# SVM-Based Routability-Driven Chip-Level Design for Voltage-Aware Pin-Constrained EWOD Chips*

Qin Wang[1], Weiran He[1], Hailong Yao[1], Tsung-Yi Ho[2], and Yici Cai[1]
Tsinghua National Laboratory for Information Science and Technology
1. Department of Computer Science & Technology, Tsinghua University 2. National Chiao Tung University
woodythu@163.com, hewr2010@gmail.com, hailongyao@tsinghua.edu.cn
ho.tsungyi@gmail.com, caiyc@mail.tsinghua.edu.cn

## ABSTRACT

The chip-level design problem is critical in pin-constrained electrowetting-on-dielectric (EWOD) biochips, which not only affects the number of control pins and PCB routing layers from the manufacturing cost point of view, but also determines the functional reliability induced by excessive applied voltage. Existing works either greedily minimize the number of control pins with degraded routability, or disregard the differences in driving voltages on the electrodes, where the trapped charge due to excessive applied voltage causes significant reliability issue. This paper presents the first SVM-based classifier for electrode addressing in chip-level design stage, which simultaneously optimizes the number of control pins, routability, as well as reliability. Experimental results on both real-life chips and synthesized benchmarks show that, compared with the state-of-the-art method, the SVM-based electrode addressing method obtains significant improvements in both routability and reliability.

## Categories and Subject Descriptors

B.7.2 [**Integrated Circuits**]: Design Aids

## General Terms

Algorithm, Performance, Design

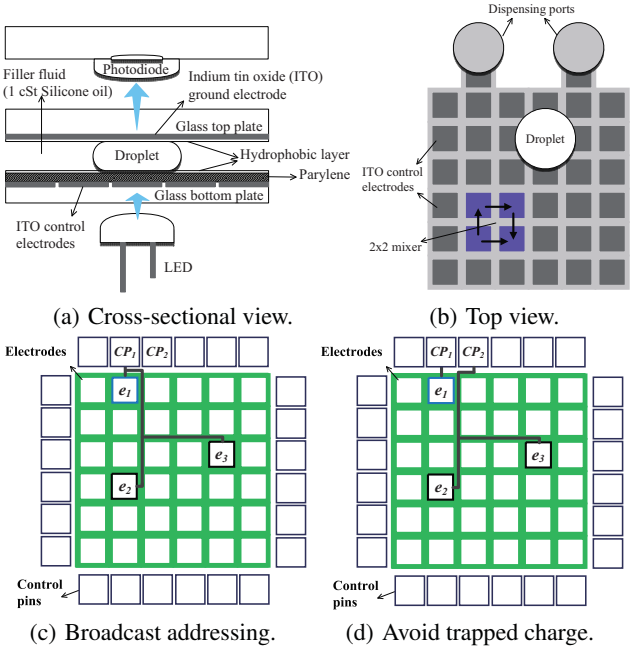
## Keywords

Digital microfluidic biochips, Electrowetting-on-dielectric, Chip-level design, Electrode addressing, SVM

## 1. INTRODUCTION

Based on the electrowetting-on-dielectric (EWOD) technology, digital microfluidic biochips (DMFBs) are revolutionizing toward miniaturization for the automation of laboratory, i.e., Lab-on-a-Chip (LoC) [1–5]. In such an LoC platform, droplets are manipulated by a 2-D array of electrodes using the electrowetting technology [5]. LoC integrates different biochemical analysis modules, such as dispenser, filter, mixer, separator, detector, etc., into a single small chip, and hence reduces sample/reagent droplets to microliter or even nanoliter scale [6]. Compared with the traditional laboratory procedures, LoC greatly improves the sensitivity, precision, and throughput, as well as reduces the analysis time and sample/reagent consumption [7]. DMFBs have many promising biochemical applications including enzymatic assays, DNA sequencing, cell-based assays, immunoassays, environmental monitoring, and clinical diagnosis [5, 8–15].

Figures 1(a) and (b) show the schematic of a DMFB based on the EWOD technology [5, 8], which controls the wetting behavior of a polarizable or conductive liquid droplet by an electric field, so as to control the movement of the droplet. Figure 1(a) shows the cross-sectional view of the DMFB. By applying a series of voltages to adjacent control electrodes, the droplets between the top and bottom plates will move to different cells as expected.




(a) Cross-sectional view. (b) Top view.

(c) Broadcast addressing. (d) Avoid trapped charge.

**Figure 1: Schematic of a digital microfluidic biochip [2, 5] and broadcast addressing considering the trapped charge problem. (a) Cross-sectional view of the EWOD chip. (b) Top view of the droplet routing layer. (c) Broadcast addressing without considering the trapped charge problem. (d) Better electrode addressing considering trapped charge for improved reliability.**

*The work of H. Yao was supported in part by the National Natural Science Foundation of China (61106104), Doctoral Fund of Ministry of Education of China (20111011328), and Tsinghua University Initiative Scientific Research Program (20121087997). The work of T.-Y. Ho was supported in part by the Taiwan Ministry of Science and Technology under grant no. MOST 102-2221-E-009-194-MY3, 103-2220-E-009-029, and 103-2923-E-009-006-MY3. The work of Y. Cai was supported by the National Natural Science Foundation of China (61274031).


*ISPD'15,* March 29–April 1, 2015, Monterey, CA, USA.

http://dx.doi.org/10.1145/2717764.2717777.

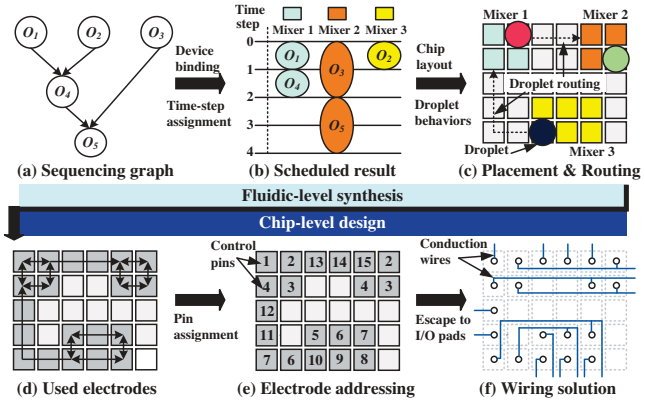



**Figure 2: Regular CAD flow of DMFBs [16].**

Here, a *cell* refers to the square room of a control electrode. Utilizing this electrowetting technology, automatic biochemical experiments can be performed. Different sample and reagent droplets can be transported to the same cell for mixing and then transported to another cell for detection. Figure 1(b) shows the 2-D electrode array. The dispensing ports are used to input/output the droplets. There are also other modules in DMFB for the biochemical experiments [2], such as mixers of different sizes, the storage cell, etc.

