Control-Fluidic CoDesign for Paper-Based Digital Microfluidic Biochips

Qin Wang[1†], Zeyan Li[1†], Haena Cheong[2], Oh-Sun Kwon[2], Hailong Yao[1]
Tsung-Yi Ho[3,4], Kwanwoo Shin[2], Bing Li[5], Ulf Schlichtmann[5], and Yici Cai[1]
1. Department of Computer Science & Technology, Tsinghua University
2. Department of Chemistry & Institute of Biological Interfaces, Sogang University
3. Institute for Advanced Study, Technical University of Munich (TUM)   4. National Tsing Hua University
5. Institute for Electronic Design Automation, Technical University of Munich (TUM)
woodythu@163.com, zy-li14@mails.tsinghua.edu.cn, wjdgosk12@naver.com, oskwon@sogang.ac.kr, hailongyao@tsinghua.edu.cn
tyho@cs.nthu.edu.tw, kwshin@sogang.ac.kr, b.li@tum.de, ulf.schlichtmann@tum.de, caiyc@tsinghua.edu.cn

ABSTRACT

Paper-based digital microfluidic biochips (P-DMFBs) have recently emerged as a promising low-cost and fast-responsive platform for biochemical assays. In P-DMFBs, electrodes and control lines are printed on a piece of photo paper using inkjet printer and conductive ink of carbon nanotubes (CNTs). Compared with traditional digital microfluidic biochips (DMFBs), P-DMFBs enjoy notable advantages, such as faster in-place fabrication with printer and ink, lower costs, better disposability, etc. Because electrodes and CNT control lines are printed on the same side of a paper, a new design challenge for P-DMFB is to prevent the interference between moving droplets and the voltages on CNT control lines. These interactions may result in unexpected droplet movements and thus incorrect assay outputs.

To address the new challenges in automated design of P-DMFBs, this paper proposes the first control-fluidic codesign flow, which simultaneously adjusts the control line routing and fluidic droplet scheduling to achieve an optimized solution. As the control line routing may not be able to address all the interferences between moving droplets and the voltages on control lines, droplet rescheduling is performed to effectively deal with the remaining interferences in the routing solution. Computational simulation results on real-life bioassays show that the proposed codesign method successfully eliminates all the interferences, while a state-of-the-art maze routing method cannot solve any of the benchmarks without conflicts.

1. INTRODUCTION

Recently, paper-based digital microfluidic biochips (P-DMFBs) have emerged, with electrodes and control lines printed on a piece of photo paper using an inkjet printer and conductive ink consisting of carbon nanotubes (CNTs) [1]. As in traditional digital microfluidic biochips (DMFBs) [2], droplets can also be precisely manipulated in P-DMFBs by a 2-D array of CNT electrodes using the electrowetting technology [3]. Compared with DMFBs fabricated on solid substrates (e.g., silicon [4], glass [5], and polymers [6]), P-DMFBs printed on paper enjoy notable advantages, such as faster in-place fabrication with printer and ink, lower manufacturing cost, better disposability, etc.

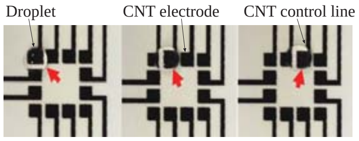


Figure 1: Paper-based digital microfluidic biochip [1].

Figure 1 shows an example of the paper-based digital microfluidic biochip [1], where CNT electrodes and control lines are printed on the paper. To enable the electrowetting technology, the CNT electrodes and control lines are coated with a hydrophobic Teflon film and a dielectric parylene-C film. Silicone oil is spin-coated on top of the P-DMFB as the lubricant to adjust the surface tension. Through external time-variant high/low control voltages enforced on the CNT control lines, the droplet is transported step-by-step across CNT electrodes. For example, for a rightward movement of the droplet in Figure 1, the electrode under the droplet needs a low voltage and the electrode in the right neighbor needs a high voltage. The driving voltage ranges from 70V to 120V [1].

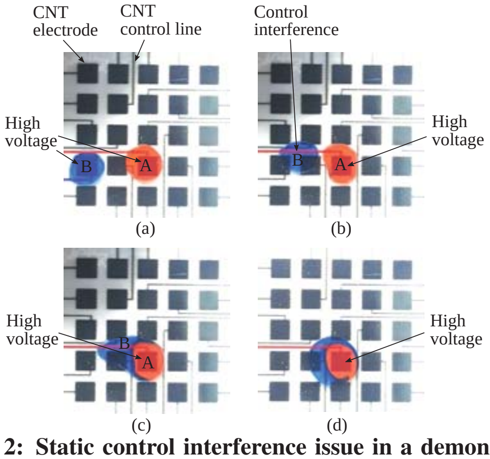


Figure 2: Static control interference issue in a demonstrative P-DMFB [7]: (a) high voltages are applied to both droplets A and B, (b) high voltage is only applied to A, (c) the high voltage on A drives B to mix with A, and (d) the final mixed droplet.

A new design challenge for P-DMFBs is the *control interference* between droplets and CNT control lines, which results in unexpected droplet movements and wrong assay outputs. The control interference on manufactured P-DMFBs has been demonstrated experimentally and documented in a video [7]. In the demonstrative P-DMFB in Figure 2, there are 5×5 CNT electrodes, the size of each electrode is $4.0mm^2$, the width of the CNT control line is $0.2mm$, the distance between adjacent electrodes is $2.0mm$, each droplet is $20mL$, and the high voltage for moving the droplets is 100V. A CCD camera was used to capture the

[†] Indicates equal student contribution.

ICCAD '16, November 07-10, 2016, Austin, TX, USA
© 2016 ACM. ISBN 978-1-4503-4466-1/16/11 . . . $15.00
DOI: http://dx.doi.org/10.1145/2966986.2967018

video, and Figure 2 shows images extracted from the recorded video. There are two types of control interferences according to whether the affected droplet is stalling or moving, i.e., *static* control interference for stalling droplets and *dynamic* control interference for moving droplets. Figure 2 shows the static control interference. In Figure 2(a), because of the high voltage on droplet B, the high-voltage control line in red for droplet A cannot move B. That is, the static control interference can be resolved by applying a high voltage on the stalling droplet. As shown in Figure 2(b), without the high voltage on the stalling blue droplet B, the high-voltage control line for A moves B. In Figure 2(c), due to the high-voltage red control line for A, the blue droplet B is unexpectedly mixed with A. The final mixed droplet due to the static control interference is shown in Figure 2(d).

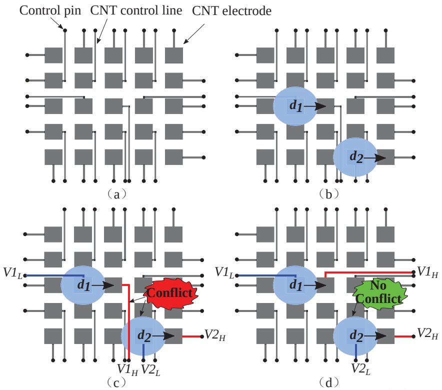


Figure 3: Dynamic control interference issue: (a) routed CNT control lines, (b) designated droplet movement, (c) dynamic control interference between droplet d_2 and high-voltage control line $V1_H$, and (d) rerouting the control line of $V1_H$ to resolve the conflict.

