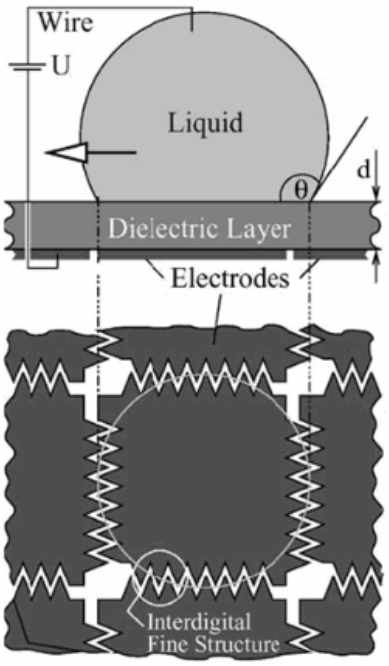

Figure 7: Techniques for Droplet movement

change in the interfacial tension due to application of electric voltage, the droplet shape is deformed. Deformation of droplets could be use for the fluidic operations like moving, mixing etc.

Physics of Wetting:- Lets we have a liquid L. If there are no external influences (that is no other phases are present around the liquid) then an amount of liquid, L will form spherical shape. This happens because of the surface tension of the liquid. Suppose the liquid, L is placed on a substrate, S and is surrounded by vapor, V. Let $\gamma SL, \gamma SV$ and γLV be the interfacial surface tension between solid-liquid, solid-vapor and liquid vapor respectively. The line where the three phases meet is called the contact line, and the angle of the liquid phase is called the contact angle θ. The contact angle θ can be computed by calculating the change in free energy due to the virtual displacement of the contact line [2] (Figure 9). In equilibrium, cosine of the angle θ is given by :-

$$cos\theta = (\gamma SV - \gamma SL)/\gamma LV \tag{1}$$

The equation 1 is called as *Young's equation*. When electric voltage is applied the contact angle is changed because the solid-liquid surface tension, γSL varies as a function of the applied voltage.

Physics of Electrowetting:- Lets consider a dielectric layer

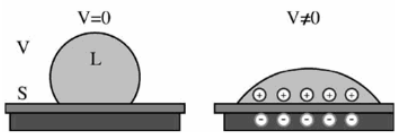

Figure 8: Effect of electric field

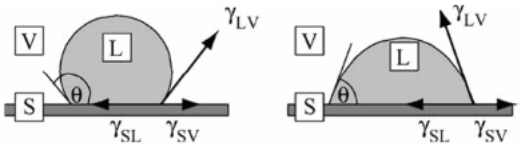

Figure 9: Change in contact angle due to field

of thickness d, below which there is a metal plate. A droplet of conducting liquid (electrolyte) is situated on the upper exposed surface. The droplet is in contact with a wire as shown in Figure 7. By applying an electric voltage U between the electrode and the droplet, charge is accumulated as in a capacitor. Because of the stored electrostatic energy, the interfacial tension between the dielectric layer and the droplet is decreased. As a result of which the droplet contact angle is changed (Figure 9). The cosine of the changed contact angle is given by,

$$cos\theta = (\gamma SV - \gamma SL + 1/2\varepsilon_r\varepsilon_0 U^2/d)/\gamma LV \tag{2}$$

Where ε_r is the relative dielectric constant of the dielectric layer and ε_0 is the dielectric constant of vaccum. In other words, the interfacial tension of the liquid to the substrate can be expressed as a function of applied voltage, V as follows.

$$\gamma SL(V) = \gamma SL(0) - 1/2(\varepsilon_r\varepsilon_0 V^2)/d \tag{3}$$

The above equation suggests that the surface to liquid tension is decreased with the application of voltage. We can conclude that when we apply voltage to an adjacent electrode and the contact interface of the droplet overlaps with this second electrode, the droplet will tend to increase its contact area on that electrode at the cost of the area of the current electrode. As a result, a droplet movement will take place to the next electrode, the electrode adjacent to the current electrode (Figure 10). Snap shot of the movement has been given in Figure 11.

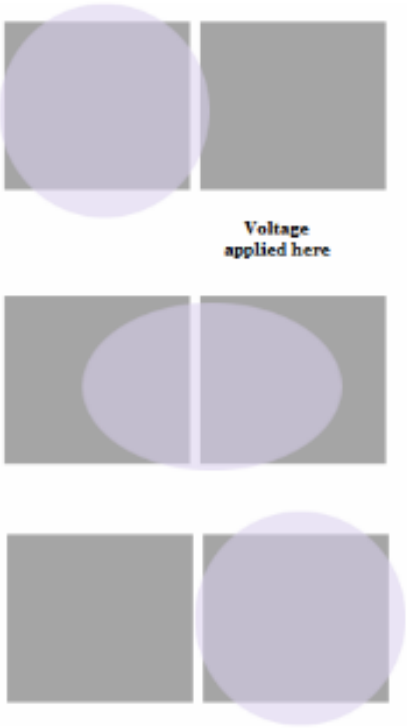

Figure 10: How droplet move

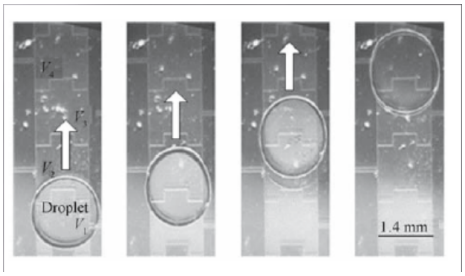

Figure 11: Snap shots of droplet movement

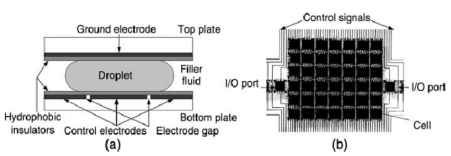

Figure 12: EWOD

3.2 Routing

The goal of droplet routing in a DMFB is to find an efficient schedule, for each droplet, which transports it from its source to target locations, while satisfying all constraints. This sounds similar to VLSI routing where wires need to be connected under design rules, but the reconfigurability of a DMFB makes fundamental differences from VLSI routing in the following aspects:
(1) DMFB routing allows multiple droplets to share the same spot during different time intervals [2, 15, 18] like time division multiplexing, while VLSI routing makes one single wire permanently and exclusively occupy the routing area.
(2) DMFB routing allows a droplet to stall/stand-by at a spot, if needed.
(3) VLSI routing requires $2D$ spacing by design rules, but DMFB routing needs $3D$ spacing by dynamic/static fluidic constraints.
A highly equivalent problem to DMFB droplet routing has been extensively studied in robotics as *mobile robot motion planning*, and solved by *prioritized A∗ search* [15]. In [21, 18, 19, 21], the *mobile robot motion planning* is shown to be *NP-hard*, and an *integer linear programming* approach is proposed. Recent research efforts in DMFB design from VLSI community attack the problem using various heuristics such as internet routing protocol (*Open Shortest Path First*) or pattern selection [21]. However, these approaches suffer from initialization overhead to build either routing tables or to discover a set of feasible routing patterns. Also, as a DMFB keeps reconfigured, this overhead occurs repeatedly, involving large storage overhead. In [8, 11], a novel *network-flow based algorithm* with negotiation is proposed for DMFB droplet routing, showing better performance than [3, 4, 8]. However, the network-flow formulation is significantly bottlenecked by the distribution of blockages. If a width of channel between blockages is less than *3 unit cells*, the channel will not be utilized in the network-flow formulation.

3.2.1 Actuation sequences

Digitized Droplets are to be moved on the $2D$ array with the help of elecrical signals. Let (x, y) and $(x', y') \in (1...n) \times (1...n)$ and $|x - x'| + |y - y'| = 1$ ($i.e, A(x, y)$ and $A(x', y')$ are adjascent). At time t, $A(x, y) = T_i$ and $A(x', y') = \phi$ and at time $t+1$, $A(x, y) = \phi$ and$A(x', y') = T_i$. For this purpose voltage of the cell $A(x, y)$ was ON and that of $A(x', y')$ was in OFF state at the time instant t while at time $t + 1$ voltage of cell $A(x, y)$ is OFF and that of $A(x', y')$ is

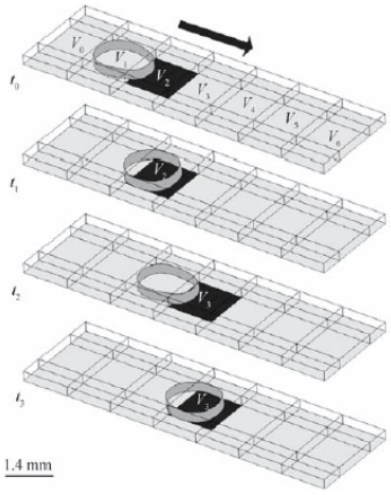

Figure 13: Actuation Sequences

ON. Due to this ON-OFF sequence, droplets placed on the entire chip are made to move and various operation are controlled. This preprogrammed[13] sequence of *ON-OFF* across the grid is called *Actuation Sequence* (Figure 13) .

