

# A Droplet-Manipulation Method for Achieving High-Throughput in Cross-Referencing-Based Digital Microfluidic Biochips

Tao Xu, *Student Member, IEEE*, and Krishnendu Chakrabarty, *Fellow, IEEE*

**Abstract**—Digital microfluidic biochips are revolutionizing high-throughput DNA, immunoassays, and clinical diagnostics. As high-throughput bioassays are mapped to digital microfluidic platforms, the need for design automation techniques for pin-constrained biochips is being increasingly felt. However, most prior work on biochips computer-aided design has assumed independent control of the underlying electrodes using a large number of (electrical) input pins. We propose a droplet-manipulation method based on a “cross-referencing” addressing method that uses “row” and “columns” to access electrodes. By mapping the droplet-movement problem on a cross-referenced chip to the clique-partitioning problem from graph theory, the proposed method allows simultaneous movement of a large number of droplets on a microfluidic array. Concurrency is enhanced through the use of an efficient scheduling algorithm that determines the order in which groups of droplets are moved. The proposed design-automation method facilitates high-throughput applications on a pin-constrained biochip, and it is evaluated using random synthetic benchmarks and a set of multiplexed bioassays.

**Index Terms**—Bioassays, droplet-based microfluidics, droplet routing, lab-on-chip, physical design.

## I. INTRODUCTION

MICROFLUIDICS technology has made great strides in recent years [1]–[3]. Promising applications of this emerging technology include high-throughput deoxyribonucleic acid (DNA) sequencing, immunoassays, environmental toxicity monitoring, and point-of-care diagnosis of diseases [4]. Microfluidics-based miniaturized devices, often referred to in the literature as biochips, are being increasingly used for laboratory procedures involving molecular biology.

Currently, most commercially available biochips rely on continuous fluid flow in etched microchannels. Fluid flow is controlled either using micropumps and microvalves [2] or using electrokinetics [5]. An alternative category of microfluidic biochips relies on the principle of electrowetting on dielectric. Discrete droplets of nanoliter volumes can be manipulated

in a “digital” manner on a 2-D electrode array. Hence, this technology is referred to as “digital microfluidics” [1].

A digital microfluidic biochip typically consists of a patterned metal electrode array (e.g., chrome or indium tin oxide), on which fluid-handling operations such as merging, splitting, mixing, and dispensing of nanoliter droplets containing biological samples are executed. Electrodes are connected to control pins for electrical activation. A number of prototypes of such biochips use a direct-addressing scheme for the control of electrodes [6]. Each electrode is connected to a dedicated control pin; it can therefore be activated independently. This method allows the maximum freedom of droplet manipulation, but it necessitates an excessive number of control pins for practical biochips. As more bioassays are concurrently executed on digital microfluidic platforms [7], [8], system complexity and the number of electrodes are expected to steadily increase. A large number of control pins and the associated interconnect routing problem significantly add to product cost. Thus, the design of pin-constrained digital microfluidic arrays is of great practical importance for the emerging marketplace.

Electrode addressing methods that allow the control of digital microfluidic arrays with a small number of pins are now receiving attention. The method presented in [9] uses array partitioning and careful pin assignment to reduce the number of control pins. However, this method leads to a mapping of pins to electrodes that is specific to a target biofluidic application. An improved design method based on array partitioning is presented in [10], but it is also specific to a given bioassay. A more promising design uses row and column addressing, which is referred to as “cross referencing.” An electrode is connected to two pins, corresponding to a row and a column, respectively [11]. However, due to the problem of electrode interference, a cross-referencing method is particularly prone to unintentional droplet movement when it attempts to simultaneously move more than two arbitrarily positioned droplets. This limitation is a major drawback for high-throughput applications, such as DNA sequencing and large-scale proteomic analysis [12].

In this paper, we propose an automated droplet-manipulation method based on the cross-referencing design for high-throughput applications. The method includes a power-efficient high-throughput droplet manipulation scheme which allows concurrent transportation of multiple droplets. The graph-theoretic concept of clique partitioning is used to determine groups of droplet that can simultaneously be moved on the microfluidic array. To enhance the efficiency of the grouping method, a routing-scheduling algorithm is introduced to

Manuscript received December 29, 2007; revised May 12, 2008. Current version published October 22, 2008. This work was supported by the National Science Foundation under Grant CCF-0541055. An earlier version of this paper was published in Proc. IEEE Design Automation and Test in Europe Conference, pp. 552–557, 2007. This paper was recommended by Associate Editor D. Z. Pan.

The authors are with the Department of Electrical and Computer Engineering, Duke University, Durham, NC 27708 USA (e-mail: tx@ee.duke.edu; krish@ee.duke.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCAD.2008.2006086

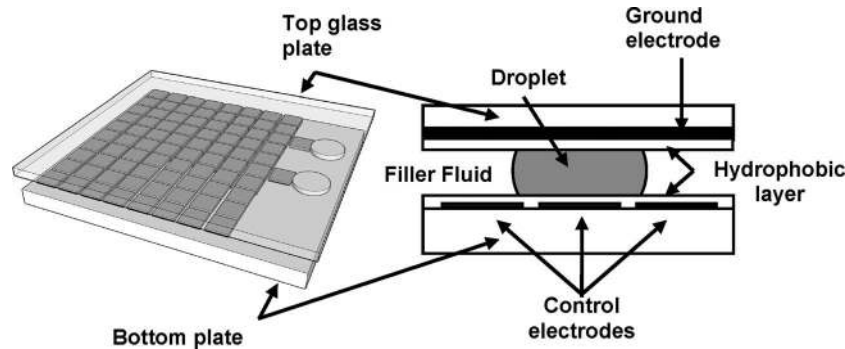


Fig. 1. Schematic diagram of a digital microfluidic biochip.

generate well-aligned manipulation snapshots from the target droplet-routing plan, thereby facilitating high throughput operations on cross-referencing-based chips.

The proposed method provides the means for carrying out high-throughput bioassays on a cross-referencing-based chip through the activation of a small number of electrodes. Therefore, this approach is particularly useful for low-power applications such as battery-operated sensors for environmental monitoring and hand-held devices for point-of-care diagnosis.

For other applications where power consumption is not a significant concern, a variant of the proposed method is presented to further enhance droplet-manipulation throughput.

The rest of this paper is organized as follows. Section II provides an overview of digital microfluidic biochips. In Section III, we discuss related prior work on pin-constrained biochip design for high-throughput applications. Section IV maps the droplet manipulation problem to graph theory and presents the proposed power-efficient high-throughput droplet manipulation method. Section V describes the route scheduling algorithm for droplet routing, which leads to enhanced concurrence of droplet movement. Section VI presents a power-oblivious version of the design technique, which leads to higher throughput. The proposed methods are evaluated using random synthetic benchmarks in Section VII. A multiplexed bioassay is also used as a case study. Finally, conclusions are drawn in Section VIII.

## II. DIGITAL MICROFLUIDIC BIOCHIPS

In this paper, we consider digital microfluidic biochips that rely on the principle of electrowetting on dielectric. Droplets of nanoliter volumes, which contain biological samples, are manipulated on a 2-D electrode array [1]. A unit cell in the array includes a pair of electrodes that acts as two parallel plates. In most prototype digital microfluidic biochips based on the direct-addressing scheme, the bottom plate contains a patterned array of individually controlled electrodes, and the top plate is coated with a continuous ground electrode. A droplet rests on a hydrophobic surface over an electrode, as shown in Fig. 1. Recently, coplanar microfluidic devices, i.e., arrays without a top plate, have also been demonstrated [13]. Using the electrowetting phenomenon, droplets can be moved to any location on a 2-D array. An alternative category of digital microfluidic biochips utilizes orthogonally placed rows of pins on the top and bottom plates. A unit cell can be activated by

selecting orthogonally positioned pins on the top and bottom plate that cross at this cell.

Both designs move droplets by applying a control voltage to a unit cell adjacent to the droplet and, at the same time, deactivating the one just under the droplet. This electronic method of wettability control creates interfacial tension gradients that move the droplets to the charged electrode. Fluid-handling operations such as droplet merging, splitting, mixing, and dispensing can be executed in a similar manner. For example, mixing can be performed by routing two droplets to the same location and then turning them about some pivot points [14]. The digital microfluidic platform offers the additional advantage of flexibility, referred to as reconfigurability, since fluidic operations can be performed anywhere on the array. Droplet routes and operation scheduling result are programmed into a microcontroller that drives electrodes in the array. In addition to electrodes, optical detectors such as LEDs and photodiodes are also integrated in digital microfluidic arrays to monitor colorimetric bioassays [7].

Demonstrated applications of digital microfluidics include the on-chip detection of explosives such as commercial-grade 2,4,6-trinitrotoluene (TNT) and pure 2,4-dinitrotoluene [6], automated on-chip measurement of airborne particulate matter [15], [16], and colorimetric assays [7]. Digital microfluidic biochips are being designed for on-chip gene sequencing through synthesis [15], and integrated hematology, pathology, molecular diagnostics, cytology, microbiology, and serology on the same platform [17]. A prototype has been developed for pyrosequencing [17], which targets the simultaneous execution of  $10^6$  fluidic operations and the processing of billions of droplets. Other lab-on-chip systems are being designed for protein crystallization, which requires the concurrent execution of hundreds of operations [18]. A commercially available droplet-based (using dielectrophoresis) lab-on-chip embeds more than 600 000  $20\ \mu\text{m}$  by  $20\ \mu\text{m}$  electrodes with integrated optical detectors [19].