Figure 3 shows an illustrative example of the key design challenge caused by the dynamic control interference issue. Figure 3(a) shows the routed CNT control lines that connect electrodes to external control pins. In Figure 3(b), assume both droplets d_1 and d_2 are scheduled to move rightward at the same time. As shown in Figure 3(c), the corresponding electrodes need to be driven by low and high voltages, respectively. However, in Figure 3(c), the control line of $V1_H$ adversely affects droplet d_2's movement due to the voltage interference (i.e., $V1_H$ to the left of d_2 also has high voltage, blocking the move of d_2 to the right). This conflict is referred to as dynamic control interference, which causes malfunction to d_2. To solve the dynamic control interference, control lines need to be carefully planned. As shown in Figure 3(d), the control line of $V1_H$ is rerouted to resolve the conflict with droplet d_2.

In the past decade, noticeable advances have been made in automated design methods for DMFBs, which can be classified into two main categories [8]: (1) Fluidic-level synthesis: this design stage includes sequencing graph construction, operation scheduling and resource binding, module placement, and droplet routing [9–18]; (2) Chip-level design: this design stage includes electrodes' activation sequence generation, electrode addressing, and control line routing [19–24]. In contrast to conventional DMFBs, there is only one existing work on P-DMFBs [25]. However, the dynamic control interference is completely ignored in [25], which will result in the wrong outcome of the bio-assays and thus waste expensive reagents. Therefore, a codesign framework considering dynamic control interference with droplet re-scheduling is necessary to obtain the correct overall solution.

This paper proposes the first practical and effective codesign framework for P-DMFBs, which seamlessly integrates the control line routing and droplet scheduling stages, and iteratively adjusts each design stage from feedbacks of the other stage. Conflicts between the control lines and moving droplets are all effectively resolved in the proposed codesign method. Major contributions of the paper are as follows:

- The first codesign framework including control-line routing and droplet re-scheduling is proposed, which effectively resolves the interferences between routed control lines with high voltages and moving droplets, while minimizing the total length of control lines and the bioassay's total execution time.
- An effective escape routing method based on the minimum cost network flow formulation is employed, which concurrently routes the CNT electrodes to control pins with optimized total wire length and routing completion rate.
- An effective obstacle-avoiding routing method is proposed, which addresses a new routing requirement that different sets of routing obstacles need to be avoided for different CNT electrodes to be routed.
- A new A*-searching-based droplet re-scheduling method is proposed to further eliminate both static and dynamic control interferences after control line routing, which enhances the success rate of droplet scheduling process with the computed control lines.

The remainder of this paper is organized as follows. Section 2 presents the problem formulation. Section 3 presents the overall flow of the proposed codesign framework. Section 4 presents the droplet routing and scheduling method. Section 5 presents details of the control line routing algorithm based on the minimum cost flow formulation. Section 6 describes droplet re-scheduling method for the computed control line routing solution. Section 7 presents and discusses the experimental results. Finally, a conclusion is drawn in Section 8.

2. PROBLEM FORMULATION

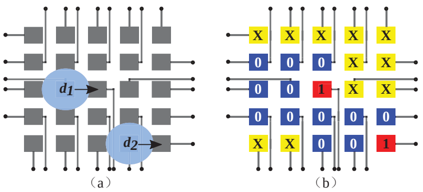


Figure 4: Example of electrode activation pattern: (a) designated droplet movement at the next time step, and (b) electrode activation pattern for the next time step.

Similar to DMFBs [8], fluidic-level synthesis is required in P-DMFBs to generate the droplets' routing paths on the paper. Moreover, scheduled droplet transportation is also generated, which indicates the droplets' positions at any time step. In other words, droplets are scheduled to make their movements according to the control signals. At each time step, a droplet may either stall at its current position or move to the next position along its routing path. To control the droplet movement, the underlying electrodes are activated by "0-1-X" control signals [19]. Here, "0" denotes a low voltage, "1" denotes a high voltage, and "X" denotes a "don't-care" voltage (i.e., the electrode can be driven by either high or low voltage without affecting the designated droplets' movement). Figure 4 shows an example of the electrode activation pattern for all electrodes. For the designated movements of droplets d_1 and d_2 in Figure 4(a), the electrode activation pattern for the next time step is shown in Figure 4(b). In Figure 4(b), for correct movement of droplet d_1, the eight electrodes surrounding d_1 should be activated by either "1" or "0". To move d_1 to the right, the electrode to the right should be "1" and the other surrounding electrodes should be "0". Moreover, to avoid the static and dynamic control interferences between droplets and high-voltage control lines, it is preferred to replace "X" with "0" in the activation pattern. According to the droplets' transportation requirements, a sequence of activation patterns is generated for all time steps.

As mentioned above, activation bit “X” is preferably replaced by “0” in P-DMFBs to avoid the static and dynamic control interferences between moving droplets and high-voltage control lines. If the electrode e_i is activated at the time step k, the control line of e_i has the voltage “1”, so that activation bit $a_{i,k}$=“1”. At the same time step, if there is a moving droplet on another electrode e_j which is neighboring the control line, the status of the droplet may be affected by the voltage on the control line of e_i. Furthermore, if a moving droplet appears on such a neighboring electrode e_j at the time step $k-1$ and should be moved away from e_j at the time step k, the moving operation may also be disturbed by the voltage on the control line. Considering these situations together, we define the conflict condition for the control line of the electrode e_i as

DEFINITION 1 *(Conflict electrodes). For an electrode e_i, its control line must avoid the electrodes in its* **conflict electrode** *set $C(e_i)$={e_j | there is a moving droplet on e_j at the time step k or $k-1$ and $a_{i,k}$=“1”, $\forall j \neq i$, $\forall k$}.*

According to the definition above, conflict electrodes $C(e_i)$ of e_i represents the set of electrodes that must not sit near the control line of e_i. That is, when routing the control line of e_i, the neighborhood of electrodes in $C(e_i)$ should be avoided.

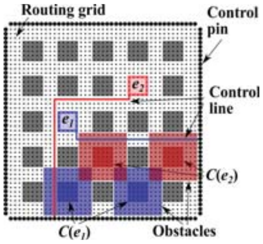


Figure 5: Routing obstacles in control line routing: routed control lines avoid specific obstacles from conflict electrodes, i.e., e_1 only needs to avoid obstacles from $C(e_1)$, and e_2 only needs to avoid obstacles from $C(e_2)$.

When the droplets’ movements have been scheduled, conflict electrodes can be determined for each electrode. Then the corresponding routing obstacles can be computed. Figure 5 shows an example of control line routing with obstacle avoidance. In Figure 5, the conflict electrodes $C(e_1)$ and $C(e_2)$ for electrodes e_1 and e_2 are computed, respectively. The shaded neighborhood of the conflict electrodes are set as routing obstacles. As the control line of e_1 only needs to avoid routing obstacles induced by $C(e_1)$, it can pass the obstacles induced by $C(e_2)$. Similarly, the control line of e_2 can pass the obstacles induced by $C(e_1)$. This is a specific feature for control line routing in P-DMFBs. In DMFBs, except for the electrodes, there are typically no other types of routing obstacles. Moreover, in DMFBs there are no specific routing obstacles only for one electrode as shown in Figure 5. This makes the control routing problem for P-DMFBs much harder to solve.