3.2.2 Fluidic Constraint

During droplet routing, a minimum spacing between droplets must be maintained to prevent accidental mixing, except for the case when droplet merging is desired (e.g., in 3-pin nets[6]). We view the microfluidic modules placed on the array as obstacles in droplet routing. In order to avoid conflicts between droplet routes and assay operations, a segregation region is added to wrap around the functional region of microfluidic modules. In this way, droplet routing can easily be isolated from active microfluidic modules.
For multiple droplet routes that may intersect or overlap with each other, fluidic constraint rules must be introduced to avoid undesirable behavior. Without loss of generality, we refer to two given droplets as D_i and D_j. First, to avoid mixing, we assume that their initial locations at time slot t are not directly adjacent or diagonally adjacent to each other. Let us represent the microfluidic array by two-dimensional coordinates (X, Y), and let $X_i(t)$ and $Y_i(t)$ denote the location of D_i at time t. We must ensure that either $\mid X_i(t)$ - $X_j(t) \mid \geq 2$ or $\mid Y_i(t)$ - $Y_j(t) \mid \geq 2$ for these two droplets. To select the admissible locations of the droplets at the next time slot $t + 1$,

[6]Discussed in section 6

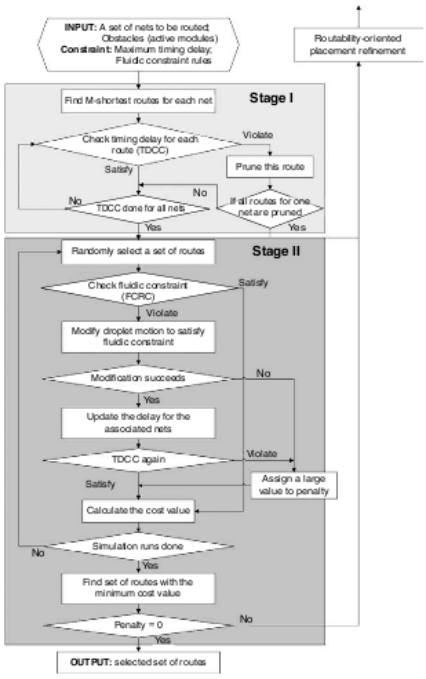

Figure 14: Flow Chart of routing

fluidic constraint rules need to be satisfied as follows.
Rule 1: | $X_i(t+1)$ - $X_j(t+1)$ |≥ 2 or | $Y_i(t+1)$ - $Y_j(t+1)$ |≥ 2, i.e., their new locations are not adjacent to each other (Figure 15).
Rule 2: | $X_i(t+1)$ - $X_j(t)$ |≥ 2 or | $Y_i(t+1)$ - $Y_j(t)$ |≥ 2, i.e., the

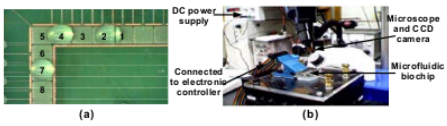

Figure 15: Experimental Set up for Rule verification

activated cell for droplet D_i cannot be adjacent to D_j. Otherwise, there is more than one activated neighboring cell for D_j, which may leads to errant fluidic operation (Figure 16).
Rule 3: | $X_i(t)$ - $X_j(t+1)$ |≥ 2 or | $Y_i(t)$ - $Y_j(t+1)$ |≥ 2.
Note that *Rule*1 can be considered as the static fluidic constraint, whereas *Rule*2 and *Rule*3 are dynamic fluidic constraints.

These fluidic constraint rules have been verified through a set of laboratory experiments. A simple digital microfluidic array was used as the platform for all experiments; electrodes 1 through 8 of this chip are labeled in Figure 15. The experimental setup is also

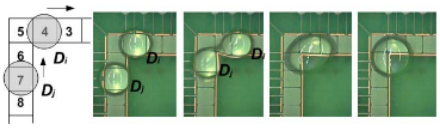

Figure 16: Verification of Rule 2 and Rule 3

shown in Figure 14. The placements of droplets are shown in Figure 15 to illustrate minimum spacing requirement for static droplets. To demonstrate *Rule*1 as shown in Figure 15, we forced droplets D_i and D_j to move by activating electrodes 2 and 3,and deactivating electrodes 1 and 4, simultaneously. Since the new locations of these two droplets, i.e., electrode 2 and 3, were adjacent to each other, they were in contact with each other's surfaces, as shown in Figure 20. To attain minimum surface energy, the two droplets merged into one droplet,which was centered equally over electrodes 2 and 3; see Figure 20. Obviously, the violation of *Rule*1 leads to an unintended mixing of different droplets.We next investigated the consequences of violating dynamic fluidic constraint rules, i.e.,*Rule*2 or *Rule*3. As shown in Figure 16, two droplets D_i and D_j were placed on electrodes 4 and 7 initially. Then, to move D_i and D_j rightward and upward, respectively, electrodes 3 and 6 were actuated simultaneously. Unfortunately, *Rule*3 was violated due to two activated neighboring cells for D_i (i.e., it is directly adjacent to electrode 3 and also diagonally adjacent to electrode 6). We observed that droplet D_j moved rapidly to electrode 6, while droplet D_i showed very little movement towards electrode 3, as Figure 20. Then they contacted each other, thus leading to the mixing of these two droplets, as shown in Figure 20. The purpose of the above experiments was to demonstrate that adherence to *Rule*1 is not sufficient to prevent merging.Both *Rule*2 and *Rule*3 must also be satisfied during droplet routing. Moreover, these fluidic constraint rules are not only used for rule checking, but they can also provide guidelines to modify droplet motion (e.g., force some droplets to remain stationary in a time-slot) to avoid constraint violation if necessary (Figure 19).
Due to these *Rules* sometimes number of clock cycles increases much more as evident from a small example given in Figure 18.

3.2.3 Timing constraints

There is one more constraint in the DMFB that is upper limit on the droplet transport time. This constraint arise from the assumption made in the *Architectural level synthesis of the digital micrifluidic biochips*. It is assumed in [3, 4, 15, 18] that since droplet

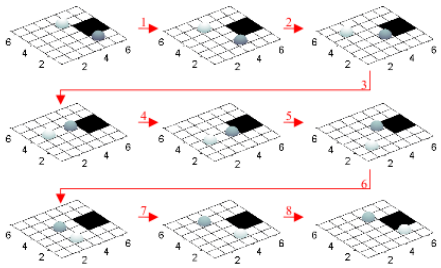

Figure 17: Droplet Routing

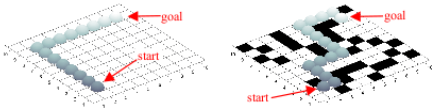

Figure 18: Effect of Forbidden cells in routing

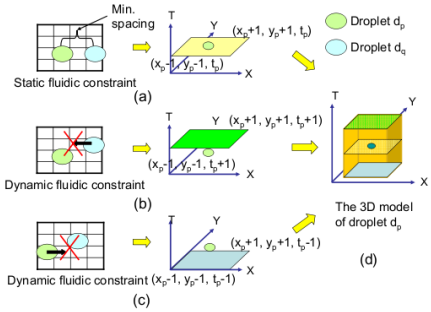

Figure 19: Constraint example

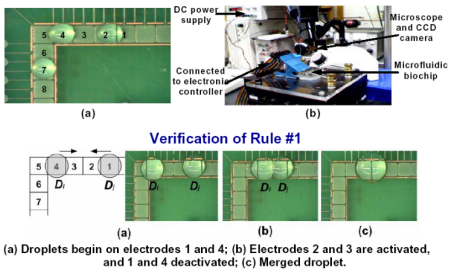

Figure 20: Verification of Rule 1

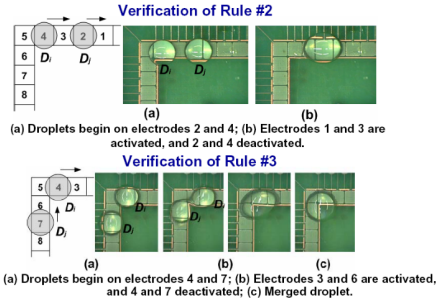

Figure 21: Verification of rules

movement in the routing is very fast compared to the assay operations (e.g.,mixing, dilution, storage, detection etc.), we can ignore the droplet routing time for scheduling assay operations. This assumption has been verified from the laboratory experiment that it takes only $10ms$ to move across one cell with $100Hz$ clock frequency) while it takes 10 seconds for mixing on 2×4 array.
To ensure that the above assumptions is valid for complex set of concurrent assays, we need to ensure that the delay for each droplet route does not exceed some maximum value i.e., 10% of the time-slot used in the scheduling. Otherwise, the schedule obtained from the synthesis procedure is no longer valid. This timing constraint is analogous to the interconnect delay constraints in VLSI routing that require each wire net (circular path) to meet its timing budget. Note that since a droplet may be held at a location in some time slots during its route, the delay for each droplet route is not identical to the route length. The delay for a route therefore consists of the transport time as well as the idle time.