Figure 2 shows the typical CAD flow for DMFBs, which consists of two main stages: (1) fluidic-level synthesis, and (2) chip-level design [16]. In the past decade, there have been noticeable advances in computer-aided design (CAD) methodology for fluidic-level synthesis, including resource binding, operation scheduling, module placement, as well as washing and functional droplet routing [17–35]. Typical objectives are to minimize the assay execution time and the number of used cells, such that the driving electrodes can be minimized for power and interconnection savings. However, chip-level design is also of great importance, which directly determines the PCB (printed circuit board) fabrication cost and reliability. If the wires for electrode addressing fail to be routed, additional PCB routing layers are needed, which will unavoidably increase the fabrication cost. Besides, chip-level design significantly affects DMFB's reliability, which is a critical issue in future portable point-of-care devices. Therefore, this paper mainly addresses the routability and reliability challenges in the chip-level design stage.

To control the movement of the droplets in a programmable way, the underlying electrodes need to be connected to the peripheral electrical pads via *control pins*, where the time-varying voltages are injected by the controller. The controller generates the *actuation sequences* to the control pins for driving the electrodes, which are essentially sequences of voltage values: (1) value "1" for logic high value, (2) value "0" for logic low value, and (3) "X" denotes a don't-care value which can either be "1" or "0" without affecting the designated droplet movements. For correctly controlling the movement of the droplets, each electrode along the droplet paths is assigned an actuation sequence.

The mapping between the electrodes and the control pins is called *electrode addressing*. There are two types of electrode addressing schemes: (1) *direct addressing*, and (2) *broadcast addressing*. DMFBs in early stages use direct addressing, where each electrode is driven by an independent control pin. However, the large chip size nowadays makes direct addressing infeasible due to large number of electrodes and limited number of control pins. The DMFBs with constrained number of control pins are called *pin-constrained DMFBs* (*PDMFBs*). Broadcast addressing scheme is required for PDMFBs, where each control pin may drive multiple electrodes as long as the assay executes correctly [36]. In [36], Xu et al. presented a compatible graph to model the compatibility in actuation sequences between electrodes, and then performed clique partitioning on the graph to find compatible electrodes for sharing the same control pin.

Figures 1(c) shows an example of broadcast addressing. Assume the actuation sequences ($s$ for short) are as follows: (1) $s(e_1)$ = "01X01X110X", (2) $s(e_2)$ = "0X00111X01", (3) $s(e_3)$ = "01X0X1 11X1". Then the three electrodes are compatible with each other, and a single control pin with actuation sequence "0100111101" can correctly drive all the three electrodes simultaneously. Therefore, control pin $CP_1$ is introduced to drive the three electrodes ($e_1$, $e_2$, and $e_3$). Manhattan wires are routed for connecting the control pin and the electrodes on the *escape routing layer*, which actually form a Steiner tree. Please note that there is typically a single escape routing layer, and hence wires cannot cross each other. When there are routing failures, an additional routing layer will be required with increased fabrication cost. Therefore, the electrode addressing and routing is critical in reducing the total manufacturing cost.

Another critical issue with broadcast addressing is the trapped charge problem [37–39]. Different electrodes require different driving voltages for different types of droplet operations, e.g., droplet dispensing from input reservoir may require 60-80 volts, while droplet transportation may require at least 10-20 volts [40]. If a control pin drives two electrodes, one for droplet dispensing and one for transportation, then the minimum driving voltage needs to be 60-80 volts for effectively driving both the two electrodes. In that case, charge is trapped in the dielectric insulating layer around the electrode for droplet transportation, due to excessive applied voltage. The trapped charge reduces the electrowetting force, and thus causes wrong assay results and even permanent dielectric breakdown. For applications such as patient health monitoring, clinical diagnosis, etc., reliability is of great importance [41]. The reliability issue is even more critical in future portable point-of-care devices. Therefore, the trapped charge issue should be avoided in broadcast addressing, i.e., electrodes with different preferred driving voltages should avoid sharing the control pin as much as possible. Figures 1(d) shows an example of electrode addressing to avoid the trapped charge problem. Assume electrode $e_1$ needs to be driven by much higher voltage than $e_2$ and $e_3$. Then the three electrodes must not be driven by a single control pin. Therefore, another control pin $CP_2$ is used to drive $e_2$ and $e_3$, and $e_1$ is driven independently by $CP_1$. Please note that $e_1$ may also share the control signal with other electrodes requiring high voltages for minimized number of control pins.

Regarding the reliability issue, Huang et al. presented a method to optimize the maximum actuation time on the electrodes for better reliability [15]. However, high actuation time is not critical and will not cause the reliability issue with appropriate actuation voltage. Yeh et al. presented the first work to address the trapped charge issue with the minimum cost maximum flow formulation [39], which is an extension of [14]. The presented network flow algorithm greedily reduces the number of control pins without appropriate prediction of the routing demand. Thus, routability is a big issue in their presented method. The works in [14] and [43] presented to improve routability by simultaneous electrode addressing and wire routing. And the work in [42] presented to use decluster and re-route approach rather than ripup and re-route to improve the routability, which, to the best of our knowledge, is the latest work for routability enhancement with best reported results. However, the above works do not consider the reliability issue, and thus are not practical for real applications.

This paper presents the first routability- and reliability-driven chip-level design method based on the SVM (Support Vector Machine) classifier. The SVM-based classifiers effectively improve routability in two aspects: (1) routability between the electrodes in each cluster, and (2) routability between the clusters and the control pins. Experimental results show that the presented method obtains 100% routing completion rate for all the benchmarks. Moreover, the reliability issue induced by the trapped charge problem is also effectively addressed. Major contributions of the paper are as follows.