With the above definitions and descriptions, this paper addresses the following problem:

Control and Fluidic Co-Design for Paper-Based Digital Microfluidic Biochips.
Given: A set of droplet routing subproblems with droplets’ source and target positions for a given bioassay, and the routing blockages.
Find: The droplet routing and scheduling results, and control line routing paths connecting all electrodes to control pins.
Subject to: (i) Both static and dynamic fluidic constraints in DMFBs. (ii) Minimum line width and spacing design rules for control line routing. (iii) No crossing is allowed between different control lines. (iv) Both static and dynamic control interference constraints between the high-voltage control lines and droplets should be satisfied.
Objective: Minimize weighted sum of total wire length of control lines, and the bioassay’s execution time.

3. OVERALL FLOW

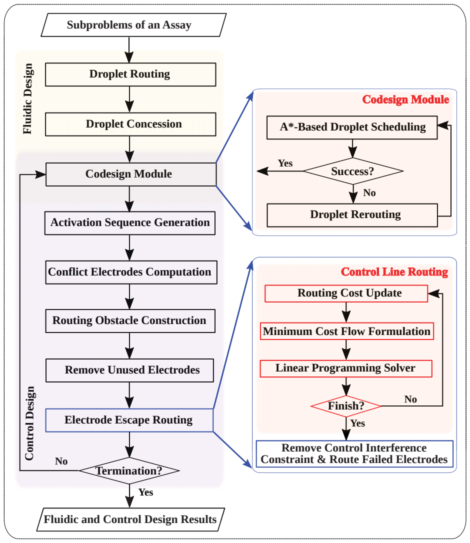

Subproblems of an Assay
Fluidic Design
Droplet Routing
Droplet Concession
Codesign Module
Activation Sequence Generation
Conflict Electrodes Computation
Routing Obstacle Construction
Remove Unused Electrodes
Electrode Escape Routing
Control Design
No
Termination?
Yes
Fluidic and Control Design Results
Codesign Module
A*-Based Droplet Scheduling
Yes
Success?
No
Droplet Rerouting
Control Line Routing
Routing Cost Update
Minimum Cost Flow Formulation
Linear Programming Solver
Finish?
No
Yes
Remove Control Interference Constraint & Route Failed Electrodes

Figure 6: Overall flow of the proposed codesign framework.

Figure 6 shows the overall flow of the proposed codesign framework. The input to the system are subproblems of an assay. The fluidic design module obtains the droplets’ routing and scheduling results. For a certain droplet routing result, it is possible that no valid scheduling solution exists. In that case, droplet rerouting is required to find a feasible droplet scheduling solution. Then, according to the scheduled droplets’ movements, droplets’ positions in each clock cycle are obtained. So the activation sequence on the underlying CNT electrodes can be computed. Next, the set of conflict electrodes with static and dynamic control interferences is computed for each electrode according to the droplets’ movements. The conflict electrodes along with their neighborhood area form specific routing obstacles for the corresponding electrode. Therefore, when routing an electrode to peripheral control pins, the routed control line should avoid a specific set of routing obstacles. Since each electrode has a different set of conflict electrodes, routing obstacles are constructed for each electrode individually. This brings new design challenges to routing methods in P-DMFBs. Please note that those electrodes without any droplets passing by are marked as unused, which do not need to be connected by control lines.

After routing obstacles are computed for each electrode, escape routing for the electrodes is performed to compute the CNT control lines. The minimum line width and spacing constraints need to be satisfied during routing. During the electrodes’ escape routing stage, several iterations will be conducted to adjust the control line routing solution for avoiding the static and dynamic control interferences. If there are still static and dynamic control interferences after a pre-specified number of iterations, the routing process will be finished by ignoring these conflicts. After the electrodes’ escape routing, the termination condition will be checked. If all the static and dynamic control interferences are resolved or the threshold on the number of codesign iterations is reached, the codesign flow will be terminated with final fluidic and control design results. Otherwise, the conflict situations between electrodes and the routed control lines will be analyzed for the codesign stage. During the codesign stage, droplets are re-scheduled according to the routed control lines. If the re-scheduling algorithm cannot find a feasible solution according to the control line routing result, droplet rerouting will be invoked. Then the whole design flow will be iterated once again. Details of the proposed method are described in the following sections. The notations used in this paper are summarized in Table 1.

Table 1: Notation table.

Notations	Meaning
R	Set of all the routing grids
r_i	The i^{th} routing grid
R_g	Set of general routing grids
R_z	Set of routing grids corresponding to electrode e_z
R_c	Set of routing grids corresponding to the control pins
E	Set of all the electrodes
C	Set of conflict electrodes for each electrode in E
P	Set of all the available control pins
$f_{i,j}$	$(0 \leq f_{i,j} \leq 1)$ Floating variable for flow from routing grid r_i to r_j
$s_{i,j}$	Constant value denoting cost of unit flow corresponding to $f_{i,j}$
S	Matrix of constant cost values for unit flows

4. FLUIDIC DESIGN

4.1 Droplet Routing

The first step in fluidic design is droplet routing. The target of droplet routing is to connect the list of nets in each subproblem. Moreover, the droplets' paths must satisfy both the fluidic constraints and the timing constraint.

Fluidic constraint: During droplet movement, the spacing between different droplets must be large enough to avoid unexpected mixing errors. Fluidic constraints can be further divided into the *static* and *dynamic* constraints [26]. Assume d_i at (x_i^t, y_i^t) and d_j at (x_j^t, y_j^t) are two independent droplets at time t. Then the following constraints must be satisfied for any t during droplet transportation:

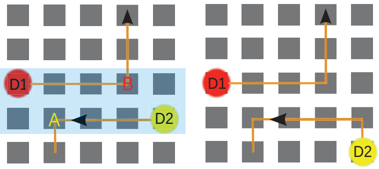
1. Static constraint: $|x_i^t - x_j^t| > 1$ or $|y_i^t - y_j^t| > 1$.
2. Dynamic constraint: $|x_i^{t+1} - x_j^t| > 1$ or $|y_i^{t+1} - y_j^t| > 1$ or $|x_i^t - x_j^{t+1}| > 1$ or $|y_i^t - y_j^{t+1}| > 1$.

The static constraint states that the minimum spacing between two droplets is one electrode for any time step t during droplet movement. The dynamic constraint states that the activated high-voltage electrode for d_i (d_j) cannot be adjacent to droplet d_j (d_i) because there may be more than one activated neighboring electrodes for d_j (d_i). Therefore, without satisfying the dynamic constraint, we may have an unexpected mixing between droplets d_i and d_j.

Timing constraint: This constraint gives an upper limit on the transportation time of droplets along their paths. The timing constraint is used to ensure the execution time of an assay. Those paths that violate the timing constraint will be pruned away during droplet routing.