3.3 Various Operation

There are various kind of operation that are to be performed on the DMFB. Some of the *BASIC OPERATIONS* have been given as under:-

3.3.1 Dispensing

Transducer produces the droplets according to some mechanism. For $(x, y) \in (1...n) \times (1...n)$ and some $i \in (1...t)$, a droplet is generated at co-ordinate (x, y) if $A(x, y) = \phi$ at times t and $A(x, y) = T_i$ at time $t + 1$ (where T_i is a type of droplet).

3.3.2 Transportation

For almost all purposes droplets are to be moved from one place to other. Let (x, y) and $(x', y') \in (1...n) \times (1...n)$ and $|x - x'| + |y - y'| = 1$ $(i.e, A(x, y)$ and $A(x', y')$ are adjascent. At time t, $A(x, y) = T_i$ and $A(x', y') = \phi$ and at times $t + 1$, $A(x, y) = \phi$ and $A(x', y') = T_i$.

3.3.3 Merging

When two or more droplets of same or different kinds are merged to form a bigger droplet it is called *merging*. Let (x, y) and $(x', y') \in (1...n) \times (1...n)$ and $|x - x'| + |y - y'| = 1$ $(i.e, A(x, y)$ and $A(x', y')$ are adjascent. At time t, $A(x, y) = T_i$ and $A(x', y') = T_j$ and at times $t + 1$, $A(x, y) = T_k$ is and $A(x', y') = \phi$. Where T_k is the droplet type that results in merging types T_i and T_j.

3.3.4 Splittting

Sometimes it is needed to split a droplet to form a smaller droplet. Let (x, y), (x_1, y_1) and $(x_2, y_2) \in (1...n) \times (1...n)$ and $|x - x_1| + |y - y_1| = 1$ and $|x - x_2| + |y - y_2| = 1$ $(i.e, A(x, y)$ and $A(x_1, y_1)$ are adjascent. At time t, $A(x, y) = T_i$, $A(x_1, y_1) = \phi$ and $A(x_1, y_1) = \phi$ and at times $t + 1$, $A(x, y) = \phi$ is and $A(x_1, y_1) = T_i'$. Where T_i' is the droplet type, generally with half the volume of the original one, that results after *splitting* of the droplet type T_i.

3.3.5 Dilution

Dilution is the process of aquiring a droplet of lower concentration from the higher concentration one. Let (x, y) and $(x', y') \in (1...n) \times (1...n)$ and $|x - x'| + |y - y'| = 1$ $(i.e, A(x, y)$ and $A(x', y')$ are adjascent. At time t, $A(x, y) = T_i$ and $A(x', y') = T_w$ and at times $t+1$, $A(x, y) = T_k$ is and $A(x', y') = \phi$. where T_k is the droplet type that results in dilution of types T_i with water type T_w.

3.3.6 Storage

Sometime it is seen that a droplet or a group of droplet are needed after certain clock cycles but not immedialtely then is unavoidable to store them as a resource for the future operations. Suppose a droplet is made to be placed at $A(x, y)$ at time instant say, $t = t_0$. And for $t = t_0...t_m$ it is at $A(x, y)$ at $t = t_{m+1}$ it moved to $A(x, y)$. This operation is called *storage* operation. And droplet is said to remain stored at $A(x, y)$ for $t = t_0$ to $t = t_m$.

3.3.7 Detection

After assay operations are over the droplet mixed with the proper reagent is made to reach to the site of detection. Where photo-diode and other sensors are placed over it to detect for the result and verify.

3.3.8 Disposing

Disposing is just the reverse operation of *Droplet Formation*. $(x, y) \in (1...n) \times (1...n)$ and some $i \in (1...t)$, a droplet is disposed-off from co-ordinate (x, y) if $A(x, y) = T_i$ at times t and $A(x, y) = \phi$ at time $t + 1$.

3.3.9 Contamination removal

During so many operation, that too, in a very small area it may so happen that biohip is contaminated with chemicals which may result in error for future operation. So a *wash-droplet* is made to route through the entire grid after a regular interval of time so as to remove contamination present, if any, in the system. This is called *Contamination-removal*.

4 Problem Definition

Let us first specify the important physical properties and design parameters of a digital microfluidic system. Then we can move on to a more abstract DMFS model that is independent of specific implementation details.

Layout: Typically, a DMFS consists of a rectangular array A with $n \times n$ cells (but, e.g., an arrangement of hexagonal cells is also possible).

Control circuitry: Various addressing schemes are possible to activate individual cells in a DMFS. We can distinguish, e.g., individually addressable electrodes for each cell, or simpler row/column addressing. For the latter, entire rows and columns are activated, and the droplet is attracted to a neighboring cell $A(x, y)$ only if it lies at the intersection of active column x and row y.

Parallelism: The DMFS controller allows simultaneous activation of more than one cell, and the total number of active cells limited by a number significantly smaller than $n \times n$ generally.

Location of cells with special functions: Droplet generators, reservoirs, cells for merging and splitting of droplets, sensors, waste, etc. may require dedicated cells with special embedded hardware. These specifications provide a physical framework within which a DMFS can operate. Based on this framework, we can establish a formal description of the problem of controlliig droplets in a DMFS. Once a sufficiently general DMFS model exists, we can investigate algorithmic solutions at an abstract level, without worrying about the specific details of varying hardware implementations.

Problem Denition:
A digital microfluidic system is given by an array A with d droplets, their start locations, and their goal locations. Our aim is to, automatically, generate a strategy to move the droplets from start to goal.
More specifically, droplets can be of several types T_i (e.g., T_1 = "*D1 water*",T_2 = "*buffer solution*", etc.), with $i \in (1...t)$, and t is the total number of different droplet types. Each cell $A(x, y)$ in the array can be either occupied by a droplet (denoted as "T_i") empty, ("ϕ") or blocked by an obstacle "X". Thus, at any given time the system can be described by $A(x, y) = c_{xy}$, for $(x, y) \in (1...n) \times (1...n)$ and $c_{xy} \in C = (T_1, ..., T_i, \phi, X)$.
In particular, given a start placement $A_s \in C^{n\times n}$ and a goal placement $A_g \in C^{n\times n}$, we need to find a sequence of valid transitions that results in the desired droplet motion from A_s to A_g.

5 Nature of the problem

The complexity of the problem can be realized by the fact fact that even population of few droplets can lead to a vast no. of possibility. Suppose the number of droplets is d. In the simplest case, all are of the same type T_0. Then the number of different placements of droplets on the array, for modest numbers $m = n = 10$ and $d = 10$ yields more than $\sim 1.7 \times 10^{13}$ possibilities. If all droplets are of distinct type T_1, T_2, T_3,...,T_d then this number increases by $d!$ (i.e,$\sim 3 \times 10^{19}$). One might hope that in practice, most of these choices need not be explored. However, at each step, d droplets offer up to $4d$ choices to be moved, assuming 4 neighbor cells per droplet. Thus, finding a strategy with s steps could mean checking up to

$(4^d)^s$ choices or risk missing the solution, resulting again in astronomical numbers even for $s < 10$. The research goal of this report is to generate optimal sequences of control signals to move droplets from start to goal positions in the shortest number of steps. With growing array size and number of droplets, this becomes increasingly challenging: closely related optimizations are the *travelling salesman problem* and *motion planning with multiple moving robots* [21], which are known to be computationally complex (i.e., *NP-hard* [1, 2]).