- The first SVM-based electrode addressing methods are presented, which obtain significant routability improvements compared with the state-of-the-art method.
- Our SVM-based electrode addressing methods can effectively improve the reliability induced by the trapped charge problem.
- Effective ripup and rerouting methods are adopted, with declustering functionality, for improving the routability.

The rest of the paper is organized as follows. Section 2 presents the problem formulation. Section 3 presents the overview of the whole chip-level design flow. Section 4 presents the SVM-based electrode addressing method. Section 5 presents the escape routing method along with ripup and rerouting technique. Section 6 presents the experimental results. Finally, conclusion is drawn in Section 7.

## 2. PROBLEM FORMULATION

This paper addresses two major problems in chip-level design, which need to be considered early in the electrode addressing stage. (1) Routability: Routing is not a trivial task because there is typically a single routing layer. Routing failures will unavoidably increase the number of routing layers, which may dramatically increase the fabrication cost. (2) Reliability (trapped charge problem): When the electrode is driven by excessive applied voltage, due to inappropriate control signal sharing in broadcast addressing, chip malfunction or even dielectric breakdown may occur. Thus, the trapped charge problem must be addressed during electrode addressing.

The routability and reliability driven chip level design problem can be stated as follows.

**Given:** (1) A set of electrodes $E = \{e_1, e_2, \ldots, e_n\}$, (2) the actuation sequences $S = \{s_1, s_2, \ldots, s_n\}$ corresponding to the electrodes in $E$, (3) the preferred voltage values $V = \{v_1, v_2, \ldots, v_n\}$ corresponding to the electrodes in $E$, (4) a threshold voltage value $V_{th}$, above which the driving voltage tends to cause the trapped charge problem, (5) the maximum number of allowed control pins $C_{max}$ for external controller, and (6) the control layer design rules.

**Find:** A feasible routing solution from all the electrodes in $E$ to the control pins with minimized total routing cost.

**Subject to:** (1) Control pin constraint: the number of used control pins must be less or equal to $C_{max}$, (2) Routing constraint: each electrode is successfully routed to a control pin without any design rules violations, (3) Broadcast-addressing constraint: the actuation sequences of the electrodes within the same cluster must be compatible with each other, and (4) Voltage constraint: for each cluster of electrodes, the driving voltage at the corresponding control pin should not be less than the preferred voltage of any member electrode.

For the trapped charge problem, we use the same measurement model as [39]. In the model, a variable $TC_i$ is introduced to represent the trapped charge on electrode $e_i$ due to excessive driving voltage. $TC_i$ is defined as

$$TC_i = \begin{cases} v_i^* - max(V_{th}, v_i), & v_i^* \geq V_{th} \\ 0, & v_i^* < V_{th} \end{cases} \quad (1)$$

where $v_i^*$ and $v_i$ represent the actual driving voltage and the preferred voltage for electrode $e_i$, respectively. $TC_i$ represents the trapped charge on $e_i$ due to excessive driving voltage.

Based on Equation (1), the overall cost of the trapped charge problem, denoted as $TC$, is computed as

$$TC = max\{TC_i | e_i \in E\} \quad (2)$$

Then the total routing cost considering the trapped charge problem is computed as

$$C = \alpha \cdot |CP| + \beta \cdot WL + \gamma \cdot TC \quad (3)$$

where $|CP|$ represents the total number of used control pins, $WL$ represents the total wire length, and $TC$ is for trapped charge as defined above. Here, $\alpha$, $\beta$ and $\gamma$ are user-defined parameters.

In the above problem formulation, the electrode addressing stage is not included. However, the electrode addressing process is of great importance, which greatly affects all the three optimization items (i.e., $|CP|$, $WL$, and $TC$), and hence determines total routing cost. Therefore, this paper mainly focuses on the electrode addressing problem targeting for enhanced routing solution with minimized total cost.

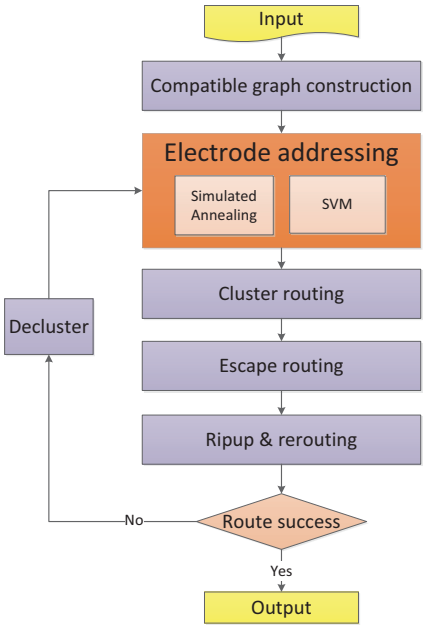



**Figure 3: Design flow of our approach.**

## 3. OVERVIEW

Figure 3 presents the overall flow of our chip-level design method. There are five major steps, i.e., compatible graph construction, electrode addressing, cluster routing, escape routing, and ripup and rerouting. First of all, we construct a compatible graph according to the actuation sequences of electrodes. In the following stages, we interconnect the electrodes within each cluster first, and then route them to the control signals by escape routing. When necessary, ripup and rerouting along with declustering are performed to improve the routing completion rate.

We propose the SVM-based strategy in electrode addressing module. The SVM-based strategy randomly generates a set of candidate clustering solutions first. Then a ranking model based

on SVM is used to obtain a set of clustering solutions with higher ranking score. Table 1 presents the variables used in the following sections and their meanings.