In this work, the droplet routing method is based on the classic A* search algorithm. In order to obtain a promising solution, we introduce an additional variable *used* to record whether the current cell has been used by other droplets' paths. By setting the surrounding cells of finished routes as *used* with a higher routing cost, the fluidic constraints can be satisfied in most cases. As a result, it will become easier for the droplet scheduling process to find a valid solution.

4.2 Droplet Concession



(a) Droplet Deadlock (b) Droplet Concession

Figure 7: Example of droplet deadlock and droplet concession.

After droplet routing, we need to detect the deadlocks for droplet scheduling caused by unreasonable routing results. As shown in Figure 7(a), droplets $D1$ and $D2$ have disjoint routing paths. However, there is no valid scheduling solution for the two droplets. This is caused by the violations to fluidic constraints in the blue region in Figure 7(a). Even in extreme cases where we schedule droplets one after another, there is still no feasible solution. For example, if $D1$ moves first, it will violate the fluidic constraint with $D2$ at position B. But if $D2$ moves first, it will violate the fluidic constraint with $D1$ at position A.

The droplet scheduling deadlocks can be resolved by droplet rerouting. However, the whole rerouting process may not converge quickly. Therefore, we adopt the droplet concession strategy to solve this problem [15]. As shown in Figure 7(b), when a droplet scheduling deadlock occurs, one of the droplets (say $D2$) will first be moved to an empty position for concession. Then the other droplet ($D1$) can be moved to break the tie. After that, the compromised droplet will be moved back to its source position for the planned transportation. The droplet concession steps can be incorporated in the droplet scheduling method, and will be recorded in the scheduling solution. Please note that droplet rerouting will still be invoked if droplet concession cannot successfully resolve the deadlocks. In extreme cases, even prior design stages, such as placement, need to be refined to obtain feasible droplet routing and scheduling solutions [28].

4.3 Droplet Scheduling

The droplet scheduling stage obtains a scheduling solution for the movement of each droplet. During the movements of droplets, unexpected droplet mixing must be avoided and the timing constraint has to be satisfied. Moreover, in order to speed up the assay execution, the total execution time should be minimized. Existing heuristic scheduling methods (e.g., [27]) may lead to deadlocks in special cases and cannot guarantee optimal scheduling solution. To address these problems, we proposed an A*-searching-based droplet scheduling method, which is able to obtain the optimal solution. The proposed A*-searching-based droplet scheduling method are used in two scenarios: (1) directly after droplet routing without control line routing solutions; and (2) during the codesign stage with the control line routing solution and thus with more scheduling constraints for eliminating the static and dynamic control interferences. Details of the proposed scheduling algorithm are presented in Section 6.2.

5. CONTROL DESIGN

During control line routing, the control lines from electrodes to control pins are computed, which avoid the obstacles induced by the conflict electrodes. There are two major objectives: (1) to minimize the total length of control lines for reducing the cost of CNT ink material, and (2) to maximize the number of successfully routed control lines if all electrodes are not simultaneously routable. Without considering the specific routing obstacles for avoiding static and dynamic control interferences, the electrode escape routing problem can be solved optimally using the minimum cost network flow formulation.

5.1 Minimum Cost Flow Formulation

Inspired by [29], we propose the minimum cost flow formulation for escape routing and obtain the solutions using a linear programming solver. According to the minimum width (w_{min}) and minimum spacing (s_{min}) design rules, the routing area can be partitioned into a mesh by horizontal and vertical lines with uniform spacing (see Figure 8(a) for an example). By setting the spacing value as $w_{min} + s_{min}$, no design rule violations will occur when routing is performed along these horizontal and vertical lines. *Routing grids* are the intersection points between these horizontal and vertical lines. A network flow formulation can be constructed on this mesh, where an *ingoing/outgoing flow* of a routing grid refers to the flow going into or out of the node corresponding to the routing grid. The network flow graph $G_F = (V_F, E_F)$ (see Figure 8(b)) and the corresponding minimum cost flow problem are constructed as follows:

- A *super source node s* and *super target node t* are added into V_F, with capacity ∞ and unit flow cost 0.

- For each control pin $p_j \in P$, a target node t_j is added into V_F, with capacity 1 and unit flow cost 0.
- For each electrode $e_i \in E$, a source node s_i is added into V_F, with capacity 1 and unit flow cost 0.
- For each general routing grid $r_i \in R_g$ that is not covered by electrodes or control pins, a node is added into V_F, with capacity 1 and unit flow cost 0.
- A directed edge is added into E_F for each (s, s_i), (t_j, t), (r_i, t_j), and (s_i, r_i) according to the connectivity in the mesh, with capacity 1. The unit flow cost of (s, s_i) is a negative constant value (e.g., -α), and the unit flow cost of other edges is 0.
- A bi-directional edge is added into E_F between each pair of general adjacent routing grids according to the mesh, with capacity 1 and unit flow cost 1.

Since the capacities and demands of the nodes are all integers, according to the integrality property [30], the above network flow problem has an integer minimum cost flow. Therefore, the above minimum cost network flow problem can be formulated as the following linear programming problem:

- **Objective:** Minimize $\sum s_{i,j} \cdot f_{i,j} - \alpha \cdot \sum x_c$
- **Subject to:**

$$\sum_{g_j \in R_z} \sum_k f_{j,k} \geq x_c \qquad \forall\, e_z \in E \qquad (1)$$

$$\sum_{g_j \in R_z} \sum_k f_{i,j} = 0 \qquad \forall\, e_z \in E \qquad (2)$$

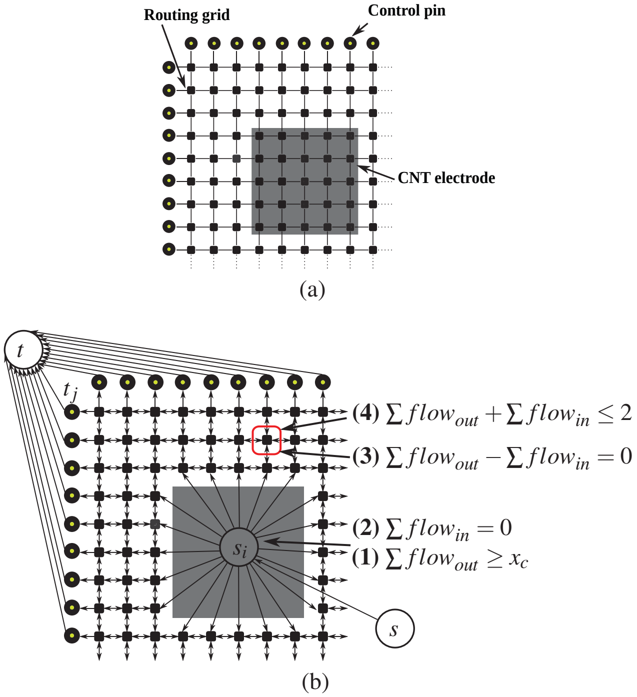
$$\sum f_{j,k} - \sum f_{i,j} = 0 \qquad \forall\, r_j \in R_g \qquad (3)$$