6 Previous Work

There are a different sort of approaches taken from time to time. Wire routing is a well-studied problem in VLSI design. Due to the analogy between digital microfluidics and digital electronics, many classical VLSI routing techniques can be leveraged for the droplet routing problem [22]. However, there exist some important differences. For example, whereas electrical nets must not be short-circuited in VLSI routing (i.e., they cannot intersect each other), different droplet routes can be overlapped on some locations as long as they satisfy fluidic constraints. This is due to the property of virtual nets in digital microfluidic biochips (i.e., the droplet route is dynamically formed by sequentially activating the corresponding control electrodes). Consequentially, capacity constraints that result from fixed routing regions in VLSI design are not as important in droplet routing.
Instead, a different set of constraints (e.g., fluidic constraints) need to be taken into account here. In addition, the traditional approach to VLSI routing is a two-step approach: global routing is performed first, and it is followed by detailed routing [27, 28, 29].
While this two-step technique is suitable for VLSI chips that contain millions of transistors, it suffers from high computation cost. For a digital microfluidic array with a much smaller level of integration (typically tens or hundreds of modules), a simpler one-step routing technique is more suitable.
Recently reconfigurable devices such as FPGAs and dynamic networks-on-chips (NoCs) have received much attention [22, 23, 24]. In order to overcome pin limitations of FPGA-based logic, some time-multiplexed routing methods such as Virtual Wires [25], have been proposed. Some packet routing algorithms are also proposed for NoCs [24]. By intelligently multiplexing each physical wire among multiple logical wires, such programmable routing is in some ways

similar to the droplet routing problem. However, because of the differences in physical structure between FPGAs (NoCs) and digital microfluidic biochips, these time-multiplexed routing methods cannot be directly used for droplet routing. For example, unlike FPGAs that have well-defined roles of interconnect and logic blocks, there are no physical interconnects in digital microfluidic biochips. The same cells in the microfluidic array can be used for transporting droplets, as well as microfluidic operations such as mixing. Such features make the droplet routing problem different from FPGA routing.

A channel routing problem for continuous-flow microfluidic biochips has been investigated in [24]. The goal of this work is to determine the precise paths of fabricated microchannels that connect a microfluidic component port to a fluidic I/O well on the chips edge. Since these biochips are fabricated in a single layer of glass, silicon, or plastic, all microchannels must be routed in a planar fashion. Moreover, unlike virtual nets in digital microfluidic biochips, these microfabricated channels must not intersect. Thus, this routing problem is similar to the classical single-layer VLSI routing problem.

A problem related to droplet transport planning has been analyzed in [34]. If we interpret each droplet as a mobile robot, the droplet path-planning problem for digital microfluidic biochips can be viewed as a *motion-planning problem with multiple moving robots*. First a simple algorithm based on A^* search was proposed in [34], but its computational complexity is exponential in the number of cells in the array and the number of droplets. To reduce computational complexity, a prioritized A^* search technique was then presented, which assigns priorities to each droplet and generates paths successively, starting with the highest-priority droplet. Low-priority droplets consider higher-priority droplets as time-varying obstacles. However, priority assignment (i.e., the order of droplet routing) becomes a crucial issue. It was suggested in [34] that this order can be assigned at random, or it can be based on application-specific guidelines. However, it is impossible to devise a general procedure that is suitable in all situations. In some extreme cases, low-priority droplet transport can take a long time; thus practical timing or throughput constraints are not considered in [34]. To address the above limitations, we propose a new path planning method that does not rely on a routing order. Another drawback in [34] is that only routes between two terminals are considered. However, many droplet routes connect multiple terminals when practical bioassays or sets of bioassays are applied to the digital microfluidic platform.

Thus, droplet pathways must be modeled as multipin nets.
A second approach for coordinating multiple droplets in digital microfluidic bio-chips has been presented in [27]. This method first applies a semiautomated approach to generate the array layout in terms of defined components (e.g., streets or connectors). By viewing the microfluidic array as a network, the droplet path-planning problem is reduced to a network flow problem [33]. Since droplet motion is only limited to the fixed streets, this approach does not exploit some of the important benefits of digital microfluidics such as dynamic reconfigurability and virtual devices.
This section presents the proposed droplet routing method. The quality of the solution obtained by this method is independent of the routing order of nets. The inputs to the algorithm are a list of nets to be routed in each subproblem as well as constraints imposed by the designer.
The droplet routing algorithm consists of two basic stages. The first stage generates M alternative routes for each net, where M can be fine-tuned through experiments . The algorithm attempts to find the shortest routes for each net (*2-pin net or 3-pin net*). In addition, the obtained routes also need to pass the timing delay constraint check (TDCC) in this stage. Those that violate the timing constraint are pruned from the set of alternative routes. Note that, if all the shortest routes for some net do not satisfy the timing constraint, placement refinement is required to increase the corresponding routability; this refinement can be carried out manually or by a routability-oriented placement algorithm (e.g., the low-temperature simulated annealing LTSA method).
The second stage of the routing algorithm first randomly selects a single route from the alternatives for each net. The set of randomly selected routes for a given net list then go through the fluidic constraint rule check (FCRC). If necessary, some modifications are made to the routes to ensure that all fluidic constraints are satisfied. The delay for the corresponding net is updated, and then TDCC is performed again. The objective function for this set of routes is also obtained by calculating the number of cells used in routing. Through an appropriate number of random selection runs, a set of routes with the minimum cost function, subject to both timing and fluidic constraints, is finally selected to be the output of the routing algorithm. If no suitable solution for droplet routing is found, routability-oriented placement refinement is invoked again.
Advantages of the routing methodology described above include the avoidance of the net-routing-order dependence problem and the use of dynamic reconfigurability. Some key details of the algorithm are

as follows.

PHASE I: M-SHORTEST ROUTES

In this phase, M alternative routes for each net are generated. Lee algorithm is modified, a popular technique used in grid routing [22, 23], for the droplet routing problem in digital microfluidic biochips.

Two-Pin Nets

As discussed above, any route on which a single droplet is transported from one microfluidic module (or on-chip reservoir) to another one can be modeled as a *2-pin net*. Obviously, the shortest route problem for *2-pin nets* is equivalent to the single-pair shortest path problem. An advantage of the *Lee algorithm* is that it is guaranteed to find the shortest path between two pins, which can be included among the M alternatives.

The basic Lee algorithm includes three phases: filling phase, retrace phase, and label clearance. The underlying idea is to reduce the number of cells searched in the algorithm, which is proportional to the running time.

1. Double fan-out: During the filling phase, labeling waves are propagated from both the source and destination cells simultaneously. Labeling continues until contact points between two wavefronts are found. This technique approximately halves the number of cell searched.

2. Boundary window: It has been proved that the length $L(P)$ of a path P is given by $L(P) = MD(S, D) + 2 \times d(P)$, where $MD(S, D)$ is the Manhattan distance between the source cell S and the destination cell D, and the detour number $d(P)$ is defined as the number of cells directed away from the destination cell D in a path P. We should ensure that the timing constraint is satisfied in droplet routing (i.e., $L(P) \leq T_d$, where $L(P)$ denotes the number of cells that constitute P, and T_d is the maximum time slot). Here we assume that the droplet moves across one cell in one time slot. Thus, the detour number of the path should also satisfy $d(P) \leq (T_d MD(S, T))/2 = d_{max}$. Therefore, an artificial boundary can be placed around the net pairs to be connected. The filling phase is only limited to cells bounded in this window; thus it significantly speeds up the algorithm. Lee algorithm is further modified to find M shortest paths for each net. Note that there may be more than one contact point between two labeling wavefronts. Thus, each of them can be chosen as the starting point of the retrace procedure. In addition, there generally exists a choice of cells during the retrace phase. Therefore, M different routes can be obtained by selecting

different contact points as well as different retracing choices. Note that these M alternative routes may include the next shortest paths, the lengths of which are longer than the shortest one. After the M shortest paths for each net are determined, TDCC is performed. Therefore, paths are pruned for which the $L(P)$ exceeds T_d. If M alternative routes for some net are all eliminated due to the violation of the timing constraint, placement refinement is required to ensure biochip routability.

Three-Pin Nets

3-pin nets is used to model the routes along which two droplets are transported toward a microfluidic module (e.g., a mixer); the droplets can mix together during their transportation. The shortest-route problem for such nets is equivalent to the *Steiner Minimum Tree (SMT)* problem . Since *SMT* is known to be *NP-hard*, a heuristic approach is needed to efficiently solve this problem.