**Table 1: Notations used in our approach.**

| Notations | Meaning |
|---|---|
| $C_N$ | Number of clusters in a clustering solution |
| $C_{N_i}$ | Number of clusters belong to quadrant $i$ |
| $\|E\|$ | Number of electrodes for a benchmark |
| $C_S$ | Total area of a chip |
| $P_C$ | Number of clusters which have only one electrode |
| $T_B$ | Total bounding box area for the whole chip |
| $T_{B_i}$ | Bounding box area for quadrant $i$ |
| $T_O$ | Total area of bounding box overlap for the whole chip |
| $T_{O_i}$ | Area of bounding box overlap for quadrant $i$ |
| $T_{P_i}$ | Number of electrodes in cluster $i$ |
| $B_{P_i}$ | Number of electrodes on the edge of the chip in cluster $i$ |
| $O_{L_i}$ | Area of bounding box overlap for cluster $i$ |
| $B_{B_i}$ | Area of bounding box for cluster $i$ |
| $v^*_{C_i}$ | Actual driving voltage for cluster $i$ |
| $v_i$ | Preferred voltage for electrode $e_i$ |
| $V_{th}$ | Threshold voltage given by benchmarks |

## 4. SVM-BASED CLUSTERING

There are two key steps in chip-level design flow, i.e., electrode addressing and routing. There is a big design gap between the two steps, which results in many routing iterations and waste of time. What's more, sometimes the electrode addressing cannot find an feasible solution for successful routing even after many iterations. In order to minimize this gap, we propose a routing prediction model to find a electrode addressing solution with enhanced routability and reliability. The core idea of our prediction model is based on SVM (Support Vector Machine). Figure 4 gives the fundamental principle of SVM [44]. In order to discriminate the two classes, we need to find a decision boundary, which should be far away from the data of both classes. Thus, we should maximize the margin $m$, which is computed as

$$m = \frac{2\gamma}{||W||} \tag{4}$$

where $W$ is the normal vector of decision boundary, and $\gamma$ is a parameter related to the intercept of the line.

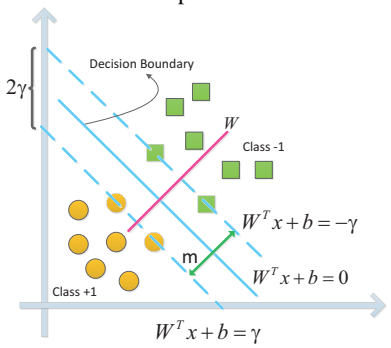



**Figure 4: Fundamental principle of SVM [44].**

SVM classifies sample vectors by generating a boundary with maximum margin of different classes. The vectors forming boundaries are called support vectors. By transforming the original problem into binary classification, multi-class classification and ranking problems can also be solved by SVM. In this paper, we use the SVM kernel in [45].

Figure 5 presents the flow of the training part in the SVM-based electrode addressing method. In this flow, the clustering module

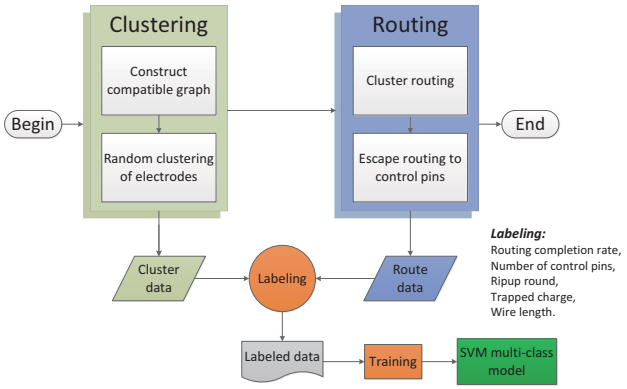



**Figure 5: Training flow of our method.**

first computes the compatible graph, and then randomly generates a set of clustering solutions according to the compatible graph. Then the routing module computes the routing solutions for each clustering solution, which includes two major steps: (1) cluster routing for each cluster, and (2) escape routing from clusters to control pins. In the clustering module, SVM features for each clustering solution are extracted as *cluster data*. When the *route data* are obtained from the routing module, the *cluster data* are labeled by the *route data*. The labeling data includes wire length, routing congestion rate, number of used control pins, trapped charge etc. We use the Equation (18) to evaluate the quality of a clustering solution. And we classify the quality of electrode clustering solutions into several levels according to the value of *Score*. Then the *training set* is obtained for the SVM classifier. Finally, we learn a SVM multi-class classifier based on the training set using the SVM kernel in [45].

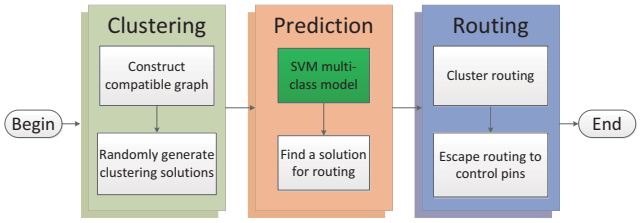



**Figure 6: Testing flow of our method.**

Figure 6 shows the SVM testing flow. After the training stage, we obtain the SVM-based multi-class classifier, which is used for the prediction module. In clustering module, a certain number of clustering solutions are generated randomly. Then the SVM classifier in prediction module is applied to obtain several clustering solutions with top ranking scores from the set of candidate clustering solutions. In the experiments, around 5 percent of the original candidate solutions are chosen. Finally the routing solution is obtained from the routing module.

Feature extraction is one of the most important step in the approach based on machine learning. In our approach, we obtain the features empirically with experimental calibration. To be brief, we divide these features into three parts: (1) general features, (2) context features, and (3) cluster features. The general features describe a clustering solution in the global view. These features are overall characteristics of a clustering solution. The context features are used to represent the routing resource information and congestion information when the clustering solution is determined. Finally, we extract each cluster's features of a solution to record some detail information includes the proportion of electrodes on the edge of the chip, bounding box area and bounding box overlap area for each cluster.