$$\sum f_{j,k} + \sum f_{i,j} \leq 2 \qquad \forall\, g_j \in R_g \qquad (4)$$

where x_c ($0 \leq x_c \leq 1$) is a floating variable for maximizing the number of successfully routed paths, α is a user-defined parameter to make ($\sum x_c$) dominate ($\sum s_{i,j} \cdot f_{i,j}$), and the other notations are explained in Table 1. Constraint (1) computes the sum of all outgoing flows from the routing grids of an electrode, and sets x_c as the lower bound. Therefore, by maximizing $\sum x_c$ the total number of successfully routed control lines is maximized. Constraint (2) states that all ingoing flows are 0 for all the routing grids of electrode e_z. Constraint (3) states the flow conservation constraint for all the general routing grids. Constraint (4) not only avoids crossings between control lines, but also enforces flow capacity constraints on the flow edges. General routing grids exist from electrodes to control pins. This guarantees the validity of the flow capacity constraint. In the above problem formulation, general routing obstacles are easily incorporated, i.e., by removing the corresponding routing grids from the network flow graph, or by enforcing the corresponding ingoing and outgoing flows to 0. The above problem can easily be solved by a linear programming solver [31]. Figure 8(b) demonstrates the linear programming constraints on the network flow graph. As explained above, directed flow edges are drawn between the routing grids. The capacity of the flow edges is 1 (i.e., $0 \leq f_{i,j} \leq 1$).

5.2 Electrode Escape Routing

As mentioned in Section 5.1, general routing obstacles are easily considered in the minimum cost flow formulation. However, specific routing obstacles for each electrode caused by its conflict electrodes are difficult to handle. Algorithm 1 presents the electrode escape routing algorithm, which addresses the specific routing obstacles for each electrode to be routed. The basic idea of the algorithm is to iteratively route the control lines using minimum cost network flow formulation, and rip-up and reroute those illegal control lines in subsequent loops. Updated edge costs are used to encourage failed electrodes to choose alternative routing solutions. Moreover, the routed control lines passing through the neighborhood of failed electrodes are ripped-up and rerouted using the same network flow formulation for enhanced overall routability.


Routing grid
Control pin
CNT electrode
t
t_j
s_i
s
(4) $\sum flow_{out} + \sum flow_{in} \leq 2$
(3) $\sum flow_{out} - \sum flow_{in} = 0$
(2) $\sum flow_{in} = 0$
(1) $\sum flow_{out} \geq x_c$

(a)

(b)

Figure 8: Network flow formulation: (a) top-left corner of the P-DMFB with one CNT electrode, and (b) the corresponding network flow graph.

In Algorithm 1, specific routing obstacles for each electrode are first constructed. Then elements in the cost matrix for unit flows in the minimum cost network flow formulation are initialized as 1.0. E_0 is initialized to hold the electrodes to be routed. In each loop, routed control lines are treated as general routing obstacles O_b. Then a while-loop is entered, where electrodes are routed by the minimum cost network flow formulation and linear programming solver. If computed control line l_i of electrode e_i passes the obstacles induced by its corresponding conflict electrodes $C(e_i)$, then l_i will be ripped-up and rerouted in the next loop. Moreover, the routing costs are increased for those edges in the next loop. To further improve the routability, routed control lines are ripped-up and rerouted if they are within Manhattan distance τ from the electrodes to be routed. The algorithm will be terminated when all electrodes are successfully routed or the unit-flow cost ξ is large enough ($> \gamma$) to allow extensive solution space exploration with large routing detours. Assume an electrode takes $w_E \times w_E$ grids, and spacing between electrodes takes s_E grids. And assume the size of the paper chip is $w \times h$. In the experiments, $\beta = w_E + s_E$, $\gamma = max\{w,h\}$, and $\tau = s_E$. Time complexity of Algorithm 1 is dominated by the linear programming solver [31], which runs in polynomial time in most cases.

6. CODESIGN METHOD

6.1 Control Interference Report

Even after several iterations of control line rerouting, there may still exist static and dynamic control interferences. The static and dynamic control interference constraints will be removed at the last iteration to ensure all the control lines can be routed. After the control line routing stage, all the control lines are fixed. Given the routed control lines, the remaining static and dynamic control interferences will be resolved by the droplet re-scheduling method.

6.2 Droplet Re-Scheduling With Control Line Routing

The droplet re-scheduling module is the major part of codesign module. In our droplet re-scheduling method, the above interference constraint along with fluidic and timing constraints are

```
Input: Electrodes $E$, conflict electrodes $C$, control pins $P$, and routing grids $R$.
Output: Routed control lines $L$ from $E$ to $P$.
Construct routing obstacles $O(e)$ from $C(e)$ for each electrode $e_i \in E$ (see Figure 5);
Initialize cost matrix $S_0 \leftarrow \{1.0\}$, cost value $\xi \leftarrow 1.0$, electrodes $E_0 \leftarrow E$, and routing obstacles $O_b \leftarrow \phi$;
Set counter $r \leftarrow 0$, $flag \leftarrow false$;
while $|E_0| \neq 0$ do
    Construct a network flow problem with $E_0$ according to Figure 8;
    Build a minimum cost network flow problem with $O_b$ and $S_r$;
    Solve the network flow problem by a linear programming solver;
    Construct the control lines $L_r$ computed by linear programming;
    Set $\xi \leftarrow \xi + \beta$;
    for $i = 1$ to $|L_r|$ do
        Obtain electrode $e_i$ from control line $l_i$;
        if $l_i$ passes routing grids in $O(e_i)$ then
            Rip-up $l_i$;
            if $\xi > \gamma$ then
                Add routing grid of $O(e_i)$ into $O_b$;
            else
                Update $S_r$ for edges passing any grid in $O(e_i)$: $s_{j,k} \leftarrow \xi$;
        else
            Insert $l_i$ into $L$, and remove $e_i$ from $E_0$;
            Add routing grids of $l_i$ into $O_b$;
    if $flag = true$ then break;
    if $\xi > \gamma$ then Set $flag \leftarrow true$;
    for $i = 1$ to $|E_0|$ do
        for $j = 1$ to $|L|$ do
            if $dis(e_i, l_j) < \tau$ then
                Rip-up control line $l_j$ and insert $e_j$ into $E_0$;
    Set counter $r \leftarrow r + 1$;
if $|E_0| \neq 0$ then Report failed electrodes in $E_0$.
```

Algorithm 1: Electrode escape routing algorithm.

all considered. During the control line routing, it may not be possible to avoid all the interferences between high-voltage control lines and moving droplets. Therefore, we propose to re-schedule the transportation of droplets according to the control line routing solution. The re-scheduling step effectively reduces the remaining conflicts. During the droplet re-scheduling process, both the static and dynamic fluidic constraints are satisfied. Moreover, the control logic for actuating the electrodes are considered to avoid the static and dynamic control interferences.

The key idea of our droplet re-scheduling method is summarized as follows. For any time step t, we define a *state* s_t, which denotes the temporary droplet scheduling result of all the droplets, including the current positions of the droplets at t. Then at the next time step $t+1$, each droplet has the following three choices of movement: (1) move forward by one electrode along its routing path (denoted by moving 1 step), (2) stall at its current position without movement (denoted by moving 0 step), and (3) move backward by one electrode along its routing path (denoted by moving -1 step). From state s_t to s_{t+1}, at least one droplet moves with a non-zero step, i.e., at least one droplet moves forward/backward by one electrode along its path. In the initial state s_0, all the droplets are on their source positions, and the *time cost* TC of s_0 is 0. At each of the following time steps, the time cost is increased by 1, i.e., $TC(s_{t+1}) = TC(s_t) + 1$. In the final state s_t of droplet scheduling, all the droplets are on their target positions. The goal of the droplet scheduling algorithm is to find a sequence of state transformations from s_0 to s_t with no fluidic violations as well as minimized time cost.