In a recent review paper on the use of microfluidics for protein crystallization [20], the following question was posed: can we purchase identical crystallization devices, produced under adequate quality control? The authors go on to say, "Drawing upon integrated circuits as an analogy, microfluidics devices may be reducible to a standard set of discrete operations which can then be custom assembled to form more complex operations as needed. With this approach, the success of manufacturing investment does not have to rest upon a single application."

The discrete droplet-based biochip being considered in this paper is perfectly suited as a platform technology, since it avoids the common pitfall of custom devices offered by other continuous-flow microfluidic technologies.

### III. RELATED PRIOR WORK

Recent years have seen a steady increase in the level of integration and system complexity of digital microfluidic biochips [15]. These advances in technology serve as a powerful driver for research on computer-aided design tools for biochip design. Classical architectural and geometric-level synthesis method can be adapted for the automated design of biochips that can execute laboratory protocols [21]–[25]. A unified synthesis method, which combines operation scheduling, resource binding, and module placement, has been proposed in [22]. Systematic droplet routing strategies have also been developed [23], [26]–[28]. These early design automation techniques are useful for biochip design and rapid prototyping, but they all rely on the availability of a direct-addressing scheme.

Pin-constrained design of digital microfluidic biochips was recently proposed in [9]. This method uses array partitioning and careful pin assignment to reduce the number of control pins. However, it requires detailed information about the scheduling of assay operations, microfluidic module placement, and droplet routing pathways. Thus, the array design in such cases is specific to a target biofluidic application. An improved design method based on array partitioning is presented in [10] but it is also specific to a given bioassay.

In another method proposed in [7], the number of control pins for a fabricated electrowetting-based biochip is minimized by using a multiphase bus for the fluidic pathways. Every  $n$ th electrode in an  $n$ -phase bus is electrically connected, where  $n$  is small number (typically  $n = 4$ ). Thus, only  $n$  control pins are needed for a transport bus, irrespective of the number of electrodes that it contains. Although the multiphase bus method is useful for reducing the number of control pins, it is only applicable to a 1-D (linear) array.

An alternative method based on a cross-reference driving scheme is presented in [11]. This method allows control of an  $N \times M$  grid array with only  $N + M$  control pins. The electrode rows are patterned on both the top and bottom plates, and orthogonally placed. In order to drive a droplet along the  $X$ -direction, electrode rows on the bottom plate serve as driving electrodes, while electrode rows on the top serve as reference ground electrodes. The roles are reversed for movement along the  $Y$ -direction, as shown in Fig. 2. This cross-reference method facilitates the reduction of control pins. However, due to electrode interference, this design is particularly prone to unintentional droplet manipulation; the simultaneous movement of more than two droplets is attempted. The resulting serialization of droplet movement is a serious drawback for high-throughput applications.

The minimization of the assay completion time, i.e., the maximization of throughput, is essential for environmental monitoring applications where sensors can provide early warning. Real-time response is also necessary for surgery and neonatal clinical diagnostics. Finally, biological samples are sensitive to

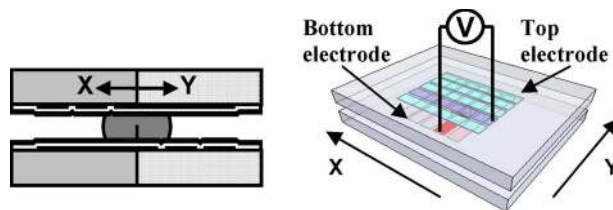


Fig. 2. Cross sections of a cross-referencing microfluidic device that uses single-layer driving electrodes on both top and bottom plates.

the environment and to temperature variations, and it is difficult to maintain an optimal clinical or laboratory environment on chip. To ensure the integrity of assay results, it is therefore desirable to minimize the time that samples spend on-chip before assay results are obtained.

Increased throughput also improves operational reliability. Long assay durations imply that high actuation voltages need to be maintained on some electrodes, which accelerate insulator degradation and dielectric breakdown, reducing the number of assays that can be performed on a chip during its lifetime.

Recently, Griffith *et al.* [29] proposed a droplet-manipulation method that can ensure concurrent manipulation of multiple droplets without unintentional droplet movements. While this method leads to higher throughput for cross-referencing-based biochips, the increase in throughput is limited by the precomputed droplet routing pathways. To overcome this bottleneck, efficient algorithms are needed to generate efficient routing pathways that facilitate high-throughput droplet manipulation.

Moreover, the droplet-manipulation method of [29] is inefficient in terms of power consumption. Power consumption is typically neglected in direct-addressing-based chips because it has been found to be negligible for small prototypes (the activation of one cell typically requires  $1 \sim 2 \mu\text{W}$  power) [30]. However, in a cross-referencing-based design, to activate a single unit cell, two pins (a column pin and a row pin) must be activated. For an  $N \times N$  array, the current drawn by a pin is  $N$  times that for a direct-addressing chip, since all electrodes on the corresponding row or column are activated. Therefore, the power consumption for activating a cell in a cross-referencing-based chip is also  $N$  times higher than that for a direct-addressing chip. The method in [29] relies on the activation of multiple column and row pins simultaneously to carry out concurrent droplet manipulations. Therefore, this scheme can potentially lead to a sharp increase in biochip power consumption. High power consumption is a serious problem for battery-driven chemical/biosensors and hand-held lab-on-chip devices. High power consumption (and the resulting on-chip heat generation) is also detrimental for thermal-sensitive samples such as proteins, which are commonly used in bioassays [31]. Therefore, a power-efficient droplet-manipulation method is needed for cross-referencing biochips.

### IV. POWER-EFFICIENT INTERFERENCE-FREE DROPLET MANIPULATION BASED ON DESTINATION-CELL CATEGORIZATION

In this section, we focus on the problem of manipulating multiple droplets based on digital microfluidic biochips that use cross-referencing to address the electrodes.

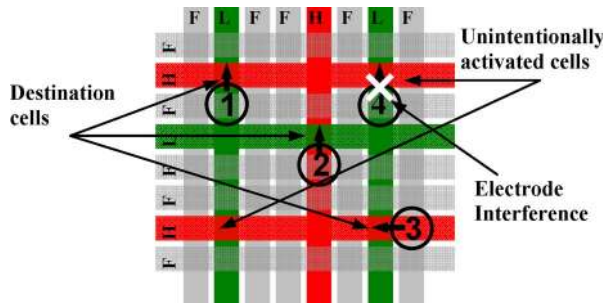


Fig. 3. Example to illustrate the problem of electrode interference. H/L stands for high/low voltage pairs to activate the cells, and unselected row/column pins are left floating (F).

A. Electrode Interference

For the concurrent manipulation of multiple droplets on a cross-referencing-based biochip, multiple row and column pins must be selected to activate the destination cells, i.e., cells to which the droplets are supposed to move. However, the selected row and column pins may also result in the activation of cells other than the intended droplet destinations. An example is shown in Fig. 3. The goal here is to route Droplets 1–3 simultaneously to their destination cells. Droplet 4 is supposed to remain in its current location. However, two additional cells are activated unintentionally when the activation voltage is applied to the row and column pins corresponding to the destination cells. As a result, Droplet 4 is unintentionally moved one cell up (along the *Y*-direction). We refer to this phenomenon as electrode interference.

B. Fluidic Constraints

Droplet manipulations must also conform to rules referred to as the fluidic constraints [23]. These constraints are given by the following set of inequalities: 1)  $|P_i(t) - P_j(t)| \geq 2$ ; 2)  $|P_i(t + 1) - P_j(t)| \geq 2$ ; 3)  $|P_i(t) - P_j(t + 1)| \geq 2$ ; 4)  $|P_i(t + 1) - P_j(t + 1)| \geq 2$ , where  $P_i(t)$  is the position of droplet  $i$  at time  $t$  and  $P_j(t)$  is the position of droplet  $j$  at time  $t$ . The fluidic constraints specify the minimum distance between droplets needed to avoid unintentional fluidic operations that arise due to the overlapping of droplets over adjacent electrodes. These constraints apply to both direct-addressing-based and cross-referencing-based biochips.

C. Destination-Cell Categorization

As shown in Fig. 3, the concurrent manipulation of multiple droplets must be carried out without introducing any electrode interference. For simplicity, here we only focus on the implementation of a set of multiple droplet manipulations that can be carried out concurrently (in a single routing step) on a direct-addressing-based chip, without violating any fluidic constraints. We refer to such a set of droplet manipulations as a *droplet-manipulation snapshot*.

We propose a solution based on destination-cell categorization. Note that the problem highlighted in Fig. 3 can be avoided if the destination cells of the droplets being simultaneously moved reside on the same column or row. However, electrode interference may still occur within the same column or row, as

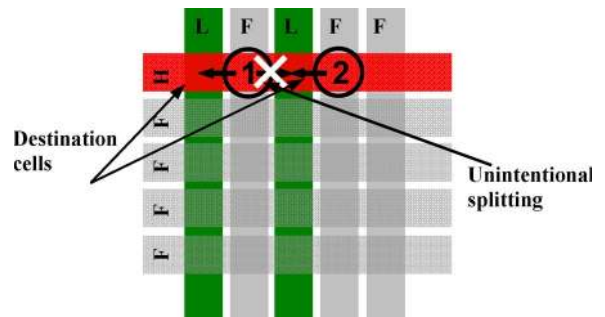


Fig. 4. Example of electrode interference within the same row.