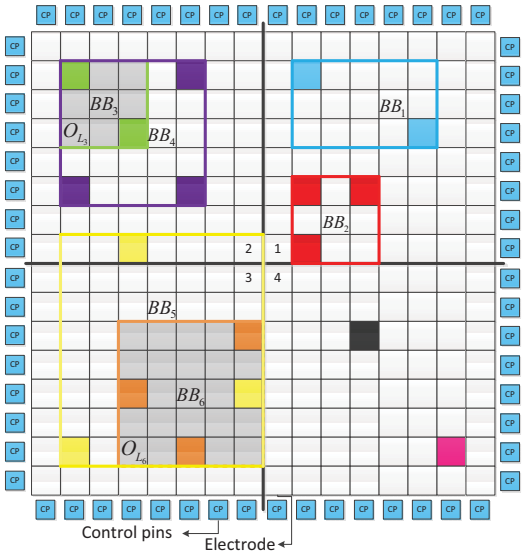



**Figure 7: Context features extraction.**

First of all, our approach calculates the bounding box for each cluster. Then we obtain some basic information of a clustering solution: (1) number of clusters, (2) total area of bounding boxes, (3) number of clusters with a single electrode, and (4) total area of bounding box overlaps. Here we use vector $G = (g_1, g_2, g_3, g_4)$ to represent the above general features. In addition, the area of the chip and number of electrodes are used for normalization. In this way, our model can be applied to different types of benchmarks with various. The definitions of the above features are as follows:

$$g_1 = \frac{C_N}{|E|}, \quad g_2 = \frac{C_S}{T_B}, \quad g_3 = \frac{P_C}{C_N}, \quad g_4 = \frac{C_S}{T_O} \tag{5}$$

Figure 7 presents an example of context features extraction. In order to obtain the context features, we propose the following model. We first compute the bounding box for each cluster, then we divide the whole chip into four quadrants. If the center point of a bounding box is in quadrant $i\{i \in (1,2,3,4)\}$, we define that this cluster belongs to this quadrant. Each quadrant collects the information of clusters belong to itself. In Figure 7, where the whole chip is partitioned into 4 quadrants, the electrodes with the same color belong to the same cluster. Then each quadrant calculates the bounding box area and bounding box overlap area separately. In this example, bounding boxes $BB_1$ and $BB_2$ belong to quadrant 1. Bounding boxes $BB_3$, $BB_4$ and bounding box overlap $O_{L_3}$ belong to quadrant 2. Quadrant 3 has $BB_5$, $BB_6$ and bounding box overlap $O_{L_6}$. Quadrant 4 has two clusters with only one electrode. Finally, the data of the four quadrants forms a context feature vector, denoted as vector $C = (P,R,N)$. It contains three vectors, which are represented as follows:

$$P = (p_1, p_2, p_3, p_4), \quad p_i = \frac{C_{N_i}}{C_N} \tag{6}$$

$$R = (r_1, r_2, r_3, r_4), \quad r_i = \frac{T_{B_i}}{T_B} \tag{7}$$

$$N = (n_1, n_2, n_3, n_4), \quad n_i = \frac{T_{O_i}}{T_O} \tag{8}$$

where $p_i$ denotes the proportion of clusters belonging to quadrant $i$. $r_i$ records the proportion of bounding box area in quadrant $i$, and $n_i$ represents the proportion of overlap area in quadrant $i$.

Cluster features describe a clustering solution in detail, which are helpful to routability especially for escaping routing from cluster to control pins. Vector $D = (B,O,A)$ represents the cluster features, where $B$, $O$, and $A$ are defined as follows:

$$B = (b_1, b_2, b_3, b_4, b_5) \tag{9}$$

$$b_i = \frac{(\sum_{j=1}^{C_N} \mathcal{P}(\frac{B_{P_j}}{T_{P_j}}))}{C_N} \tag{10}$$

$$O = (o_1, o_2, o_3, o_4, o_5) \tag{11}$$

$$o_i = \frac{(\sum_{j=1}^{C_N} \mathcal{P}(\frac{O_{L_j}}{C_S}))}{C_N} \tag{12}$$

$$A = (a_1, a_2, a_3, a_4, a_5) \tag{13}$$

$$a_i = \frac{(\sum_{j=1}^{C_N} \mathcal{P}(\frac{B_{B_j}}{C_S}))}{C_N} \tag{14}$$

Here, vectors $B$, $O$, $A$ describe the distribution of some variables. And these variables may be related to routability and reliability of a clustering solution. $m_i$ and $n_i$ are user-defined parameters. In Equations (10), (12), and (14), $\mathcal{P}$ is 1 when $\frac{B_{P_j}}{T_{P_j}} \in (m_i, n_i)$, $\frac{O_{L_j}}{C_S} \in (m_i, n_i)$, or $\frac{B_{B_j}}{C_S} \in (m_i, n_i)$. Otherwise, $\mathcal{P}$ is 0. In the experiment, $(m_i, n_i)$ are set to be (0.1, 0.3), (0.3, 0.5), (0.5, 0.7), (0.7, 0.9), (0.9, 1), where $i$ is from 1 to 5. $C_S$ is used for normalization.

To deal with the trapped charge problem, we present a feature $V$. We extract feature $V$ from the definition of trapped charge problem, which is computed as

$$V = \frac{(\sum_{i=1}^{C_N} \mathcal{P}(v^*_{C_i} > V_{th}))}{C_N} \tag{15}$$

$$v^*_{C_i} = max\{v_j | e_j \in cluster \quad i\} \tag{16}$$

In Equation (15), $\mathcal{P}$ is 1 when $v^*_{C_i} > V_{th}$. Otherwise, $\mathcal{P}$ is 0.

In the routing module, our approach records the routing completion rate $F_s$ before ripup and rerouting, and the total ripup round $R_t$. These two variables form variable $R$, which evaluates of routability of electrode addressing solution (see Equation (17)). After the routing stage, we define a function *Score* to evaluate the quality of a clustering solution as follows:

$$R = \frac{\omega \cdot F_s}{\theta \cdot R_t} \quad (\omega + \theta = 1) \tag{17}$$

$$Score = \frac{R}{\alpha \cdot |CP| + \beta \cdot WL + \gamma \cdot TC} \cdot C_S \cdot E_C \tag{18}$$

The $C_S$ and $E_C$ are also used for normalization. $\omega$ and $\theta$ are user-defined parameters and their sum is 1. They are coefficients measuring the importance of the two factors. Our approach classify the clustering solutions into $n$ classes according to the value of *Score*. In the experiments, $\omega$ is set to be 0.7, $\theta$ is set to be 0.3, $\alpha$, $\beta$, $\gamma$ are all set to be 1. Because we suppose that, final routing completion rate is more important than ripup rounds. But total wire length, number of used control pins, and trapped charge are equally important.