Algorithm 2 and Algorithm 3 present details of the proposed A*-searching-based droplet scheduling method, which can be used either with or without the control line routing solution. The only difference is in the neighbor state searching step. In the proposed A*-searching algorithm, the g, h, and f values of a state s are defined as follows: (1) $g(s)$: the *time cost* of state s, for which the droplets are scheduled from their source positions to current positions in s; (2) $h(s)$: the sum of all the droplets' distances from their current positions to the target positions in s; (3) $f(s) = g(s) + h(s)$. As defined above, the time cost refers to the number of time steps used in droplet scheduling. The time complexity of A*-based searching algorithm is $O(b^d)$, where b is the branching factor (i.e., the average number of successors of each state), and d is the depth of the solution. Please note that the above definition of $h(s)$ does not guarantee the optimal solution because it is not strictly less than the optimal $h^*(s)$. However, a scheduling result with acceptable solution quality can be efficiently obtained. As shown in Section 7, the proposed method runs much faster than the optimal A* search method, and we verify that the obtained solution is fairly close to the optimal one.

```
Input: Droplets' routing paths $R$ and conflict electrodes $C$.
Output: Scheduled droplets' paths $R_S$.
Initialize start state $s_0$ with each droplet at its source position;
Set the cost of $s_0$ as $g(s_0) \leftarrow 0$ which denotes the overall execution time;
Push $s_0$ into set $Open$;
Initialize an empty set $Closed \leftarrow \phi$;
while $|Open| \neq 0$ do
    Pop out the state $s_t$ with minimum cost $f$ in $Open$;
    if $h(s_t) = 0$ then
        Construct scheduled paths $R_S$ by traversing from $s_0$ to $s_t$;
        Report Success;
    if $s_t \in Closed$ then
        continue;
    else
        Add $s_t$ into $Closed$;
    Call $S = Neighbor(0, \phi, s_t)$ to find all valid neighbors $S$ of $s_t$ avoiding the conflict electrodes in $C$;
    foreach $s \in S$ do
        if $\exists s_k \in Closed$ with the same droplet positions as in $s$ then
            if $g(s_k) > g(s)$ then
                Replace $s_k$ with $s$;
            else
                continue;
        else
            Add $s$ into $Open$;
Report Failed.
```

Algorithm 2: A*-searching-based droplet scheduling algorithm.

```
Input: Set of droplets $D$, droplet No. $n$ for searching, conflict electrodes $W$ of
       each electrode, conflict electrodes $C$ of each droplet, and current state $s$.
Output: Set of new states $S$
if $n = 0$ then
    foreach droplet $d \in D$ do
        foreach droplet $d_i \neq d$ and $d_i$ is not planned to merge with $d$ do
            Add the eight neighboring electrodes around $d_i$'s position to $C(d_i)$;
foreach droplet $d \in D$ do
    if $d$ is on a conflicting electrode of $C(d)$ then
        return;
if $n \geq |D|$ then
    Add $s$ to $S$;
    return;
for $k \in \{-1, 0, 1\}$ do
    Set state $s_c \leftarrow s$;
    Update $s_c$ by moving droplet $n \in s_c$ by $k$ steps along its path;
    Let $p$ denote droplet $n$'s position in $s_c$;
    Set $f(s_c) \leftarrow f(s_c) + 1$;
    Set $C_c \leftarrow C$. if $k \neq 0$ then
        foreach droplet $d \in D$ that is not the same as droplet $n$ do
            Add $W(p)$ into $C_c$;
    foreach drplet $d$ that is not planned to merge with droplet $n$ do
        Add the eight neighboring electrodes around $p$ into $C_c(d)$;
    Call $Neighbor(n+1, C_c, s_c)$;
```

Algorithm 3: Neighbor state searching algorithm.

6.3 Control Line Rerouting

During droplet re-scheduling for a given control line routing solution, there may still be static and dynamic control interferences that cannot be successfully addressed. In such cases, ripping-up and rerouting certain control lines may be effective in resolving this conflict. The basic idea is as follows. Based on the re-scheduled droplet movements, we find those critical control lines

Table 2: Design parameters.

Design	Size	#Electrodes	Routing area	#CP	#SUB
in-vitro_1	16×16	256	131×131	516	11
in-vitro_2	14×14	196	115×115	452	15
protein_1	21×21	441	171×171	676	64
protein_2	13×13	169	107×107	420	78
protein_2A	13×13	169	107×107	420	40
protein_2B	13×13	169	107×107	420	38
random_1	21×21	441	171×171	676	8
random_2	20×19	380	163×155	628	21
random_3	29×15	435	235×123	708	11
random_4	21×12	252	171×99	532	6
radnom_5	17×16	256	139×131	532	9

that cause the static and dynamic control interferences, and then try to rip-up and reroute these control lines without causing new static or dynamic control interferences. If rip-up and rerouting of control lines cannot resolve the conflicts, another round of codesign iteration will be performed based on the new droplet scheduling results, and new control line routing solutions will be obtained to avoid the conflicts.

7. EXPERIMENTAL RESULTS

We have implemented our P-DMFB codesign system in C++ programming language. Our system is tested on a 2.62GHz Intel Xeon Linux server with 32 cores and 132GB memory. The Gurobi optimizer is used to solve the linear programming problem [31]. Table 2 shows the details of the benchmarks, where "Design" gives the names of the benchmarks including both real bioassays and synthesized testcases. "Size" gives the sizes of the CNT electrode array, "#Electrodes" gives the number of electrodes to be routed, "Routing area" gives the total number of routing grids, "#CP" gives the number of candidate control pins and "#SUB" denotes the number of subproblems in an assay. In the experiments, each electrode takes 5×5 routing grids ($w_E = 5$), and the spacing between adjacent electrodes is set to be 3 routing grids ($s_E = 3$) in Table 3. This setting is very strict considering the resolution of current office printers and the conductivity of CNT control lines. Relaxed spacing between adjacent CNT electrodes will improve overall routability. However, the design may not be applicable in current printing technology for P-DMFBs, because office printers can not print such thin lines.

Table 3 shows the experimental results of control line routing, where "#E_u" denotes the number of used electrodes for an assay, the column under "#E_r" gives the total number of routed electrodes, *Rate* gives the routing completion rate in percentage value, "Imp." gives the improvement ratio, and "WL" gives the total length of routed control lines. To verify the effectiveness of our control line method, we have implemented the maze routing algorithm. During maze routing for an electrode, the specific routing obstacles induced by its conflict electrodes are avoided. From the results, our proposed control line routing method shows a much better performance on routing completion rate, with on average 35.8% improvement compared with the maze routing algorithm. Meanwhile, all of the benchmarks are routed in 5 minutes. Experimental results show our control line routing method is effective.