First, as in the case of two-pin nets, the *Lee algorithm* with double fan- out and boundary window is used to obtain the shortest path connecting two source cells. Here the contact point between two labeling wavefronts is taken as the mixing location, where two droplets are intended to merge with each other. Then this *mixing point is treated as source*. Again, the shortest path between the mixing point and the destination cell is obtained using a similar procedure. This two-phase process can find a feasible route connecting three pins. Note that the interconnection obtained by this heuristic method is not guaranteed to be of minimum length. However, it is a desirable route in practice, allowing concurrent mixing during transportation. The delay for this *3-pin net* is set to $L(P) = L(P1)/2+L(P2)$, where $P1$ is the path between two source cells and $P2$ is the path connecting the mixing point and the destination cell. As in the case of *2-pin nets*, M alternative routes for each *3-pin net* can be obtained by selecting different contact points and retrace choices. TDCC is then carried out to verify that the delay constraint is met for these routes.

PHASE II: RANDOM SELECTION

In the second phase of the algorithm, a single route from the M_i alternatives for each net i is selected, where $i \in 1, 2, ...N$ and N is the number of nets. Note that $M_i \leq M$, since some routes that violate the timing constraint have already been eliminated. A random-selection approach is then used to select i_k for each net i, where i_k represents the k-th alternative route for net i, and $k \in 1, 2, ..., M_i$. A desirable feature of this random method is that it avoids the net-routing-order dependence problem. The algorithm seeks to minimize the total number of microfluidic cells used in routing, while

adhering to both the timing constraint and fluidic constraints.

FCRC AND DROPLET MOTION MODIFICATION

In the first stage of the algorithm, M alternative routes for each net are obtained irrespective of the existence of other nets. The only obstacles that is considered are microfluidic modules that are active during this time interval. However, since different net routes may share the same cells, or they may be close to each other, it is needed to ensure that they do not violate the fluidic constraints. Otherwise, different droplets may accidentally merge, thus leading to incorrect assay operations and sample contamination. Assume that two droplet routes (i.e., P_i and P_j) have been obtained using the modified Lee algorithm. To adhere to fluidic constraint rules,it is needed to check these two droplets D_i and D_j in each time slot. Interestingly, even if a rule violation is found, it can still modify droplet motion (i.e., force a droplet to stay in the current cell instead of moving) to override the violation. If some droplet (e.g., Pi) is required to stay in its current location, $X_i(t+1) = X_i(t)$ and $Y_i(t+1) = Y_i(t)$. Also, the delay for this path is updated with $DL(Pi) = DL(Pi) + 1$. If the modification fails, the corresponding routing paths are deemed to be infeasible. It is further considered the case of more than two droplet pathways. It need to ensure that every pair of droplets satisfies all the fluidic constraint rules. At each time slot, droplet routes are first sorted in the descending order of their delay values. The first route $P1$ is selected to sequentially perform FCRC withP_i ($i \geq 2$). The modification based on the rules [33] is applied if necessary. After that, thesecond route $P2$ is selected to perform FCRC as well as modification with P_i ($i \geq 3$). FCRC is continued until all pairs of routes are checked. If any route pair fails both FCRC and modification, these routes are not feasible solutions. Thus, the features of dynamic reconfigurability and independent controllability is exploited to not only check for violation of fluidic constraints, but also to temporarily suspend the droplet motion to override rule violation.

7 Our Approach

7.1 Algorithm

We have employed a scheme of *SHORTEST MANHATTAN DISTANCE FIRST ALGORITHM* in which first of all we sort the droplets according to their *MANHATTAN DISTANCE* and form a

set S_0. Then we make a subset S of the *MOVABLE DROPLETS*[7] out of the set S_0. And in this subset the priority is given to the droplet with *minimum MANHATTAN DISTANCE* and try to route the entire subset. After this we modify the subset and do the same thing in the next clock cycle. If *deadlock* occurs then we *backtrack* until all the droplets are made to reach their respective distance. The detailed algorithm has been given as under:-

SHORTEST MANHATTAN DISTANCE FIRST ALGORITHM:-

INPUT:- An $N \times N$ Grid labelled as (x_0, y_0), (x_1, y_1), (x_2, y_2),..., (x_{N^2-1}, y_{N^2-1}) and m droplets labelled as $d_0, d_1, d_2, \ldots, d_{m-1}$ placed at $(x_{i_0}, y_{i_0}), (x_{i_1}, y_{i_1}), (x_{i_2}, y_{i_2}), \ldots, (x_{i_{m-1}}, y_{i_{m-1}})$ respectively.

OUTPUT:-An $N \times N$ Grid labelled as (x_0, y_0), (x_1, y_1), (x_2, y_2),..., (x_{N^2-1}, y_{N^2-1}) and m droplets labelled as d_0, d_1, d_2,...,d_{m-1} placed at (x_{f_0}, y_{f_0}), (x_{f_1}, y_{f_1}), (x_{f_2}, y_{f_2}),..., $(x_{f_{m-1}}, y_{f_{m-1}})$. respectively.

STEP 1 :- First of all we'll find the "Manhattan Distance" of the droplets (let us say $d_0, d_1, d_2, ..., d_{n-1}$)
STEP 2:- Sort them according to the increasing order of their Manhattan distance. Say $d'_{s_0}, d'_{s_1}, d'_{s_2}, ..., d'_{s_{m-1}}$.
STEP 3:-Route the d'_{s_i} ,which is shortest among the leftover droplets, first and along with that take all of those in the subset, $S \subseteq S_0$, of droplets whose movement is possible and that movement result in the decrease of their respective Manhattan-Distance. Give the clean passage to this droplet so as to make it reach to the destination i.e, another droplet, say, d'_{s_j} which has already reached to its respective destination then make it move one or two place so as to make the path of the droplet in hand to be routable.
STEP 4:-Modify S and *ROUTE* all the droplets along with d'_{s_i} in randomly selected sequence.
STEP 5:-If droplet d'_{s_i} reaches to the desired destination then go to STEP 6, else go to STEP 4, else if *DEADLOCK* occured then *BACKTRACK* and go to STEP 4.
STEP 6:-$S_0 \leftarrow S_0$ - $\{d'_{s_i}\}$
STEP 7:-If $S_0 = \phi$ then ROUTING SUCCESSFUL else if $S_0 \neq \phi$ go to STEP 4

[7]Those droplets whose movement in the next cycle doesn't violet the Routing Rules

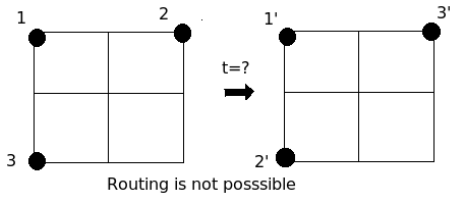

Figure 22: Routing not possible

BACKTRACKING:-

INPUT:-An $N \times N$ Grid labelled as (x_0, y_0), (x_1, y_1), (x_2, y_2),..., (x_{N^2-1}, y_{N^2-1}) and m droplets labelled as $d_0, d_1, d_2, \ldots, d_{m-1}$ placed at $(x_0(t), y_0(t))$, $(x_1(t), y_1(t))$, $(x_2(t), y_2(t))$,..., $(x_{m-1}(t), y_{m-1}t)$ respectively, at time instant t.
OUTPUT:-An $N \times N$ Grid labelled as (x_0, y_0), (x_1, y_1), (x_2, y_2),..., (x_{N^2-1}, y_{N^2-1}) and m droplets labelled as $d_0, d_1, d_2, \ldots, d_{m-1}$ placed at positions as $(x_0(t+1), y_0(t+1))$, $(x_1(t+1), y_1(t+1))$, $(x_2(t+1), y_2(t+1))$, ..., $(x_{m-1}(t+1), y_{m-1}(t+1))$ respectively at time instant $t+1$ where all $x_i(t+1)s$ and $y_i(t+1)s$ are positions of $d_i s$ at time instant $t-1$.
STEP 1:-for $i \leftarrow 0$ to $m-1$
$x_i(t+1) \leftarrow x_i(t+1)$
$y_i(t+1) \leftarrow y_i(t-1)$

ROUTE:-

INPUT:-A droplet d_i with position $(x_i(t), y_i(t))$ on given $N \times N$ Grid.
OUTPUT:-The changed position of the droplet d_i as $(x_i(t+1), y_i(t+1))$ on given $N \times N$ Grid.
STEP 1:-if(satitic rule and dynamic rule of routing both are satisfied) then
$(x_i(t+1) \leftarrow (x_i(t) \pm 1$ or $(y_i(t+1) \leftarrow (y_i(t) \pm 1$ or both
else
$(x_i(t+1) \leftarrow (x_i(t)$ and $(y_i(t+1) \leftarrow (y_i(t)$
end if

7.2 Examples

From Figure 28 and Figure 29 we see that when we follow our approach we are able to route the droplets.