In order to obtain a SVM model with better performance, we design two different feature vectors $feature_1$ and $feature_2$. These two vectors are applied to train different SVM models, i.e., $SVM_1$ and $SVM_2$. In Section 6, we compare the experimental results of the two models. The two feature vectors can be represented as follows:

$$feature_1 = (G,C,V), \quad feature_2 = (G,C,V,D) \tag{19}$$

Vector $D$ records the cluster data, i.e., proportion of electrodes on the edge of the chip, bounding box area of a cluster, bounding box overlap area of a cluster which are supposed to contribute to routability classify. And our experimental results show that $SVM_2$ has better performance than $SVM_1$ on routability and running time as expected.

**Table 3: Results without trapped charge consideration.**

| Benchmark | First | | Final | | #Ripup | | \|CP\| | | WL | | RT (s) | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ACER | $SVM_2$ | ACER | $SVM_2$ | ACER | $SVM_2$ | ACER | $SVM_2$ | ACER | $SVM_2$ | ACER | $SVM_2$ |
| amino-acid-1 | 92.31 | 80.36 | 100.00 | 100.00 | 1 | 2 | 14 | 13 | 290 | 276 | 0.05 | 0.32 |
| amino-acid-2 | 87.50 | 91.74 | 100.00 | 100.00 | 1 | 1 | 17 | 17 | 364 | 324 | 0.07 | 0.34 |
| protein-1 | 29.63 | 71.67 | 70.97 | 100.00 | 50 | 6 | 22 | 37 | 698 | 727 | 12.60 | 3.52 |
| protein-2 | 21.62 | 52.86 | 100.00 | 100.00 | 8 | 9 | 45 | 45 | 1135 | 1108 | 1.59 | 1.42 |
| dilution | 47.06 | 40.88 | 100.00 | 100.00 | 16 | 9 | 44 | 42 | 1395 | 1341 | 4.70 | 3.42 |
| multiplex | 83.33 | 85.24 | 100.00 | 100.00 | 6 | 6 | 50 | 49 | 1394 | 1411 | 0.46 | 1.29 |
| random-1 | 66.67 | 76.30 | 100.00 | 100.00 | 3 | 2 | 15 | 11 | 453 | 458 | 0.21 | 0.33 |
| random-2 | 42.11 | 66.75 | 85.71 | 100.00 | 50 | 4 | 18 | 21 | 1052 | 898 | 9.27 | 1.42 |
| random-3 | 34.69 | 38.13 | 100.00 | 100.00 | 10 | 9 | 57 | 46 | 1926 | 2040 | 7.24 | 8.94 |
| random-4 | 35.29 | 27.25 | 98.73 | 100.00 | 50 | 16 | 78 | 77 | 3977 | 4801 | 37.76 | 239.64 |
| random-5 | 29.49 | 31.87 | 100.00 | 100.00 | 25 | 13 | 92 | 70 | 9039 | 6576 | 331.89 | 978.17 |
| random-6 | 26.32 | 34.79 | 90.80 | 100.00 | 50 | 13 | 79 | 74 | 8251 | 8034 | 383.34 | 312.43 |
| random-7 | 19.17 | 34.41 | 93.53 | 100.00 | 50 | 28 | 130 | 117 | 12976 | 11418 | 2242.50 | 462.60 |
| Avg. | 47.32 | 56.33 | 95.21 | 100.00 | 25 | 9 | 51 | 48 | 3304 | 3032 | 232.21 | 154.91 |

**Table 2: Statistics of benchmarks.**

| Benchmark | Width | Height | Area | #E | Voltage(v) |
|---|---|---|---|---|---|
| amino-acid-1 | 6 | 8 | 1008 | 20 | 50 |
| amino-acid-2 | 6 | 8 | 1008 | 24 | 50 |
| protein-1 | 13 | 13 | 3136 | 34 | 50 |
| protein-2 | 13 | 13 | 3136 | 51 | 50 |
| dilution | 15 | 15 | 4096 | 54 | 50 |
| multiplex | 15 | 15 | 4096 | 59 | 50 |
| random-1 | 10 | 10 | 1936 | 20 | 50 |
| random-2 | 15 | 15 | 4096 | 30 | 50 |
| random-3 | 20 | 20 | 7056 | 60 | 50 |
| random-4 | 30 | 30 | 15376 | 90 | 50 |
| random-5 | 50 | 50 | 41616 | 100 | 50 |
| random-6 | 50 | 50 | 41616 | 100 | 50 |
| random-7 | 60 | 60 | 59536 | 150 | 50 |

# 5. ESCAPE ROUTING TO CONTROL PINS

When the clusters are generated using the above presented methods, we start the escape routing process to connect the control pins. The routing process consists of two major stages: (1) routing between the electrodes within each cluster, and (2) escape routing from the clusters to the peripheral control pins. When all the clusters are successfully routed, the number of used control pins is equal to the number of clusters. The objective of the escape routing problem is to compute the routing paths connecting clusters of electrodes with properly selected control pins for minimizing the total wire length with enhanced routing completion rate.

For routing within each cluster of multiple electrodes, the minimum spanning tree (MST) is first constructed to determine the connection topology. When the MST edges are computed, the edges are sequentially routed one by one using the A* search algorithm [46]. Using randomly determined order for MST edges, there are three different cases: (1) routing between two electrodes, (2) routing between a electrode and a partially routed path, and (3) routing between two partially routed paths. For the three different cases, we adopt different routing methods, i.e., point-to-point, point-to-path, and path-to-path A* search algorithms. The modified multi-source multi-target A* search algorithm enhances routability with reduced total wire length. For escape routing from clusters to the control pins, a similar multi-source multi-target A* search algorithm is used, which simultaneously searches from all the routing grids along the paths of the cluster to all the available control pins.

After the escape routing process, the whole routing process will be finished if all the electrodes are successfully routed. However, routing failures may occur in congested designs. As a result, the declustering and rerouting process is needed for improving the routing completion rate. In this stage, the blocking paths are identified and ripped up, which possibly declusters the original cluster into smaller ones. These smaller clusters are then routed to the control pins independently. The declustering and rerouting process is iterated, until all the electrodes are successfully routed or a predefined threshold value is reached.