As Table 3 shows, there are some failed electrodes due to the static and dynamic control interference constraints. Therefore, we remove all the control interference constraints to finish control line routing for all the electrodes. Then we solve the control interference issue during droplet re-scheduling. In order to reduce the runtime of the most complex benchmark "protein_2", it is divided into "protein_2A" and "protein_2B". Please note that the accuracy of an assay is not influenced by dividing, because all of the subproblems in an assay are independent. Table 4 presents the experimental results of our codesign flow, where "#Conflicts" gives the number of electrodes with conflict constraint violations, "#G" gives the number of routing grids between adjacent electrodes, and "Re-scheduling" represents the number of failed subproblems after droplet rerouting and re-scheduling. When we remove the static and dynamic control interference constraints in the last iteration of control line rip-up and rerouting, the routing completion rates are all 100% in our control line routing method. However, maze routing still has failed routes, resulting in failed designs. Furthermore, droplet rerouting is not necessary if no scheduling failure occurs, and the whole design flow will be finished. Please note that not all of the routing completion rates of "Maze" in Table 4 are better than the results in Table 3, because the path order decision in the maze based routing method is based on random strategy. Figure 9 indicates that our codesign flow can solve the static and dynamic control interference problem effectively and finish all of the benchmarks successfully without any fluidic or control interference violations. Moreover, our control line routing method obtains average 45.1% improvement compared with the maze routing algorithm.

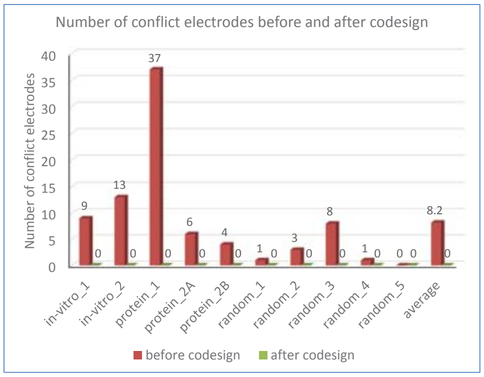


Figure 9: Number of conflict electrodes before and after codesign.

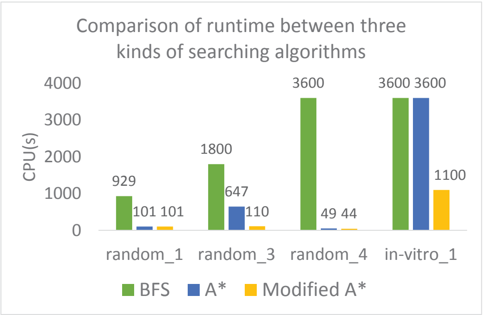


Figure 10: Runtime of three different types of searching algorithms for droplet scheduling.

Figure 10 presents the comparison between three different types of searching algorithms for droplet scheduling. The runtime values of BFS (Breadth-First-Search) with branch and cut strategy and A* search algorithm with optimal $h*$ are not acceptable for benchmark "in-vitro_1" and "random_4". In order to find a feasible solution in acceptable runtime, we propose to modify the h value in the A*-searching-based droplet scheduling method, so that close-to-optimal solutions can be obtained in greatly reduced runtime.

8. CONCLUSION

We have proposed the first practical control-fluidic codesign flow for the emerging paper-based digital microfluidic biochips. The codesign flow effectively and efficiently addresses the new design challenge of avoiding static and dynamic control interferences between droplets and the voltages on CNT control lines. Experimental results show that all CNT electrodes are successfully routed to control pins with minimized total wire length. Moreover, all the routing constraints are satisfied without any violations between droplets and control lines after the droplet re-scheduling stage, confirming the effectiveness of our codesign flow.

Table 3: Experimental results of control line routing.

Design	#E$_u$	#SUB	#E$_r$		*Rate* (%)			WL		CPU(s)	
			Maze	Ours	Maze	Ours	Imp.(%)	Maze	Ours	Maze	Ours
in-vitro_1	183	11	123	182	67.2	99.5	50.5	3196	5273	1.6	108.2
in-vitro_2	154	15	109	146	70.8	94.8	37.8	2499	3453	1.7	84.4
protein_1	402	64	193	265	48	65.9	82.6	6895	5862	10.6	280
protein_2	165	78	111	151	67.2	91.5	40.5	2330	2954	4.5	77.4
random_1	217	8	163	217	75.1	100	33.2	5738	8262	1.9	109
random_2	263	21	163	239	62	90.9	59.6	5508	8114	3.3	120
random_3	271	11	185	268	68.3	98.9	36.8	6011	9798	3.8	272
random_4	137	6	110	136	80.3	99.3	28.3	2804	3482	0.9	116.8
random_5	112	9	90	112	80.4	100	21.8	2714	3427	0.4	17.5
Avg.	-	-	-	-	68.8	93.4	35.8	-	-	3.2	114.8

Table 4: Experimental results of our codesign framework.

Design	#G	#E$_r$		*Rate* (%)			WL		#Conflicts		Re-scheduling		CPU(s)	
		Maze	Ours	Maze	Ours	Imp.(%)	Maze	Ours	Maze	Ours	Maze	Ours	Maze	Ours
in-vitro_1	3	124	183	67.8	100	47.5	3040	5645	9	9	-	0	1.4	1060
in-vitro_2	3	106	154	68.8	100	45.3	2312	3955	19	13	-	0	1.7	86.8
protein_1	8	188	402	46.8	100	71.5	10108	20501	28	37	-	0	14.5	10134.9
protein_2A	5	116	163	71.2	100	40.4	2470	3825	7	6	-	0	6.3	83.2
protein_2B	4	114	158	72.2	100	38.5	2431	3698	4	4	-	0	4	99
random_1	3	162	217	74.7	100	33.9	5591	8216	1	1	-	0	1.8	76.5
random_2	4	161	263	61.2	100	64.5	6041	10892	9	3	-	0	3.3	146.5
random_3	3	182	271	67.2	100	48.8	5384	9736	4	8	-	0	3.7	108.8
random_4	3	108	137	78.8	100	26.9	2608	3520	3	1	-	0	0.8	38
random_5	3	90	112	80.4	100	24.4	2714	3427	2	0	-	0	0.4	17
Avg.	-	-	-	68.9	100	45.1	-	-	-	-	-	-	-	-

9. ACKNOWLEDGEMENTS

The work of H. Yao was supported by Tsinghua University Initiative Scientific Research Program (20141081203). The work of T. Ho was supported in part by the Ministry of Science and Technology of Taiwan, under Grant MOST 105-2221-E-007-118-MY3 and 104-2220-E-007-021 and in part by the Technical University of Munich-Institute for Advanced Study, funded by the German Excellence Initiative and the European Union Seventh Framework Program under grant agreement no 291763. The work of K. Shin was supported by Leading Foreign Research Institute Recruitment Program(2013K1A4A3055268) and Mid-career Researcher Program (2016R1A2B3015239) funded by the Ministry of Science, ICT & Future Planning of Korea. The work of B. Li and U. Schlichtmann was supported by the IGSSE Project FLUIDA of Technical University of Munich (TUM). The work of Y. Cai was supported by the National Natural Science Foundation of China (61274031).