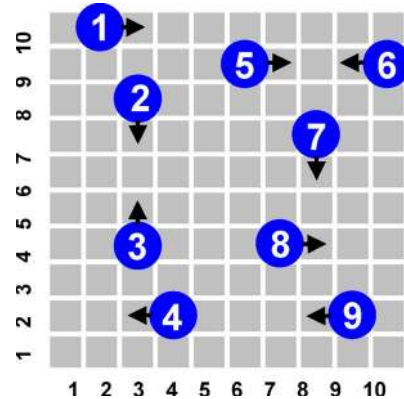


Fig. 5. Example to illustrate destination-cell-based categorization.

shown in Fig. 4. Suppose Droplet 1 and Droplet 2 are both moved one cell to the left at the same time. Although no additional cells are unintentionally activated, Droplet 1 undergoes unintentional splitting in this situation. Fortunately, further scrutiny reveals that the situation in Fig. 4 is only a false alarm. The intended multiple droplet manipulation violates the constraint  $|P_i(t + 1) - P_j(t)| \geq 2$ . Such manipulations cannot be carried out concurrently even on a direct-addressing-based chip. Thus, they will never appear in a single droplet-manipulation snapshot. Therefore, it is safe to carry out concurrent manipulation of multiple droplets whose destination cells are accessed by the same column or row.

On the basis of the above observations, we consider the droplets that can simultaneously be moved as part of the bioassay, and place them in different groups. A group consists of droplets whose destination cells share the same column or row. An example is shown in Fig. 5. A total of nine droplets are needed to be moved on a  $10 \times 10$  array. As discussed above, we group the droplet movements according to their destination cells. For example, Droplets 4 and 9 form a group since the destination cells, in both cases, reside on Row 2. Similarly, Droplets 1, 2, and 3 are placed in the same group since they are all moving to Column 3. Following this grouping process, we finally get four groups of droplets, i.e.,  $\{4, 9\}$ ,  $\{1, 2, 3\}$ ,  $\{5, 6\}$ ,  $\{7, 8\}$ .

In this way, the manipulation of multiple droplets is ordered in time; droplets in the same group can simultaneously be moved without electrode interference, but the movements for the different groups must be sequential. For example, droplet movements for the group  $\{4, 9\}$  in Fig. 5 can simultaneously be carried, as shown in Fig. 6. Droplet movements are carried

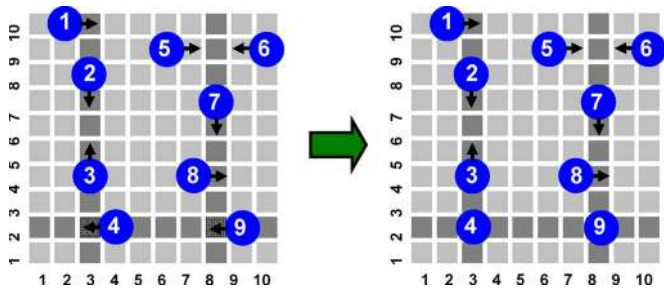


Fig. 6. Example to illustrate the concurrent movement of a group of droplets.

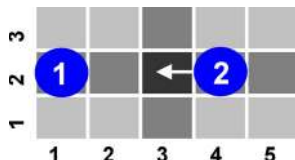


Fig. 7. Example of potential electrode interference due to asynchronous processing of multiple droplet manipulations.

out one group after another until all the droplet movements are completed.

Note that the ordering of droplet movements based only on the above grouping strategy can cause electrode interference and inadvertent mixing. An example is shown in Fig. 7. The movement of Droplet 2 alone to the left by activating Column 3 will not influence Droplet 1. Similarly, the movement of Droplet 1 alone to the right by activating Column 2 will not influence Droplet 2. Note that if these two droplets are concurrently moved, as determined by the grouping procedure, by the activation of (Column 2, Row 2) and (Column 3, Row 2), they mix at (3, 2). However, manipulations of this type violate the fluidic constraint given by  $|P_i(t + 1) - P_j(t + 1)| \geq 2$ . Thus, they cannot exist in a single droplet-manipulation snapshot. Therefore, it is safe to carry out the droplet manipulations in a single manipulation snapshot with an arbitrary ordering.

Although the grouping of droplets based on destination cells reduces the number of droplets that can simultaneously be moved, this approach provides more concurrence than the baseline method of moving one droplet at a time. Compared to direct addressing, an order of magnitude reduction in the number of control pins is obtained. Simulation results in Section VII show that there is only a small increase in the bioassay processing time compared to direct addressing. The above droplet-manipulation method is focused on minimizing power consumption because, in each step, only droplet manipulations that involve a single column or row are carried out. Additional droplet movements are typically possible, but concurrence is traded off for power in this method. An extension to allow higher concurrence is described in Section VI.

#### D. Graph-Theoretic Model and Clique Partitioning

We have thus far introduced the basic idea of multiple droplet manipulations based on destination-cell categorization, and shown that the droplets in each group can simultaneously be moved. Assuming that each step takes constant processing time, the total completion time for a set of droplet movement operations is determined by the number of groups derived from

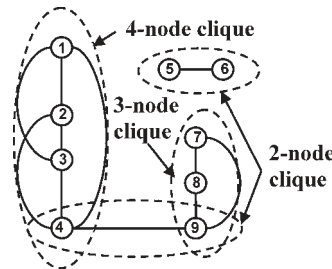


Fig. 8. Mapping of destination-cell layout to an undirected graph.

the categorization of destination cells. Note however that the grouping need not be unique. For instance, in the example of Fig. 5, we can form four groups, i.e.,  $\{4, 9\}$ ,  $\{1, 2, 3\}$ ,  $\{5, 6\}$ , and  $\{7, 8\}$ . However,  $\{1, 2, 3, 4\}$ ,  $\{5, 6\}$ ,  $\{7, 8, 9\}$  is also a valid grouping of the droplets. The latter grouping is preferable because three groups allow more concurrence and, therefore, lower bioassay completion time.

The problem of finding the minimum number of groups can directly be mapped to the clique-partitioning problem from graph theory [32]. To illustrate this mapping, we use the droplet manipulation problem defined in Fig. 5. Based on the destinations of the droplets, an undirected graph, referred to as the droplet-movement graph (DMG), is constructed for each time-step (see Fig. 8). Each node in the DMG represents a droplet. An edge in the graph between a pair of nodes indicates that the destination cells for the two droplets either share a row or a column. For example, Nodes 1 and 2, which represent the Droplet 1 and Droplet 2, respectively, are connected by an edge because the destination cells for these droplets are accessed using Column 3 in the array. Similarly, Nodes 4 and 9 are connected by an edge because the corresponding destination cells are addressed using the same row.

A clique in a graph is defined as a complete subgraph, i.e., any two nodes in this subgraph are connected by an edge [32]. Clique partitioning refers to the problem of dividing the nodes into overlapping subsets such that the subgraph induced by each subset of nodes is a clique. A minimal clique partition is one that covers the nodes in the graph with a minimum number of nonoverlapping cliques. The grouping of droplets as discussed above is equivalent to the clique-partitioning problem. The categorization of destination cells using the grouping of droplets is equivalent to the problem of determining a minimal clique partition. Cliques of different sizes for a given DMG are shown in Fig. 8. A minimal clique partition here is given by  $\{1, 2, 3, 4\}$ ,  $\{5, 6\}$ ,  $\{7, 8, 9\}$ , which corresponds to the groups derived above. Although the general clique-partitioning problem is known to be NP-hard [33], a number of heuristics are available in the literature to solve it in an efficient manner.

#### E. Algorithm for Droplet Grouping

Next, we describe a greedy algorithm to determine a (minimal) clique partition for the DMG. The algorithm determines cliques for the DMG in an iterative manner.

The largest clique is first determined and then nodes and edges corresponding to this clique are deleted from the graph. Next, the clique searching procedure is applied to the reduced graph. The algorithm terminates when all the nodes in the DMG

have been deleted, i.e., an empty graph is obtained. The computational complexity of this algorithm is linear in the number of rows/columns. Recall that the cliques can only be formed among nodes sharing the same row or column. Therefore, the largest clique can be determined by scanning the columns and rows of the array. Thus, a maximum of only  $N + M$  iterations are needed for the DMG derived from an  $N \times M$  array.

Note that although in each step of the above algorithm, the largest clique and the associated destination cells are deleted, the absence of the corresponding destination cells does not lead to any added complexity for droplet movement. This is because the droplet movements involving these destination cells are incorporated in the clique determined at this step. Therefore, when the algorithm terminates with an empty graph, all droplet movements have been processed without any electrode interference.

The steps of the complete procedure to determine the order of droplet movements can be stated as follows.