# 6. EXPERIMENTAL RESULTS

We have implemented our routability- and reliability-driven chip-level design flow in C++, and tested it on a 2.40GHz 16-core Intel Xeon Linux workstation with 40GB memory. Only a single thread is used for the experiments.

Table 2 shows the details of the benchmarks, where “Width” and "Height" represent the size of a chip, “Area” denotes the actual routing area considering the routing grids between adjacent electrodes. There exist 3 routing grids between the adjacent electrodes. “#E” gives the number of electrodes, and “Voltage” records the threshold voltage for trapped charge issue.

To evaluate the performance of our methods, we compare the results with ACER in [42]. Due to lack of source code and executable of ACER, we implemented ACER by ourselves, and then applied it to our design flow for comparison. Because ACER does not consider the trapped charge problem, we compare the methods with and without the trapped charge consideration, respectively. Table 3 presents the experimental results of the methods without considering trapped charge problem: (1) ACER, (2) $SVM_1$ in Section 4, and (3) $SVM_2$ in Section 4. “First” gives the routing completion rate immediately after the first round of routing, without ripup and rerouting. “Final” gives the final routing completion rate after ripup and rerouting with the iteration threshold set to be 50. “#Ripup” represents the number of ripup and rerouting iterations. The above factors are used to evaluate the routability of the electrode clustering solutions. “|CP|” denotes the number of used control pins, “WL” gives the total wire length, and “RT” records the total running time. “|CP|”, “WL” and “TC” (related to trapped charge in the following tables) are used to evaluate the reliability and manufacturing cost.

From Table 3, the SVM-based electrode clustering method has better performance on routability and control pin minimization

**Table 4: Results with trapped charge consideration.**

| Benchmark | First | | Final | | #Ripup | | \|CP\| | | WL | | TC (v) | | RT (s) | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ACER | $SVM_2$ | ACER | $SVM_2$ | ACER | $SVM_2$ | ACER | $SVM_2$ | ACER | $SVM_2$ | ACER | $SVM_2$ | ACER | $SVM_2$ |
| amino-acid-1 | 92.31 | 83.27 | 100.00 | 100.00 | 1 | 2 | 14 | 12 | 290 | 279 | 19 | 14 | 0.05 | 0.37 |
| amino-acid-2 | 87.50 | 90.78 | 100.00 | 100.00 | 1 | 1 | 17 | 16 | 364 | 338 | 18 | 15 | 0.07 | 0.34 |
| protein-1 | 29.63 | 75.32 | 70.97 | 100.00 | 50 | 7 | 22 | 37 | 698 | 731 | 19 | 12 | 12.60 | 3.48 |
| protein-2 | 21.62 | 49.66 | 100.00 | 100.00 | 8 | 9 | 45 | 44 | 1135 | 1118 | 19 | 17 | 1.59 | 1.32 |
| dilution | 47.06 | 50.58 | 100.00 | 100.00 | 16 | 9 | 44 | 42 | 1395 | 1373 | 19 | 18 | 4.70 | 3.52 |
| multiplex | 83.33 | 84.64 | 100.00 | 100.00 | 6 | 6 | 50 | 47 | 1394 | 1440 | 19 | 14 | 0.46 | 1.34 |
| random-1 | 66.67 | 86.20 | 100.00 | 100.00 | 3 | 3 | 15 | 11 | 453 | 454 | 17 | 13 | 0.21 | 0.29 |
| random-2 | 42.11 | 69.85 | 85.71 | 100.00 | 50 | 3 | 18 | 20 | 1052 | 889 | 19 | 11 | 9.27 | 1.52 |
| random-3 | 34.69 | 48.16 | 100.00 | 100.00 | 10 | 9 | 57 | 45 | 1926 | 2072 | 19 | 12 | 7.24 | 8.74 |
| random-4 | 35.29 | 47.25 | 98.73 | 100.00 | 50 | 17 | 78 | 77 | 3977 | 4829 | 18 | 16 | 37.76 | 239.64 |
| random-5 | 29.49 | 30.37 | 100.00 | 100.00 | 25 | 12 | 92 | 69 | 9039 | 6583 | 19 | 17 | 331.89 | 982.47 |
| random-6 | 26.32 | 39.77 | 90.80 | 100.00 | 50 | 12 | 79 | 75 | 8251 | 8054 | 19 | 12 | 383.34 | 314.13 |
| random-7 | 19.17 | 39.21 | 93.53 | 100.00 | 50 | 27 | 130 | 115 | 12976 | 11398 | 19 | 18 | 2242.50 | 463.60 |
| Avg. | 47.32 | 61.16 | 95.21 | 100.00 | 25 | 9 | 51 | 47 | 3304 | 3043 | 19 | 15 | 232.21 | 155.44 |

than ACER, especially when the size of benchmark is large. Although the SVM-based methods consume more time for certain benchmarks, the overhead is acceptable considering the performance advantage.

Table 4 gives the experimental results considering the trapped charge problem. "TC" denotes the variable defined in Equation (2). From the results, our methods, especially $SVM_2$, is much better than ACER considering the reliability issue. Moreover, the routability and number of used control pins of $SVM_2$ are also much better than ACER.

Table 5 shows that $SVM_2$ obtains better solutions on routability than $SVM_1$. This is because $SVM_2$ includes more features than $SVM_1$, and these features are effective for routability prediction. In addition, $SVM_2$ is faster than $SVM_1$ because $SVM_2$ can obtain clustering solutions with better routability, and this effectively reduces the time consumption in ripup and rerouting.

# 7. CONCLUSION

We have presented the first SVM-based chip-level design flow considering both routability and reliability enhancements for pin-constrained EWOD biochips. Our flow features effective SVM-based electrode addressing methods. Experimental results show notable improvements over the state-of-the-art method.