10. REFERENCES

[1] H. Ko, J. Lee, Y. Kim, B. Lee, C.-H. Jung, J.-H. Choi, O.-S. Kwon, and K. Shin, "Active Digital Microfluidic Paper Chips with Inkjet-Printed Patterned Electrodes", *Adv. Mater.*, 2014, vol. 26, pp. 2335-2340.

[2] V. Srinivasan, V. K. Pamula, P. Paik, and R. B. Fair, "Protein Stamping for MALDI Mass Spectrometry Using an Electrowetting-based Microfluidic Platform", *Optics East*, 2004, vol. 5591, pp. 26-32.

[3] M. G. Pollack, R. B. Fair, and A. D. Shenderov, "Electrowetting-Based Actuation of Liquid Droplets for Microfluidic Applications", *Appl. Phys. Lett.*, 2000, vol. 77, pp. 1725.

[4] H. Moon, S. K. Cho, R. L. Garrell, and C.-J. Kim, "Low Voltage Electrowetting-on-Dielectric", *J. Appl. Phys.*, 2002, vol. 92, pp. 4080-4087.

[5] U.-C. Yi and C.-J. Kim, "Characterization of Electrowetting Actuation on Addressable Single-Side Coplanar Electrodes", *J. Micromech. Microeng.*, 2006, vol. 16, pp. 2053-2059.

[6] M. Abdelgawad, S. Ferire, H. Yang, and A. R. Wheeler, "All-Terrain Droplet Actuation", *Lab Chip*, 2008, vol. 8, pp. 672-677.

[7] https://drive.google.com/open?id=0B5FxNLYyCf41SmRfeDZLUHR3em8

[8] T.-Y. Ho, K. Chakrabarty, and P. Pop, "Digital Microfluidic Biochips: Recent Research and Emerging Challenges," *Proc. of International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS)*, 2011, pp. 335-343.

[9] F. SU and K. Chakrabarty, "Architectural-Level Synthesis of Digital Microfluidics-Based Biochips," *Proc. ICCAD*, 2004, pp. 223-228.

[10] F. SU and K. Chakrabarty, "Unified High-Level Synthesis and Module Placement for Defect-Tolerant Microfluidic Biochips," *Proc. DAC*, 2005, pp. 825-830.

[11] D. Grissom, K. O'Neal, B. Preciado, et al., "A Digital Microfluidic Biochip Synthesis Framework," *Proc. VLSI-SoC*, 2012, pp. 177-182.

[12] P.-H. Yuh, C.-L. Yang, and Y.-W. Chang, "Placement of Defect-Tolerant Digital Microfluidic Biochips Using the T-Tree Formulation," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 2007, vol. 3, no. 3, Artical No. 13.

[13] F. SU, W. Hwang, and K. Chakrabarty, "Droplet Routing in the Synthesis of Digital Microfluidic Biochips," *Proc. DATE*, 2006, vol. 1, pp. 1-6.

[14] T. Xu and K. Chakrabarty, "Integrated Droplet Routing in the Synthesis of Microfluidic Biochips," *Proc. DAC*, 2007, pp. 948-953.

[15] M. Cho and D. Z. Pan, "A High-Performance Droplet Routing Algorithm for Digital Microfluidic Biochips," *IEEE Trans. on CAD*, 2008, vol. 27, no. 10, pp. 1714-1724.

[16] P.-H. Yuh, C.-L. Yang, and Y.-W. Chang, "BioRoute: A Network-Flow-Based Routing Algorithm for the Synthesis of Digital Microfluidic Biochips," *IEEE Trans. on CAD*, 2008, vol. 27, no. 11, pp. 1928-1941.

[17] T.-W. Huang and T.-Y. Ho, "A Two-Stage Integer Linear Programming-Based Droplet Routing Algorithm for Pin-Constrained Digital Microfluidic Biochips," *IEEE Trans. on CAD*, 2011, vol. 30, no. 2, pp. 215-228.

[18] Z. Xiao and E. F. Y. Young, "CrossRouter: A Droplet Router for Cross-Referencing Digital Microfluidic Biochips," *Proc. ASP-DAC*, 2010, pp. 269-274.

[19] T. Xu and K. Chakrabarty, "Broadcast Electrode-Addressing for Pin-Constrained Multi-Functional Digital Microfluidic Biochips," *Proc. DAC*, 2008, pp. 173-178.

[20] T.-W. Huang, S.-Y. Yeh, and T.-Y. Ho, "A Network-Flow Based Pin-Count Aware Routing Algorithm for Broadcast-Addressing EWOD Chips," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2011, vol. 30, no. 12, pp. 1786-1799.

[21] T.-W. Huang, T.-Y. Ho, and K. Chakrabarty, "Reliability-Oriented Broadcast Electrode-Addressing for Pin-Constrained Digital Microfluidic Biochips," *Proc. ICCAD*, 2011, pp. 448-455.

[22] S.-H. Yeh, J.-W. Chang, T.-W. Huang, and T.-Y. Ho, "Voltage-Aware Chip-Level Design for Reliability-Driven Pin-Constrained EWOD Chips," *Proc. ICCAD*, 2012, pp. 353-360.

[23] S. S.-Y. Liu, C.-H. Chang, H.-M. Chen, and T.-Y. Ho, "ACER: An Agglomerative Clustering Based Electrode Addressing and Routing Algorithm for Pin-Constrained EWOD Chips," *IEEE Trans. on CAD*, 2014, vol. 33, no. 9, pp. 1316-1327.

[24] Q. Wang, W. He, H. Yao, T.-Y. Ho, and Y. Cai, "SVM-Based Routability-Driven Chip-Level Design for Voltage-Aware Pin-Constrained EWOD Chips," *Proc. ISPD*, 2015, pp. 49-56.

[25] J. D. Li, S. J. Wang, K. S. M. Li, and T. Y. Ho, "Congestion- and Timing-Driven Droplet Routing for Pin-Constrained Paper-Based Microfluidic Biochips," *Proc. ASP-DAC*, 2016, pp. 593 - 598.

[26] F. Su, W. Hwang, and K. Chakrabarty, "Droplet routing in the synthesis of digital microfluidic biochips," *Proc. DATE*, 2006, pp. 1-6.

[27] Q. Wang, Y. Shen, H. Yao, T.-Y. Ho, and Y. Cai, "Practical Functional and Washing Droplet Routing for Cross-Contamination Avoidance in Digital Microfluidic Biochips" *Proc. DAC*, 2014, pp. 143:1-143:6.

[28] D. Grissom and P. Brisk, "Fast Online Synthesis of Digital Microfluidic Biochips," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2014, pp. 356-369.

[29] H. Xiang, X. P. Tang, and M. D. F. Wong, "Min-Cost Flow-Based Algorithm for Simultaneous Pin Assignment and Routing," *IEEE Trans. on CAD*, 2003, vol. 22, no. 7, pp. 870-878.

[30] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, Network Flows: Theory, Algorithms, and Applications, Prentice Hall, 1993, p. 318.

[31] Gurobi Optimizer. http://www.gurobi.com/.