- 1) Obtain the required droplet movements (from a synthesis tool such as [23]), and organize these movements in the form of snapshots corresponding to different time-steps. The fluidic constraints described in Section IV-B need to be satisfied for each snapshot.
- 2) Compare consecutive snapshots to determine the destination cells for the droplets.
- 3) Scan each row and each column to find the row/column with the largest set of destination cells. The destination cells thus determined form a group of droplets that can simultaneously be moved. If no row/column contains more than one destination cells, set the flag END to 1.
- 4) If  $END = 1$ , process the remaining movements in multiple steps, but with two droplets at each step. Else, carry out the droplet movements indicated by Step 3).
- 5) Check if all the movements in the snapshot have been processed. If the check yields a negative outcome, repeat Step 3).
- 6) Check whether all the snapshots are processed. If not, get the next snapshot and repeat Step 2), else terminate the procedure.

V. SCHEDULING OF ROUTING FOR EFFICIENT GROUPING

The column- and row-scan methods described above enable the simultaneous manipulation of multiple droplets on the cross-referencing chip. However, the efficiency of this approach depends on the prealignment of the destination cells corresponding to the droplet movements in the target droplet-routing snapshot. The better aligned the destination cells are, i.e., they share the same column/row, the larger the number is of droplets that can simultaneously be moved. Therefore, to increase efficiency, it is important to generate routing snapshots with well-aligned destination cells.

A. Routing Plan Decomposition

Note that routing snapshots are obtained from the schedule of droplet movements corresponding to the droplet-routing plan. Typically, several schedules are feasible for a given droplet-routing plan. For example, the routing plan shown in Fig. 9,

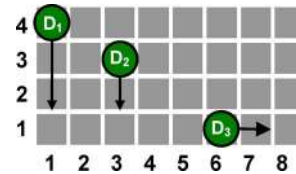
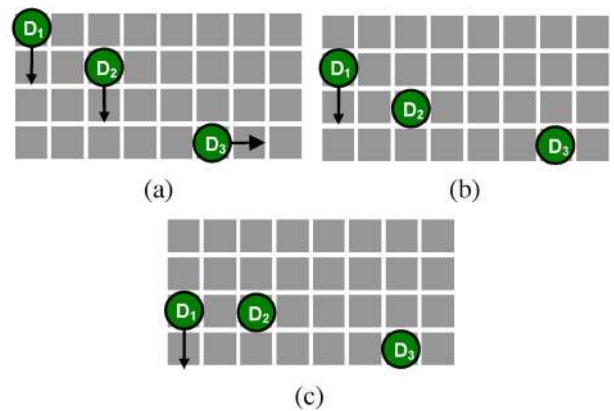


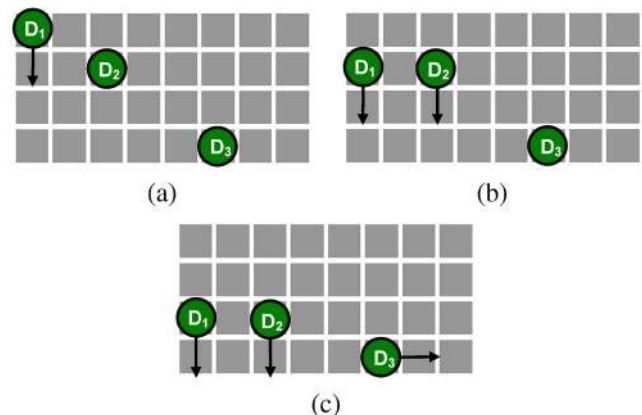
Fig. 9. Example of a droplet-routing plan.



Snapshot	1	2	3
Activation sequence	{R3, C1}, {R2, C3}, {R1, C7}	{R2, C1}	{R1, C1}

(d)

Fig. 10. Schedule A. (a) Snapshot 1. (b) Snapshot 2. (c) Snapshot 3. (d) Activation sequence (in terms of rows and columns).



Snapshot	1	2	3
Activation sequence	{R3, C1}	{R2, C1&C3}	{R1, C1&C3&C6}

(d)

Fig. 11. Schedule B: (a) Snapshot 1. (b) Snapshot 2. (c) Snapshot 3. (d) Activation sequence (in terms of rows and columns).

derived from routing methods such as [23], can be scheduled, i.e., implemented, in two ways (see Figs. 10 and 11).

Using the grouping algorithm described in Section IV, the three snapshots in Fig. 10 take five manipulation steps in total (three steps for Snapshot 1, one step for Snapshot 2, and one step for Snapshot 3). However, the three snapshots in Fig. 11 only require three manipulation steps (one step for each). Therefore, the key to obtaining a well-aligned snapshot is the schedule of droplet routes. Compared to the schedule in Fig. 10, the schedule in Fig. 11 carefully orders the droplet movements.

---

**Scheduling algorithm for droplet routing**


---

- 1 Start from the droplet closest to the boundary of the array, whose initial moving direction facing the inner of the array. Mark its initial moving direction as the *starting direction*. We chose such a droplet as starting point since it is most likely to be far away from all the other droplets.
  - 2 Move this droplet along its pathway until it shares the same column/row with the starting point of another droplet moving in the same direction. Move the two droplets in parallel. By this means, the farthest droplet is “pulled” closer to other droplets, thereby improving the alignment and increasing the probability of concurrent manipulation.
  - 3 Repeat Step 2 and keep adding new droplets for parallel movement until a droplet in the set of droplets moving in parallel confronts either a turn or another droplet in its pathway.
  - 4 Count the number of droplet movements oriented in each of the three directions other than the reverse of the starting direction at the current column or row, and store these numbers in a candidate moving set  $S$ .
  - 5 Choose the direction of the next droplet manipulation corresponding to the largest number of unfinished droplet movements in  $S$ .
  - 6 Repeat Steps 2-5 until no droplet can be moved in the three directions other than the reverse of the starting direction. The reverse direction of the starting direction is excluded in this step because allowing droplet moving in this direction may cause dead loop. Therefore, we handle such movement separately in Step 8.
  - 7 Repeat Steps 1-6 starting from the reverse of the starting direction. This step schedules the leftover droplet movements from Step 6.
  - 8 Add the routing schedule from Step 7 to the end of the schedule obtained from Steps 1-6 in the first iteration to obtain the schedule for the entire routing snapshot.
- 

Fig. 12. Pseudocode of scheduling algorithm for droplet routes.

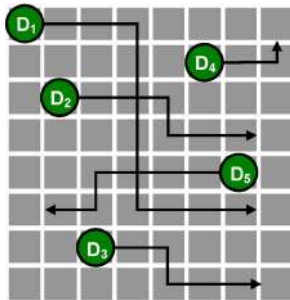


Fig. 13. Routing example to illustrate scheduling algorithm.

Droplets movements are scheduled in such a way that each snapshot brings the droplets closer to each other, therefore their destination cells are more likely to share the same column/row. By this means, the alignment of the droplets for each snapshot is improved, which explains the increase in routing efficiency.

Based on the above observation, we present an efficient scheduling algorithm to generate well-aligned snapshots. The pseudocode for this algorithm is sketched in Fig. 12.

We next use an example to illustrate the algorithm. Fig. 13 shows a routing plan with five droplet pathways. Initially, the droplets reside at the starting points shown in Fig. 13. The algorithm first moves  $D_1$  one electrode to the right. Next,

TABLE I  
RANDOM SYNTHETIC BENCHMARKS, SAMPLE SIZE = 1000

DIP	Array Size	NSR
0.1	25×25	0.31
0.1	50×50	0.24
0.1	75×75	0.19
0.15	25×25	0.28
0.15	50×50	0.20
0.15	75×75	0.14

droplets  $D_1$  and  $D_2$  are moved one electrode to the right in parallel.  $D_3$  is added and moved one electrode to the right with  $D_1$  and  $D_2$ . At this time,  $D_1$  meets a turning point in its pathway. The algorithm stops and counts the number of droplet movements in each direction, i.e., two rightwards, one downwards. Therefore, the algorithm stores these numbers in  $S$  and keeps on moving  $D_2$  and  $D_3$  until they reach their respective turning points. Next, the algorithm moves them one electrode downwards and one electrode to the right when they meet  $D_4$ . Then,  $D_2$ ,  $D_3$  and  $D_4$  are moved one electrode to the right.

Next, the algorithm goes back to move the leftover droplets  $D_1$  and  $D_4$ . Finally, the algorithm generates a routing schedule for  $D_5$  and then integrates it into the schedule for droplets  $D_2 \sim D_4$ .

Next, we evaluate the computational complexity of the algorithm using step by step analysis. We assume that there are  $M$  droplets on an array of  $N \times N$  electrodes. Droplet  $i$  has a pathway of length  $L_i$ . As shown in Fig. 12, Step 1) takes  $O(M)$  time to determine the starting droplet. Step 2) and Step 3) check if there are multiple droplets that can be moved concurrently after each manipulation step. Each checking operation takes  $O(M)$  time. Step 4) calculates the number of droplet movements in different directions and stores this information in  $S$ . This step takes  $O(M)$  time in the worst case. Step 5) also takes  $O(M)$  time. Note that Steps 2)–5) can be repeated. However, the number of repetitions is bound by the sum of the lengths of all the droplet pathways, i.e., in the worst case, Steps 2)–5) are repeated for each single droplet movement. Therefore, the time taken by Steps 2)–5) is simply  $(M + M + M) \sum_{i=1}^M L_i = O(M \sum_{i=1}^M L_i)$ . Note that  $L_i \leq 2N$ , where  $1 \leq i \leq M$ . Thus,  $\sum_{i=1}^M L_i \leq 2MN$ . We therefore conclude that the worst case time complexity for Steps 2)–5) is  $O(NM^2)$ . The time complexity of Step 1) is  $O(N)$ . For Steps 7)–8), we also have  $O(NM^2)$ , therefore, the overall time complexity of the scheduling algorithm is  $O(NM^2)$ .