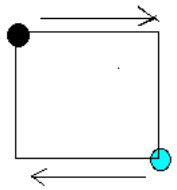

Figure 23: Unstable(Statically and Dynamically both)

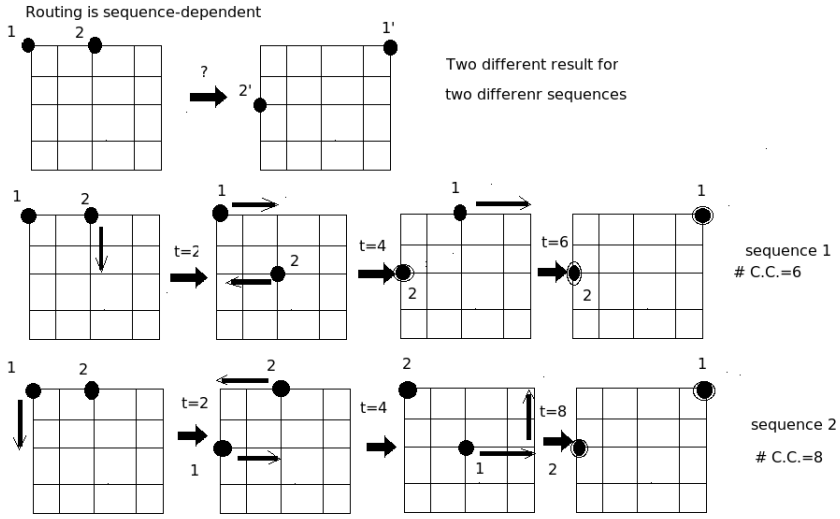

Figure 24: Sequence-dependency

8 Experimental Observations

There are some interesting observations during the execution of the routing method and they are as under:-

1. The time taken for the routing is ***sequence-sensitive*** i.e, if we follow two different sequence of movements of droplets movements we may get two different "***time of completion***" as evident from example given through Figure 24 .

2. There are some intial configuration which are stable but no matter in what manner or in what sequence you route the droplets it is *not possible* to reach to the final configuration. As evident from the example in Figure 22.

3. Sometime ***stall*** is helpful for avoiding *deadlock* but not always.

4. The no. of clock cycles as well as feasibility of *successful* routing depends a lot upon ***density*** of the droplets, ***position*** of the

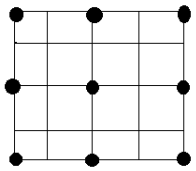

Figure 25: No more droplets can be accommodated(odd)

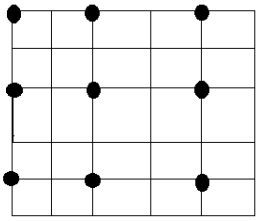

Figure 26: No more droplets can be accommodated(even)

droplets etc. (Figure [25-32]).

5. For a Grid of size $N \times N$ deadlock is bound to happen if ***no.of droplets*** is more than or equal to $(N+1)^2/4$ when N is *odd* (Figure 25).

6. For a Grid of size $N \times N$ routing is difficult except for trivial cases if ***no.of droplets*** is more than $N^2/4$ when N is *even* (Figure 26).

7. Sometimes it may so happen that for completion of the routing we may have to disturb the position of the droplet which has already reached to its destination. But this process doesn't always increase the clock cycle.

8. If there is already 2 droplets on a subgrid of size 3×3 of a Grid then a $3rd$ droplet can be routed across it only if it is in *position 2* in Figure 28 but if it is in *position 1* of Figure 28 then it is not possible to route it across the subgrid even though the $3rd$ one is inserted into it.

9. If two droplet are to be routed ***anti parallel*** to each other the D-Tour cann't be avoided if they are on adjascent lines or on same line and are going to cross the same vertical line during their movement Figure 30.

10. If the droplets are ***sparse*** then routing is easy as well as feasible in most of the cases. Generally, density more than $\approx 25\%$ lead to deadlock (Figure [24,25]).

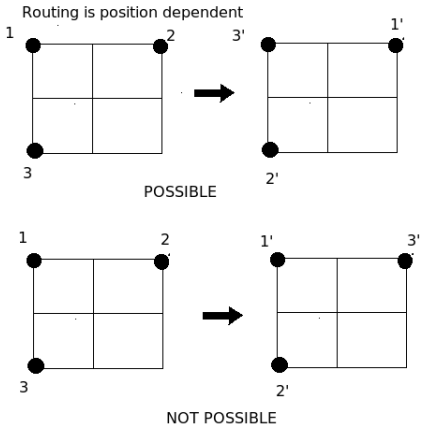

Figure 27: Dependency on position

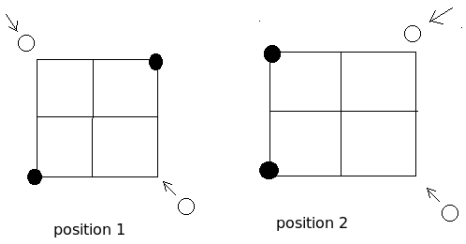

Figure 28: Position matters

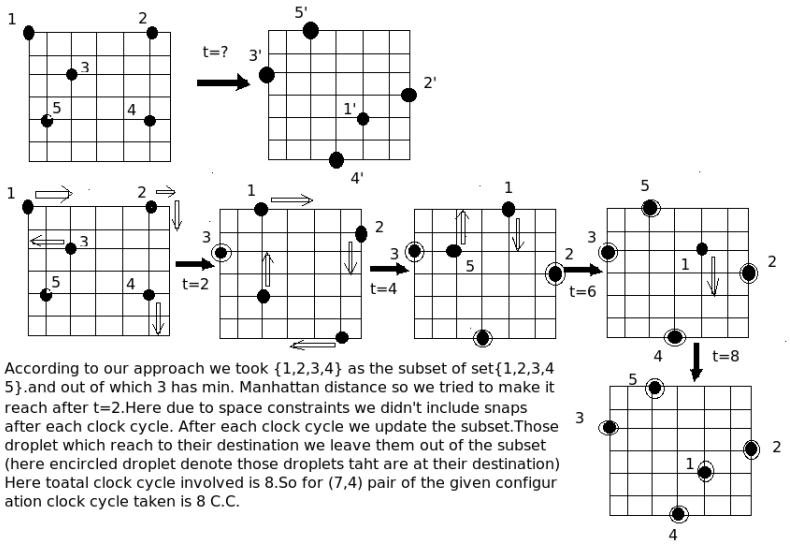

Figure 29: Steps to show how our method works

11. To break the deadlock one more alternative is there but it is not always recommended. And that alternative is to merge two droplets if they are of same type so as to have some routing space and after few clock cycle that droplet is made to split up whenever feasible to do so, etc.

12. All the above observations are for cases without any **forbidden cells**. In case of forbidden cells problem become more complicate than normal one.

13. A 5×5 grid can be considered to be made up of 4 3×3 cells. Similarily a 7×7 grid can be considered as 9 3×3 grid and so on. So we can say 3×3 grid to be ***basic*** one.

14. If we see that a given (m, n) pair if the grid is composed of *routable basic grid*[8] then we can say that routing is possible.

15. In most of the cases it is seen that routing is feasible if $m \leq 2 \times N - 3$ and $m \leq 2 \times N - 1$ if N is *even* and *odd* respectively.