## VI. VARIANT OF DROPLET-MANIPULATION METHOD FOR HIGH-THROUGHPUT POWER-OBLIVIOUS APPLICATIONS

For applications where power consumption is not critical, the method proposed in the previous section can be modified to achieve even higher throughput.

Note that when we use the low-power manipulation method of Section IV to implement the “aligned” droplet manipulations, in each step, only droplet manipulations corresponding to a single row or column are carried out. Note, however, that there may be other droplet manipulations that can also

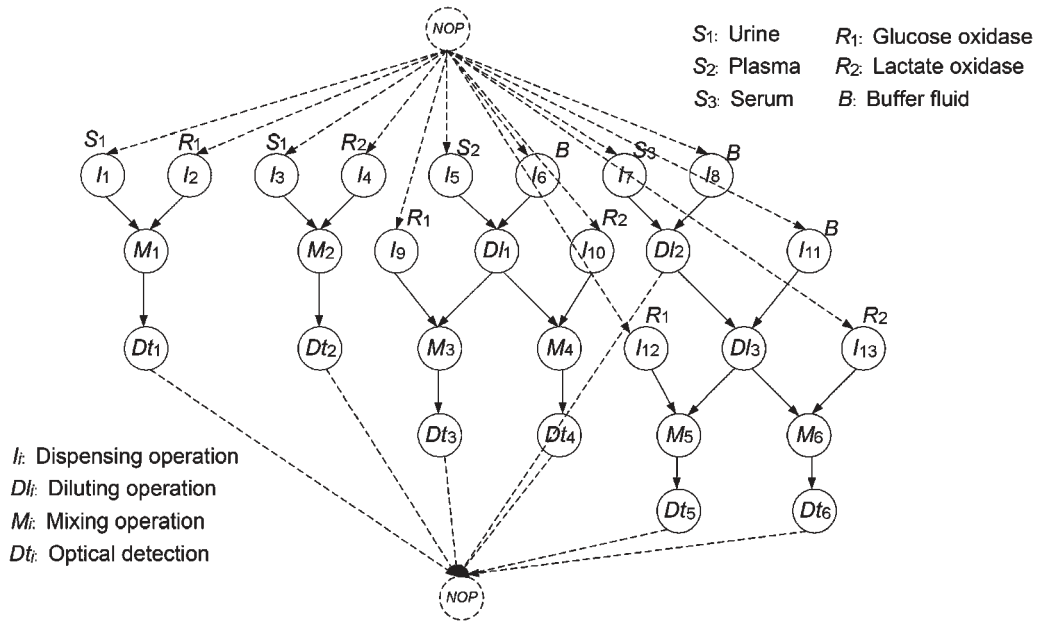


Fig. 14. Sequencing graph model of bioassay example.

be implemented without introducing electrode interference. For example, for the droplet manipulation pattern shown in Fig. 5, manipulation of  $D_7$  and  $D_8$  can also be carried out at the same time when we concurrently move  $D_4$  and  $D_9$ .

The implementation of these “compatible” droplet manipulations will result in higher power consumption, with the associated benefit of higher throughput. Based on this observation, we propose a modified droplet-manipulation method that relies on the method from [29] to carry out the droplet manipulation for the routing plans generated from the proposed routing-scheduling method. Note that the straightforward application of [29] to the droplet routes derived from the routing-scheduling algorithm leads to undesirable consequences. The alignment of the droplet movements will be broken, thereby leading to lower throughput. Therefore, we limit the use of [29] for handling droplet movements in Step 7) of the routing scheduling method, i.e., the ones that correspond to the reverse of the starting direction. Instead of being carried out using additional steps, these droplet manipulations are concurrently carried out with the ones from Step 1)–6). This approach results in higher throughput and reduced assay completion time.

VII. EVALUATION AND SIMULATION RESULTS

In this section, we use random synthetic benchmarks and a set of multiplexed bioassays to evaluate the proposed method.

A. Random Synthetic Benchmarks

We first use random synthetic benchmarks to evaluate the effectiveness of the grouping-based droplet-movement approach. Digital microfluidic arrays of size  $N \times N$ , ( $N = 25, 50, 75$ ) are considered here. For each array, we consider 1000 simulated droplet-movement plans. Each droplet-movement plan is defined by a starting snapshot and destination snapshot. The

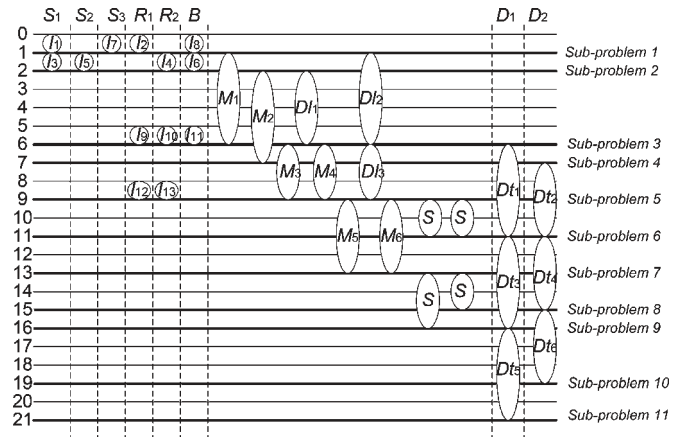


Fig. 15. Schedule obtained via architectural-level synthesis.

starting snapshot is generated by injecting a droplet in the array with probability  $k$ , referred to as the droplet injection probability (DIP). A check is incorporated in the generation process to avoid the violation of fluidic constraints. Results derived from this process can be viewed as snapshots of droplets moving around the chip. Each droplet-movement plan is provided as input to the grouping-based method, and the number of steps required for droplet movement is calculated. One-at-a-time droplet movement is also considered as a baseline, and the results are recorded for the purpose of comparison.

To evaluate the proposed method, we introduce the parameter “number-of-steps-ratio” (NSR), defined by the equation  $NSR = N_p/N_o$ , where  $N_p(N_o)$  is the number of movement steps for the grouping-based method (one-at-a-time baseline method). Small values of NSR are clearly desirable. We calculate the NSR values for different array sizes and the results (see Table I).

As shown Table I, regardless of DIP value, the NSR decreases with array size. This shows that the grouping-based



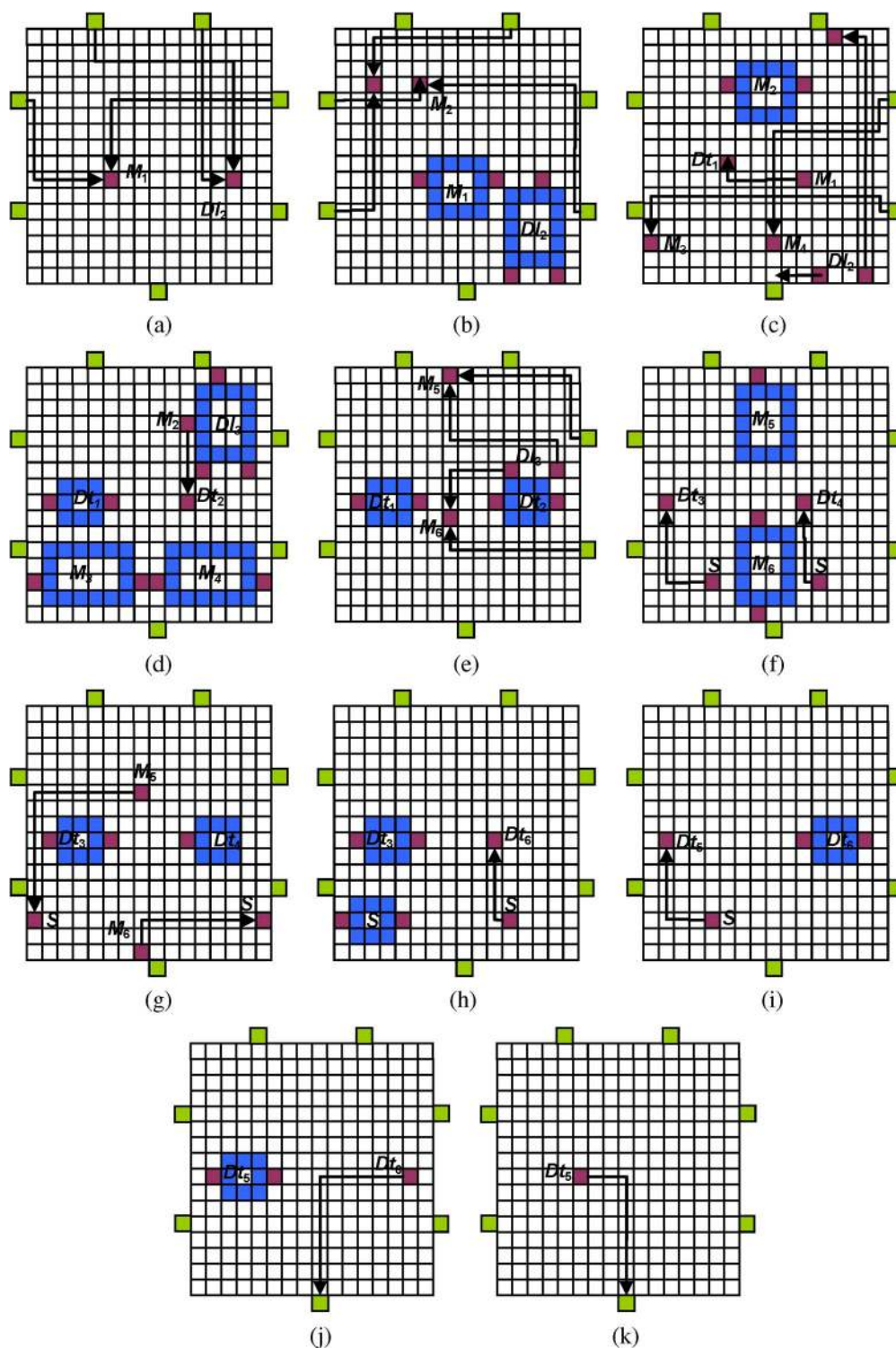


Fig. 16. Module placement (11 subproblems) for the multiplexed bioassay.

method is more efficient for concurrent droplet manipulation on large-scale digital microfluidic arrays. For a given array size, the proposed method achieves lower NSR values for higher values of DIP. Thus, we see that compared to the one-at-a-time scheme, droplets can be manipulated more efficiently for high-throughput biochips with higher concurrence in biochip operations.

### B. Multiplexed Bioassay Example

Next, we evaluate the proposed scheduling and grouping-based droplet-manipulation methods by using them to implement the routing plan for a set of real-life bioassays, namely multiplexed *in vitro* diagnostics on human physiological fluids.

As a typical example of multiplexed and concurrent assays, three types of human physiological fluids, urine, serum, and plasma are sampled and dispensed into the digital microfluidic

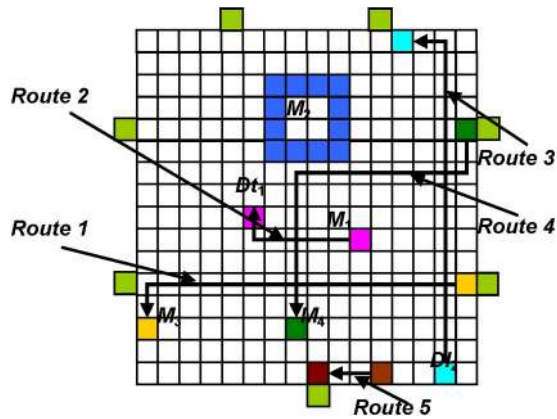


Fig. 17. Routing plan for Subproblem 3.

biochip, and glucose and lactate measurements are performed for each type of physiological fluid. The assay protocol, based on Trinder’s reaction [34], can be modeled by a sequencing graph, as shown in Fig. 14. We assume that the schedule for assay operations and resource binding have been obtained via architectural level synthesis (e.g., through the modified list-scheduling algorithm [21]), as shown in Fig. 15. Note that one time unit in this schedule is set to 2 s. Moreover, assume that a module placement on a 16 × 16 microfluidic array has also been given *a priori*, as shown in Fig. 16.

To find suitable droplet routes for this biochip, the routing problem is decomposed into 11 subproblems, as highlighted in Fig. 16. For each subproblem, a droplet routing plan is generated using the routing algorithm in [23]. Next, we apply the proposed grouping-based manipulation method to implement the routing plan. For comparison, the grouping-based manipulation method is first applied without the scheduling algorithm for aligning droplets. The snapshots are generated by assuming that from the starting point, each droplet moves one electrode along its pathway in every snapshot. Note that fluidic constraints might be violated if two droplets simultaneously move to the next cells. In this case, we can force one of them to stay in the current location at that snapshot, thereby overriding the constraint violation. We calculate the number of manipulations cycles needed using the grouping-based method. Next, we apply the manipulation method combined with the scheduling algorithm. Again, we record the number of droplet-manipulation manipulation cycles required. The computation time for the routing scheduling and the manipulation method for the entire assay is 173 s, on a Intel Core Duo 2-GHz PC with 2G of RAM.

Here, we use Subproblem 3 for illustration. As shown in Fig. 17, the routing plan for Subproblem 3 contains five routing pathways (routes). If no grouping method is used, droplet movements are carried out one per cycle. The total number of cycles needed equals the number of electrode on all the pathways, i.e., 60. However, using the grouping-based manipulation method, the number of cycles is reduced to 53 cycles. Note that no scheduling of droplet routes is used at this time to align droplets. The details of the manipulations in each cycle are listed in Table II, where the arrows indicate that the target droplet is moved one electrode toward the corresponding

TABLE II  
DROPLET MANIPULATION RESULTS FOR GROUPING-BASED METHOD WITHOUT SCHEDULING OF DROPLET ROUTES

Snap shot	Cycle number	Manipulation <D <sub>1</sub> , D <sub>2</sub> , D <sub>3</sub> , D <sub>4</sub> , D <sub>5</sub> >	Snap shot	Cycle number	Manipulation <D <sub>1</sub> , D <sub>2</sub> , D <sub>3</sub> , D <sub>4</sub> , D <sub>5</sub> >
1	1	<←, x, ↑, x, x>	8	28	<←, x, x, x, x>
	2	<x, ←, x, x, x>		29	<x, x, ↑, x, x>
	3	<x, x, x, ↓, x>		30	<x, x, x, ←, x>
	4	<x, x, x, x, ←>		9	31
2	5	<←, x, x, x, x>	32		<←, x, x, x, x>
	6	<x, ←, x, x, x>	10	33	<←, x, x, x, x>
	7	<x, x, ↑, x, x>		34	<x, x, ↑, x, x>
	8	<x, x, x, ↓, x>		35	<x, x, x, ←, x>
	9	<x, x, x, x, ←>	11	36	<←, x, x, x, x>
3	10	<x, x, ↑, ←, x>		37	<x, x, ↑, x, x>
	11	<←, x, x, x, x>	38	<x, x, x, ↓, x>	
	12	<x, ←, x, x, x>	12	39	<←, x, x, x, x>
	13	<x, x, x, x, ←>		40	<x, x, ↑, x, x>
4	14	<←, x, ↑, x, x>	41	<x, x, x, ↓, x>	
	15	<x, ←, x, x, x>	13	42	<←, x, x, x, x>
	16	<x, x, x, ←, x>		43	<x, x, ↑, x, x>
5	17	<←, x, x, x, x>	14	44	<x, x, x, ↓, x>
	18	<x, ←, x, x, x>		45	<←, x, x, x, x>
	19	<x, x, ↑, x, x>		46	<x, x, ↑, x, x>
	20	<x, x, x, ←, x>		47	<x, x, x, ↓, x>
6	21	<←, x, x, x, x>	15	48	<←, x, x, ↓, x>
	22	<x, ↑, x, x, x>		49	<x, x, ↑, x, x>
	23	<x, x, ↑, x, x>	16	50	<↓, x, x, ↓, x>
	24	<x, x, x, ←, x>		51	<x, x, ←, x, x>
7	25	<←, x, x, x, x>	17	52	<↓, x, x, ↓, x>
	26	<x, x, ↑, x, x>		53	<x, x, ←, x, x>
	27	<x, x, x, ←, x>			

direction. An entry “x” indicates that the droplet stays in the current location in that snapshot.

In Table II, two droplets are simultaneously moved in several manipulation cycles. This increases the currency of droplet movements. However, due to the severe misalignment of the snapshots, the number of such concurrent manipulation cycles is rather limited. Therefore, the reduction of routing time is quite modest, less than 12%.

Next, we apply a combination of the proposed cross-based scheduling method and grouping-based droplet-based manipulation to the routing plan. The results are shown in Table III. Compared to the results shown in Table II, droplet-movement concurrence is significantly improved. Half of the cycles now contain the concurrent manipulation of more than two droplets. In some cycles, even four droplets are simultaneously moved. As a result, only 34 cycles are required for this routing plan, which is only 57% of the time required for one-at-a-time droplet, and 64% of the time required if droplet grouping is carried out without route scheduling.

Note that the completion time obtained using the proposed droplet-manipulation method is slightly more than that for the direct-addressing method (34 cycles versus 17 cycles). However, the proposed method requires only 32 (16+16) control pins while 256 (16×16) pins are required for the direct-addressing method.