[8]The 3×3 grid with less than or equal to 2 droplets

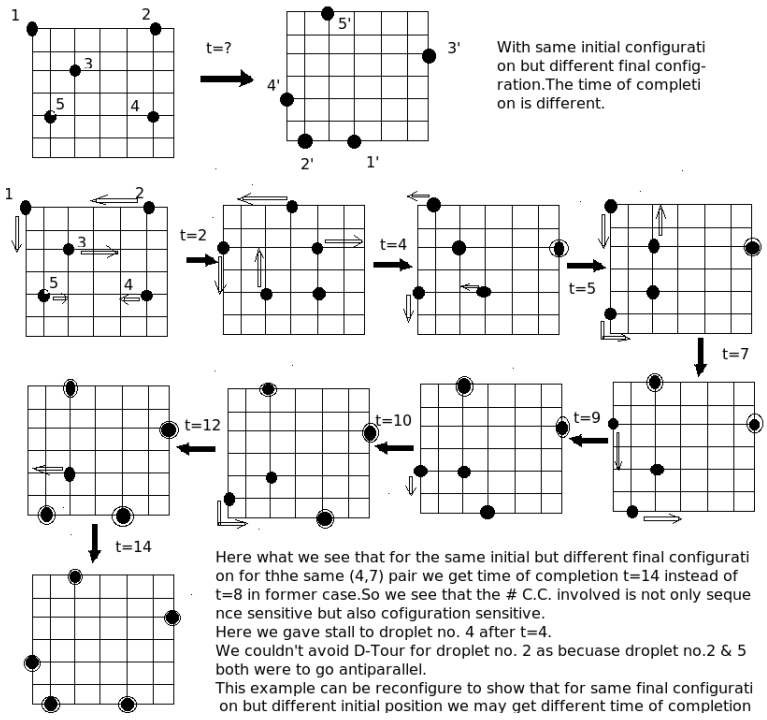

Figure 30: Configuration Dependency

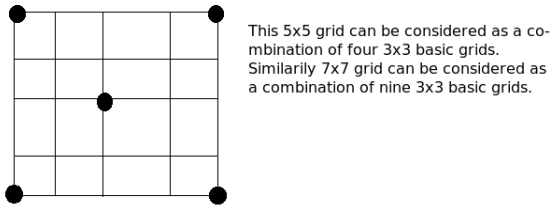

Figure 31: Easily Routable

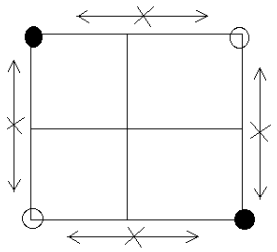

Figure 32: Density Decrease the Routability

9 Experimental Results

9.1 Tables

The following tables illustrate the experimental results:-

No. of Droplets	Time					T_{avg}	Prob where deadlock occured	Backtracking Performed?	Routing successful?
	P1	P2	P3	P4	P5				
1	1	1	2	1	2	1.4	*NONE*	*N.A.*	*ALL*
2	*N.P.*	*N.P.*	*N.P.*	*N.P.*	*N.P.*	*N.P.*	*ALL*	*N.A.*	*N.P.*

Table 1: 2X2 Grid

No. of Droplets	Time					T_{avg}	Prob where deadlock occured	Backtracking Performed?	Routing successful?
	P1	P2	P3	P4	P5				
1	1	3	2	3	4	2.6	*NONE*	*N.A.*	*ALL*
2	3	4	3	4	3	3.2	*NONE*	*N.A.*	*ALL*
3	4	8	13	5	*N.P.*	*N.P.*	5	5	$1, 2, 3, 4$
4	*N.P.*	*N.P.*	*N.P.*	*N.P.*	*N.P.*	*N.P.*	*ALL*	*N.A.*	*N.P.*

Table 2: 3X3 Grid

No. of Droplets	Time					T_{avg}	Prob where deadlock occured	Backtracking Performed?	Routing successful?
	P1	P2	P3	P4	P5				
1	1	3	6	7	2	3.8	*NONE*	*N.A.*	*YES*
2	5	2	4	8	6	5	*NONE*	*N.A.*	*YES*
3	7	6	8	9	10	8	*NONE*	*N.A.*	*YES*
4	7	10	11	7	28	12.6	5	*YES*	*ALL*

Table 3: 4X4 Grid

No. of Droplets	Time					T_{avg}	Prob where deadlock occured	Backtracking Performed?	Routing successful?
	P1	P2	P3	P4	P5				
1	2	3	5	4	8	4.4	*NONE*	*N.A.*	*YES*
2	6	5	7	8	7	6.6	*NONE*	*N.A.*	*YES*
3	7	9	11	8	9	8.8	*NONE*	*N.A.*	*YES*
4	11	9	12	7	8	9.4	*NONE*	*N.A.*	*YES*
5	12	13	21	*N.P.*	*N.P.*	*N.P.*	4,5	4,5	1,2,3,4

Table 4: 5X5Grid

No. of Droplets	Time					T_{avg}	Prob where deadlock occured	Backtracking Performed?	Routing successful?
	P1	P2	P3	P4	P5				
2	5	7	6	9	10	7.4	*NONE*	*N.A.*	*YES*
3	10	8	12	15	10	10	*NONE*	*N.A.*	*YES*
5	14	16	17	12	11	14	*NONE*	*N.A.*	*YES*
7	15	17	23	24	38	21.2	5	5	*YES*
9	*N.P.*	*N.P.*	*N.P.*	*N.P.*	*N.P.*	*N.P.*	ALL	N.A.	*N.P.*

Table 5: 6X6 Grid

No. of Droplets	Time					T_{avg}	Prob where deadlock occured	Backtracking Performed?	Routing successful?
	P1	P2	P3	P4	P5				
3	9	12	13	11	15	12	*NONE*	*N.A.*	*YES*
4	14	18	12	11	18	14.6	*NONE*	*N.A.*	*YES*
5	16	18	25	23	18	20	*NONE*	*N.A.*	*YES*
7	17	21	16	24	35	22.6	5	5	*YES*
9	18	23	27	34	38	28	4,5	4,5	*YES*
12	25	31	35	45	51	37.2	*ALL*	*ALL*	*YES*
16	*N.P.*	*N.P.*	*N.P.*	*N.P.*	*N.P.*	*N.P.*	*ALL*	*N.A.*	*N.P.*

Table 6: 7X7 Grid

No. of Droplets	Time					T_{avg}	Prob where deadlock occured	Backtracking Performed?	Routing successful?
	P1	P2	P3	P4	P5				
3	16	14	16	18	11	15	*NONE*	*N.A.*	*YES*
5	17	15	14	15	16	15.2	*NONE*	*N.A.*	*YES*
7	19	16	18	17	21	18.2	*NONE*	*N.A.*	*YES*
9	21	22	29	31	35	27.6	4,5	4,5	*YES*
12	41	32	24	17	38	30.4	1,2,5	1,2,5	*YES*
14	31	39	41	25	38	34.6	1,2,3,5	1,2,3,5	*YES*
16	*N.P.*	*N.P.*	*N.P.*	*N.P.*	*N.P.*	*N.P.*	*ALL*	*N.A.*	*N.P.*

Table 7: 8X8 Grid

No. of Droplets	Time					T_{avg}	Prob where deadlock occured	Backtracking Performed?	Routing successful?
	P1	P2	P3	P4	P5				
4	12	10	9	15	16	12.4	*NONE*	*N.A.*	*YES*
7	18	13	16	17	15	15.8	*NONE*	*N.A.*	*YES*
11	17	21	31	19	32	24	3,5	3,5	*YES*
13	18	34	38	21	24	27	3	3	*YES*
17	21	35	41	36	19	30.4	2,3,4	2,3,4	*YES*
21	23	27	41	36	29	32.4	3,4	3,4	*YES*
25	*N.P.*	*N.P.*	*N.P.*	*N.P.*	*N.P.*	*N.P.*	*ALL*	*N.A.*	*N.P.*

Table 8: 9X9 Grid

No. of Droplets	Time					T_{avg}	Prob where deadlock occured	Backtracking Performed?	Routing successful?
	P1	P2	P3	P4	P5				
4	16	10	18	15	12	14.2	*NONE*	*N.A.*	*YES*
7	18	23	14	15	17	17.4	*NONE*	*N.A.*	*YES*
11	17	31	26	24	32	26	2	2	*YES*
13	41	35	21	32	19	29.8	1,2,4	1,2,4	*YES*
17	41	30	23	36	29	33	1,4	1,4	*YES*
21	32	45	25	31	41	34.8	1,2,4,5	1,2,4,5	*YES*
25	*N.P.*	*N.P.*	*N.P.*	*N.P.*	*N.P.*	*N.P.*	*ALL*	*N.A.*	*N.P.*

Table 9: 10X10 Grid

9.2 Discussion on Results

We see from the tables that for the same number of droplet and on same grid size time of execution is different for different configuration. *P1, P2, P3, P4*, and *P5* are different instances of configuration for the same (m, n)[9] pair. This is in accordance with the fact that routing is configuration dependent. We see that when no. of

[9] m droplets placed on a $n \times n$ grid

droplets are 2,5,9 etc. for grid of size 2×2, 3×3, 5×5 respectively then routing is not possible and the fact that when no. of droplets is more than $(N+1)^2/4$ then for a Grid of size $N \times N$ deadlock is bound to happen. One more thing is to be noted here is that for same (m, n) pair and same Pi if we follow different sequence then there is quite a good chance that time of comletion would be different because of the fact that routing is *sequence sensitive*. And as the density increases chances of deadlock increases and so is the need of backtracking.
N.B.:- Here *N.P.* and *N.A.* denotes *NOT POSSIBLE* and *NOT APPLICABLE* respectively in the tables.