The above comparison is next carried out for the rest of the subproblems. The results are shown in Fig. 18. Note that some subproblems, e.g., #4, #6, #8, #9, #10, #11, contain only one droplet pathway. For these subproblems, the three methods result in the same number of cycles. Therefore, results for these

TABLE III  
DROPLET MANIPULATION RESULTS FOR GROUPING-BASED METHOD WITH SCHEDULING OF DROPLET ROUTES

Cycle number	Manipulation <D <sub>1</sub> , D <sub>2</sub> , D <sub>3</sub> , D <sub>4</sub> , D <sub>5</sub> >	Cycle number	Manipulation <D <sub>1</sub> , D <sub>2</sub> , D <sub>3</sub> , D <sub>4</sub> , D <sub>5</sub> >
1	<x, x, x, ↓, x>	18	<↓, x, x, ↓, x>
2	<x, x, x, ↓, x>	19	<↓, x, ↑, ↓, x>
3	<←, x, ↑, ←, x>	20	<x, x, ↑, x, x>
4	<←, x, x, ←, x>	21	<x, x, ↑, x, x>
5	<←, x, x, ←, x>	22	<x, x, ↑, x, x>
6	<←, x, x, ←, x>	23	<x, x, ↑, x, x>
7	<←, x, x, ←, ←>	24	<x, x, ↑, x, x>
8	<←, ←, x, ←, ←>	25	<x, x, ↑, x, x>
9	<←, ←, x, ←, ←>	26	<x, x, ↑, x, x>
10	<←, ←, x, ←, x>	27	<x, x, ↑, x, x>
11	<←, ←, x, x, x>	28	<x, x, ↑, x, x>
12	<←, ←, x, x, x>	29	<x, x, ↑, x, x>
13	<x, x, x, ↓, x>	30	<x, x, ↑, x, x>
14	<x, ↑, x, ↓, x>	31	<x, x, ↑, x, x>
15	<x, x, x, ↓, x>	32	<x, x, ↑, x, x>
16	<x, x, x, ↓, x>	33	<x, x, ←, x, x>
17	<x, x, x, ↓, x>	34	<x, x, ←, x, x>

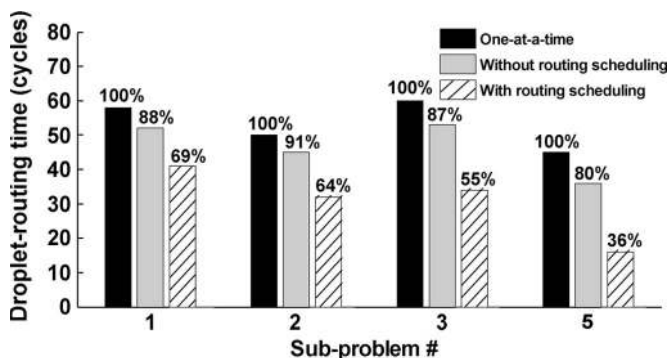


Fig. 18. Comparison of droplet-routing times for four subproblems (#1, #2, #3, #5).

subproblems are not shown explicitly. Results for subproblem 7 are also not shown since the two routing pathways in it have no overlap in any columns or rows, thereby no concurrent droplet movement is possible.

As we can see from Fig. 18, the proposed grouping-based manipulation method always achieves improved concurrence over the one-at-a-time droplet-manipulation method. This improvement is significantly enhanced when the proposed route-scheduling algorithm is applied. The percentage improvement varies with the subproblem, as shown in Fig. 18. For the entire assay, the proposed route-scheduling algorithm leads to a 47% reduction in the number of droplet-manipulation cycles compared to the one-at-a-time droplet-manipulation method, and 32% compared to the grouping-based algorithm alone.

Next, we apply the power-oblivious version of the proposed method to the same problem instance. The computation time for the power-oblivious routing scheduling and manipulation method for the entire assay is 288 s, on an Intel Core Duo 2-GHz PC with 2G of RAM. We compare the results (completion time and power consumption, the latter measured in terms of the average number of pins activated per droplet-manipulation step) with [29] and the power-efficient version of the proposed method.

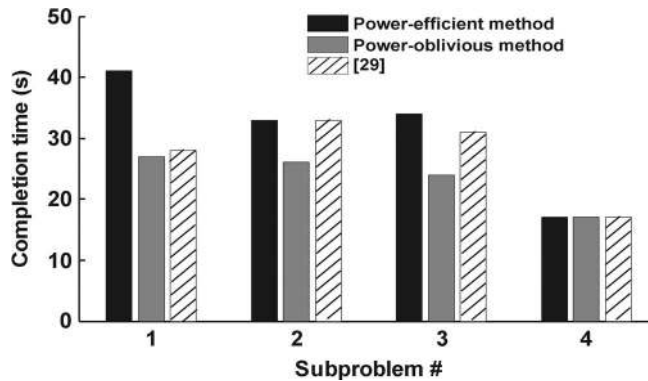


Fig. 19. Comparison of droplet-routing times for the power-efficient-method, the power-oblivious method, and the method from [29].

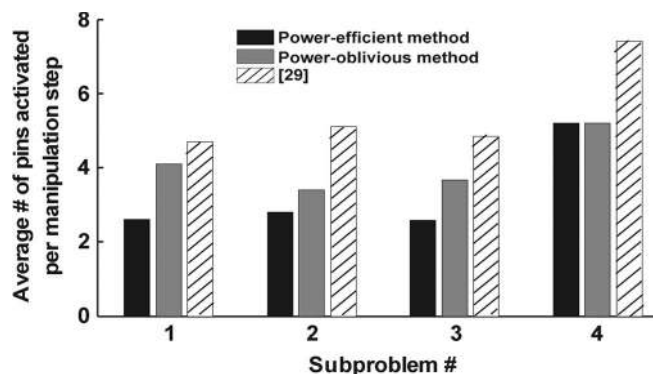


Fig. 20. Comparison of average number of control pins activated per manipulation step for the power-efficient method, the power-oblivious method, and the method from [29].

Figs. 19 and 20 show that although the proposed power-efficient droplet-manipulation method leads to a slightly longer completion time than [29] for some subproblems, it activates a much smaller number of control pins in each manipulation cycle. This results in a significant reduction in power consumption.

Compared to [29], the power-oblivious variant of the proposed method achieves a significant reduction in assay completion time. As a tradeoff, it results in increased power consumption compared to the power-efficient method. However, the expected power consumption is still smaller than in [29].

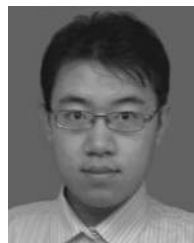
### VIII. CONCLUSION

We have proposed a power-efficient high-throughput droplet manipulation method for a “cross-referencing” addressing method that uses “rows” and “columns” to access electrodes in digital microfluidic arrays. By mapping the droplet-movement problem to the clique-partitioning problem from graph theory, the proposed method allows simultaneous movement of a large number of droplets. A linear-time heuristic algorithm based on row scanning and column scanning has been used to derive the clique partitions. To further increase routing concurrence, a route-scheduling algorithm has been developed to prealign droplet movements. A power-oblivious version of the method is

also presented; it allows higher throughput manipulations with a slight increase in power consumption. We have used random synthetic benchmarks and a set of multiplexed bioassays to evaluate the proposed method. Results show that high throughput can be obtained using a small number of control pins. This paper will allow bioassays for high-throughput sequencing, immunoassays, and clinical diagnostics to be mapped to pin-constrained and low-cost biochips, and simplify the design and implementation of such biochips.

## REFERENCES

- [1] R. B. Fair, V. Srinivasan, H. Ren, P. Paik, V. K. Pamula, and M. G. Pollack, "Electrowetting-based on-chip sample processing for integrated microfluidics," in *IEDM Tech. Dig.*, 2003, pp. 32.5.1–32.5.4.
- [2] E. Verpoorte and N. F. de Rooij, "Microfluidics meets MEMS," *Proc. IEEE*, vol. 91, no. 6, pp. 930–953, Jun. 2003.
- [3] J. Zeng and T. Korsmeyer, "Principles of droplet electrohydrodynamics for lab-on-a-chip," *Lab Chip*, vol. 4, no. 4, pp. 265–277, Aug. 2004.
- [4] T. H. Schulte, R. L. Bardell, and B. H. Weigl, "Microfluidic technologies in clinical diagnostics," *Clin. Chim. Acta*, vol. 321, no. 1/2, pp. 1–10, Jul. 2002.
- [5] Y. Wang, Q. Lin, and T. Mukherjee, "Composable behavioral models and schematic-based simulation of electrokinetic lab-on-a-chip systems," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 25, no. 2, pp. 258–273, Feb. 2006.
- [6] R. B. Fair, A. Khlystov, V. Srinivasan, V. K. Pamula, and K. N. Weaver, "Integrated chemical/biochemical sample collection, pre-concentration, and analysis on a digital microfluidic lab-on-a-chip platform," in *Proc. Lab-on-a-Chip: Platforms, Devices, Appl., Conf.*, 2004, vol. 5591, pp. 113–124.
- [7] V. Srinivasan, V. K. Pamula, and R. B. Fair, "An integrated digital microfluidic lab-on-a-chip for clinical diagnostics on human physiological fluids," *Lab Chip*, vol. 4, no. 4, pp. 310–315, Aug. 2004.
- [8] V. Srinivasan, V. K. Pamula, M. G. Pollack, and R. B. Fair, "Clinical diagnostics on human whole blood, plasma, serum, urine, saliva, sweat, and tears on a digital microfluidic platform," in *Proc. MicroTAS*, 2003, pp. 1287–1290.
- [9] T. Xu, W. Hwang, F. Su, and K. Chakrabarty, "Automated design of pin-constrained digital microfluidic biochips under droplet-interference constraints," *ACM J. Emerging Technol. Comput. Syst.*, vol. 3, no. 3, p. 14, Nov. 2007, article 14.
- [10] T. Xu and K. Chakrabarty, "Droplet-trace-based array partitioning and a pin assignment algorithm for the automated design of digital microfluidic biochips," in *Proc. IEEE/ACM Int. Conf. Hardware/Software Codes. Syst. Synthesis*, 2006, pp. 112–117.
- [11] S.-K. Fan, C. Hashi, and C.-J. Kim, "Manipulation of multiple droplets on  $N \times M$  grid by cross-reference EWOD driving scheme and pressure-contact packaging," in *Proc. IEEE MEMS Conf.*, 2003, pp. 694–697.
- [12] H. Moon, A. R. Wheeler, R. L. Garrell, J. A. Loo, and C. J. Kim, "An integrated digital microfluidic chip for multiplexed proteomic sample preparation and analysis by MALDI-MS," *Lab Chip*, vol. 6, no. 9, pp. 1213–1219, Sep. 2006.
- [13] P. Paik, V. K. Pamula, M. G. Pollack, and K. Chakrabarty, "Coplanar digital microfluidics using standard printed circuit board processes," in *Proc. MicroTAS*, 2005, pp. 566–568.
- [14] P. Paik, V. K. Pamula, M. G. Pollack, and R. B. Fair, "Electrowetting-based droplet mixers for microfluidic systems," *Lab Chip*, vol. 3, no. 1, pp. 28–33, Feb. 2003.
- [15] R. B. Fair, A. Khlystov, T. D. Taylor, V. Ivanov, R. D. Evans, P. B. Griffin, V. Srinivasan, V. K. Pamula, M. G. Pollack, and J. Zhou, "Chemical and biological applications of digital-microfluidic devices," *IEEE Des. Test Comput.*, vol. 24, no. 1, pp. 10–24, Jan./Feb. 2007.
- [16] Y. Zhao and S. K. Cho, "Microparticle sampling by electrowetting-actuated droplet sweeping," *Lab Chip*, vol. 6, no. 1, pp. 137–144, Jan. 2006.
- [17] *Advanced Liquid Logic*. [Online]. Available: <http://www.liquid-logic.com>
- [18] T. Xu, P. Thwar, V. Srinivasan, V. K. Pamula, and K. Chakrabarty, "Digital microfluidic biochip for protein crystallization," in *Proc. IEEE-NIH Life Sci. Syst. Appl. Workshop*, Bethesda, MD, 2007.
- [19] *Silicon Biosystems*. [Online]. Available: [www.siliconbiosystems.com](http://www.siliconbiosystems.com)
- [20] M. van der Woerd, D. Ferree, and M. Pusey, "The promise of macromolecular crystallization in microfluidic chips," *J. Struct. Biol.*, vol. 142, no. 1, pp. 180–187, Apr. 2003.
- [21] S. Fei and K. Chakrabarty, "Architectural-level synthesis of digital microfluidics-based biochips," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des.*, 2004, pp. 223–228.
- [22] S. Fei and K. Chakrabarty, "Unified high-level synthesis and module placement for defect-tolerant microfluidic biochips," in *Proc. Des. Autom. Conf.*, 2005, pp. 825–830.
- [23] S. Fei, W. Hwang, and K. Chakrabarty, "Droplet routing in the synthesis of digital microfluidic biochips," in *Proc. Des. Autom. Test Eur.*, 2006, pp. 323–328.
- [24] A. J. Ricketts, K. Irick, N. Vijaykrishnan, and M. J. Irwin, "Priority scheduling in digital microfluidics-based biochips," in *Proc. DATE Conf.*, 2006, pp. 329–334.
- [25] P.-H. Yuh, C.-L. Yang, and C.-W. Chang, "Placement of defect-tolerant digital microfluidic biochips using the T-tree formulation," *ACM J. Emerging Technol. Comput. Syst.*, vol. 3, no. 3, pp. 13.1–13.32, Nov. 2007.
- [26] K. F. Böhringer, "Modeling and controlling parallel tasks in droplet-based microfluidic systems," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 25, no. 2, pp. 334–344, Feb. 2006.
- [27] M. Cho and D. Z. Pan, "A high-performance droplet router for digital microfluidic biochips," in *Proc. ISPD*, Portland, OR, Apr. 2008, pp. 200–206.
- [28] P.-H. Yuh, C.-L. Yang, and Y.-W. Chang, "BioRoute: A network flow based routing algorithm for digital microfluidic biochips," in *Proc. IEEE Int. Conf. Comput.-Aided Des.*, 2007, pp. 752–757.
- [29] E. J. Griffith, S. Akella, and M. K. Goldberg, "Performance characterization of a reconfigurable planar-array digital microfluidic system," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 25, no. 2, pp. 345–357, Feb. 2006.
- [30] M. G. Pollack, "Electrowetting-based microactuation of droplets for digital microfluidics," Ph.D. dissertation, Duke Univ., Durham, NC, 2001.
- [31] J. C. Bischof and X. M. He, "Thermal stability of proteins," *Ann. N.Y. Acad. Sci.*, vol. 1066, pp. 12–33, 2005.
- [32] J. L. Gross and J. Yellen, *Graph Theory and Its Applications*. Boca Raton, FL: CRC Press, 1999.
- [33] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco, CA: Freeman, 1979.
- [34] P. Trinder, "Determination of glucose in blood using glucose oxidase with an alternative oxygen acceptor," *Ann. Clin. Biochem.*, vol. 6, pp. 24–27, 1969.



**Tao Xu** (S'07) received the B.E. degree in electrical engineering from Zhejiang University, Hangzhou, China, in 2005 and the M.S. degree in electrical and computer engineering from Duke University, Durham, NC, in 2007, where he is currently working toward the Ph.D. degree.

His research interests include design and testing of mixed-technology microsystems, electronic design automation, mixed-signal very large scale integration design, and MEMS modeling and simulation, particularly microfluidic biochips. He has published

16 papers in archival journals and refereed conference proceedings. He is a recipient of best paper award at the 2007 IEEE International Conference on VLSI Design.



**Krishnendu Chakrabarty** (S'92–M'95–SM'02–F'08) received the B.Tech. degree in computer science and engineering from Indian Institute of Technology, Kharagpur, India, in 1990 and the M.S.E. and Ph.D. degrees in computer science and engineering from the University of Michigan, Ann Arbor, MI, in 1992 and 1995, respectively.

He is currently a Professor of electrical and computer engineering with Duke University, Durham, NC.

Dr. Chakrabarty is a recipient of the National Science Foundation Early Faculty (CAREER) Award and the Office of Naval Research Young Investigator Award. His current research projects include the following: testing and design-for-testability of system-on-chip integrated circuits; digital microfluidic biochips; logic circuits based on DNA self-assembly; delay-tolerant wireless networks. He has authored five books—*Microelectrofluidic Systems: Modeling and Simulation* (CRC Press, 2002), *Test Resource Partitioning for System-on-a-Chip* (Kluwer, 2002), *Scalable Infrastructure for Distributed Sensor Networks* (Springer, 2005), *Digital Microfluidics Biochips: Synthesis, Testing, and Reconfiguration Techniques* (CRC Press, 2006), and *Adaptive Cooling of Integrated Circuits Using Digital Microfluidics* (Artech House, April 2007)—and edited the book volumes *SOC (System-on-a-Chip) Testing for Plug and Play Test Automation* (Kluwer, 2002) and *Design Automation Methods and Tools for Microfluidics-Based Biochips* (Springer, 2006). He has contributed over a dozen invited chapters to book volumes, published 280 papers in archival journals and refereed conference proceedings, and delivered over 110 keynote, plenary, and invited talks. He is the holder of a U.S. patent in built-in self-test and is a Coinventor of a pending U.S. patent on sensor networks. He is a recipient of best paper awards at the 2007 IEEE International Conference on VLSI Design, the 2005 IEEE International Conference on Computer Design, and the 2001 IEEE Design, Automation and Test in Europe (DATE) Conference. He is a recipient of Duke University's 2008 Dean's Award for Excellence in mentoring. He is also a recipient of the Humboldt Research Fellowship, awarded by the Alexander von Humboldt Foundation, Germany, and the Mercator Visiting Professorship, awarded by the Deutsche Forschungsgemeinschaft, Germany. He served as a Distinguished Visitor of the IEEE Computer Society for 2005–2007 and a Distinguished Lecturer of the IEEE Circuits and Systems Society for 2006–2007. He currently serves as an Association for Computing Machinery (ACM) Distinguished Speaker. He is an Associate Editor of IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, IEEE TRANSACTIONS ON VLSI SYSTEMS, IEEE TRANSACTIONS ON BIOMEDICAL CIRCUITS AND SYSTEMS, *ACM Journal on Emerging Technologies in Computing Systems*, an Editor of *IEEE Design and Test of Computers*, and an Editor of *Journal of Electronic Testing: Theory and Applications (JETTA)*. He served as Program Chair for the IEEE Asian Test Symposium in 2005 and the CAD, Design, and Test Conference for the 2007 IEEE Symposium on Design, Integration, Test, and Packaging of MEMS/MOEMS. He is a Distinguished Engineer of ACM and a member of Sigma Xi.