10 Future Challenges

As we know the problem become manifold complicated when the number of droplets increases. We have employed the *SHORTEST MANHATTAN DISTANCE FIRST* which is a greedy approaches.It has been seen experimentally that it gives good result but it doesn't always guarantee *optimality*. As we have seen that there are so many issues in this problem i.e, time cycle taken to complete the routing is sequence sensitive, population sensitive etc. Sometimes it is not possible to route droplets at all. One of the main problems is that we cann't predict priori whether or not a sequence we are choosing leads to deadlock or not. So in this case if deadlock occurs then we backtrack and select a random sequence which again cann't guarantee that deadlock may not occur. So backtrack gives a hope for solution but at the cost of clock cycles.We have not considered the case of forbidden cells which increases the complexity further. So this can also be considered and will need modification in present approach.
In some cases the time taken to complete the routing comes out to be more than that of the normal one sometimes due to selection of sequence and sometimes due to density. After it is seen that to accomplish the complete routing we have taken individual droplets and make it to reach to destination leaving aside the concept of parallelism for the time being.Fault tolerence should also be taken care of.Some popular techniques i.e., Genetic Algorithm ,Simulated Annealing etc. can also be applied to make solution more favorable and reliable.
So, there are lots of challenges left for future and there are a lot of scope of improvement in the approach as well. Optimality is one of the main challenges in this regime.

11 Conclusion

Our approach is to find out the solution, if it exists. It is greedy one. Since in our approach the *SHORTEST MANHATTAN DISTANCE FIRST* strategy is employed it gives solution very near to optimal in most of the cases but not in all the cases. As evident from the results the time taken for same (m, n) pair is different for the different choices of movement (Figure [24,29,30]). And some configuration are there for which no matter how we proceed we may not reach to desired configuration (Figure 22). In that case only partial solution can be achieved. There are lots of improvements and modifiacations possible to be done in this approach. From the experiments and example we can say that it is not only the value of m and n in (m, n) pair that determines the sequence of steps but also the distribution and density of droplets are also one of the most important parameter to decide about solution steps involved.

References

[1] **K. F. Bohringer**, *Towards optimal strategies for Moving Droplets in DMF IEEE Proceedings International Conference on Robotics and Automation*

[2] **E. J.Griffith and S. Akella** in *Coordinating Multiple Droplets in Planar Array DMFS The International Journal Of Robotics Reasearch November 2005*

[3] **F. Su ,W. Hwang, K. Chakraborty** in *Droplet Routing in the Synthesis of DMFS.*

[4] **M. Cho and D. Z. Pan** in *A High Performance Droplet Router For DMFS*

[5] **T. Xu and K. Chakraborty** : *Functional Testing of DMFB ,2007 IEEE.*

[6] **D. Mitra, S. Ghoshal, H. Rahamann, B. B. Bhattacharya, D. D. Majumdar,K. Chakraborty** : *Accelarated Functional Testing Of DMFB, ATS 2008*

[7] **F. Su, W. Hwang ,A. Mukherjee , K. Chakraborty** :*Tesing and Diagnosis of Realistic Defects In DMFB.*

[8] **J. Ding, K. Chakraborty** in *Scheduling of Microfluidics Operation for Reconfigurable Two-Dimensional Electrowetting Arrays IEEE Transaction on CAD Intgrated Circuits And Systems.*

[9] **T. Zhang, K. Chakraborty** in *Design of Reconfigurable Composite Microsystem On Hardware/Software Codesign Principles IEEE Transaction on CAD Intgrated Circuits And Systems .*

[10] **J. Wu, R. Yue, X. Zeng, L. Liu** in *Droplets Actuating Chip Based On EWOD Springer-Verlog-2007,* etc.

[11] **E. J. Griffith, S. Akella, and M. K. Goldberg**. *Performance characterization of a reconfigurable planar array digital microfluidic system. IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, 25(2):340352, Feb. 2006.*

[12] **K. Ichimura, S. Oh, and M. Nakagawa**. *Light-driven motion of liquids on a photoresponsive surface. Science, 288:16241626,Jun 2000.*

[13] **T. B. Jones, M. Gunji, M. Washizu, and M. J. Feldman**.*Dielectrophoretic liquid actuation and nanodroplet formation.Journal of Applied Physics, 89:14411448, Jan 2001.*

[14] **T. Mukherjee**.*Design Automation Issues for Biofluidic Microchips. In Proc. Int. Conf. on Computer Aided Design,Nov,2005.*

[15] **K. F. Bohringer**. *Modeling and Controlling Parallel Tasks in Droplet-based Microfluidic Systems. IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, 25:329339, Feb 2006.*

[16] **F. Su and K. Chakrabarty**. *Architectural-level synthesis of digital microfluidics-based biochips. In Proc. Int. Conf. on Computer Aided Design, Nov 2004.*

[17] **F. Su, K. Chakrabarty, and R. B. Fair**. *Microfluidics-Based Biochips: Technology Issues, Implementation Platforms, and Design-Automation Challenges. IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems,25:211223, Feb 2006.*

[18] **F. Su, W. Hwang, and K. Chakrabarty**. *Droplet Routing in the Synthesis of Digital Microfluidic Biochips. In Proc. Design,Automation and Test in Eurpoe, 2006.*

[19] **T. Xu and K. Chakrabarty**. *Integrated Droplet Routing in the Synthesis of Microfluidic Biochips. In Proc. Design Automation Conf., Jun 2007.*

[20] **P. H. Yuh, C. L. Yang and Y. W. Chang**. *Placement of Digital Microfluidic Biochips using the T-tree Formulation. In Proc.Design Automation Conf., Jul 2006.*

[21] **J. Peng and S. Akella**. *Coordinating multiple robots with kinodynamic constraints along specified paths. In Proc. Workshop Algorithmic Foundations Robotics, 2002.*

[22] **R. Fung,V. Betz and Chow, W.**, *Simultaneous short-path and long-path timing optimization for FPGAs, Proc. IEEE/ACM International Conference on ComputerAided Design, pp. 838845, 2004.*

[23] **L. McMurchie, C. Ebeling, and PathFinder**: *a negotiation-based performance-driven router for FPGAs, Proc. ACM Symposium on FPGAs, pp. 111117, 1995.*

[24] **M. Majer, C. Bobda, A. Ahmadinia and J. Teich**, *Packet routing in dynamically changing networks on chips, Proc. Parallel and Distributed Processing Symposium,pp. 154b154b, 2005.*

[25] **J. Babb, R. Tessier, and A. Agarwal**, *Virtual wires: overcoming pin limitations on FPGA-based logic emulators, Proc. IEEE Workshop on FPGA for Custom Computing Machines, pp. 142151, 1993.*

[26] **A.J. Pfeiffer, T. Mukherjee, and S. Hauan**, *Simultaneous design and placement of multiplexed chemical processing systems on microchip, Proc. IEEE/ACM International Conference on Computer Aided Design, pp. 229236, 2004.*

[27] **C. Sechen**, *VLSI Placement and Global Routing Using Simulated Annealing, Kluwer Academic Publishers, Boston, MA, 1988.*

[28] **N. Sherwani**, *Algorithms for VLSI Physical Design Automation, Kluwer Academic Publishers, Norwell, MA, 1995.*

[29] **S. Sait and H. Youssef**, *VLSI Physical Design Automation: Theory and Practice,IEEE Press, NY, 1995.*

[30] **T. S. Sammarco and M. A. Burns**. *Thermocapillary pumping of discrete droplets in microfabricated analysis devices. AIChe Journal, 45:350366, 1999.*

[31] **F. Su and K. Chakrabarty**. *Architectural-level synthesis of digital microfluidics-based biochips. In Proc. Int. Conf. on Computer Aided Design, Nov 2004.*

[32] **F. Su and K. Chakrabarty**. *Module Placement for Fault-Tolerant Microfluidics-Based Biochips. ACM Trans. on Design Automation of Electronics Systems, 11:682710, 2006.*

[33] **P. H. Yuh, C. L. Yang , and Y. W. Chang** *BioRoute:A Network-Flow Based Routing Algorithm for Digital Microfluidic Biochips*

[34] **www.nanolytics.com**