# 8. REFERENCES


[1] F. K. Balagadde, L. You, C. L. Hansen, F. H. Arnold, and S. R. Quake, "Long-Term Monitoring of Bacteria Undergoing Programmed Population Control in a Microchemostat," *Science*, vol. 309, no. 5731, pp. 137-140, 2005.

[2] K. Chakrabarty and F. Su, "Digital Microfluidic Biochips," *CRC Press*, 2006.

[3] G. M. Whitesides, "The Origins and the Future of Microfluidics," *Nature*, vol. 442, no. 7101, pp. 368-373, 2006.

[4] P. Yager, T. Edwards, E. Fu, K. Helton, K. Nelson, M. R. Tam, and B. H. Weigl, "Microfluidic Diagnostic Technologies for Global Public Health," *Nature*, vol. 442, no. 7101, pp. 412-418, 2006.

[5] R. B. Fair, A. Khlystov, T. D. Tailor, V. Ivanov, R. D. Evans, P. B. Griffin, V. "Chemical and Biological Applications of Digital-Microfluidic Devices," *IEEE Design and Test of Computers*, vol. 24, no. 1, pp. 10-24, 2007.

[6] T. Thorsen, S. J. Maerkl, and S. R. Quake, "Microfluidic Large-Scale Integration," *Science*, vol. 298, no. 5593, pp. 580-584, 2002.

[7] D. Mark, S. Haeberle, G. Roth, F. von Stetten, and R. Zengerle, "Microfluidic Lab-on-a-Chip Platforms: Requirements, Characteristics and Applications," *Chemical Society Reviews*, vol. 39, no. 3, pp. 1153-1182, 2010.

[8] M. G. Pollack, A. D. Shenderov, and R. B. Fair, "Electrowetting-Based Actuation of Droplets for Integrated Microfluidics," *Lab Chip*, vol. 2, no. 2, pp. 96-101, 2002.

[9] 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*, pp. 310-315, 2004.

[10] F. SU, K. Chakrabarty, and R. B. Fair, "Microfluidics-Based Biochips: Technology Issues, Implementation Platforms, and Design-Automation Challenges," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 2, pp. 211-223, 2006.

[11] I. Barbulovic-Nad, H. Yang, P. S. Park et al., "Digital Microfluidics for Cell-based Assays," *Lab Chip*, pp. 519-526, 2008.

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

[13] T.-Y. Ho, J. Zeng, and K. Chakrabarty, "Digital Microfluidic Biochips: A Vision for Functional Diversity and More Than Moore," *Prof. IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2010, pp. 578-585.

[14] 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*, vol. 30, no. 12, pp. 1786-1799, 2011.

[15] T.-W. Huang, T.-Y. Ho, and K. Chakrabarty, "Reliability-Oriented Broadcast Electrode-Addressing for Pin-Constrained Digital Microfluidic Biochips," *Proc. of IEEE/ACM International Conference on Computer-Aided Design*, 2011, pp. 448-455.

[16] 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.

[17] F. SU and K. Chakrabarty, "Architectural-Level Synthesis of Digital Microfluidics-Based Biochips," *IEEE/ACM International Conference on Computer Aided Design*, 2004, pp. 223-228.

[18] D. Grissom, K. O'Neal, B. Preciado, H. Patel, R. Doherty, N. Liao, and P. Brisk, "A Digital Microfluidic Biochip Synthesis Framework," *Proc. of IEEE/IFIP International Conference on VLSI and System-on-Chip (VLSI-SoC)*, 2012, pp. 177-182.

[19] M. Cho and D. Z. Pan, "A High-Performance Droplet Routing Algorithm for Digital Microfluidic Biochips," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 10, pp. 1714-1724, 2008.

[20] M. Cho and D. Z. Pan, "A High-Performance Droplet Router for Digital Microfluidic Biochips," *Proc. of International Symposium on Physical Design*, 2008, pp. 200-206.

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

**Table 5: Comparison between $SVM_1$ and $SVM_2$.**

| Benchmark | First | | Final | | #Ripup | | \|CP\| | | WL | | TC (v) | | RT (s) | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $SVM_1$ | $SVM_2$ | $SVM_1$ | $SVM_2$ | $SVM_1$ | $SVM_2$ | $SVM_1$ | $SVM_2$ | $SVM_1$ | $SVM_2$ | $SVM_1$ | $SVM_2$ | $SVM_1$ | $SVM_2$ |
| amino-acid-1 | 88.31 | 83.27 | 100.00 | 100.00 | 1 | 2 | 12 | 12 | 289 | 279 | 11 | 14 | 0.66 | 0.37 |
| amino-acid-2 | 79.85 | 90.78 | 100.00 | 100.00 | 1 | 1 | 17 | 16 | 324 | 338 | 12 | 15 | 0.52 | 0.34 |
| protein-1 | 64.80 | 75.32 | 100.00 | 100.00 | 7 | 7 | 33 | 37 | 973 | 731 | 16 | 12 | 0.63 | 3.48 |
| protein-2 | 51.48 | 49.66 | 100.00 | 100.00 | 9 | 9 | 42 | 44 | 1190 | 1118 | 19 | 17 | 1.34 | 1.32 |
| dilution | 34.54 | 50.58 | 100.00 | 100.00 | 9 | 9 | 41 | 42 | 1496 | 1373 | 18 | 18 | 3.12 | 3.52 |
| multiplex | 88.48 | 84.64 | 100.00 | 100.00 | 6 | 6 | 48 | 47 | 1372 | 1440 | 15 | 14 | 2.05 | 1.34 |
| random-1 | 84.60 | 86.20 | 100.00 | 100.00 | 1 | 3 | 11 | 11 | 429 | 454 | 18 | 13 | 0.25 | 0.29 |
| random-2 | 71.48 | 69.85 | 100.00 | 100.00 | 5 | 3 | 25 | 20 | 979 | 889 | 19 | 11 | 1.86 | 1.52 |
| random-3 | 39.89 | 48.16 | 100.00 | 100.00 | 12 | 9 | 47 | 45 | 2459 | 2072 | 18 | 12 | 38.66 | 8.74 |
| random-4 | 27.98 | 47.25 | 100.00 | 100.00 | 9 | 17 | 54 | 77 | 3433 | 4829 | 15 | 16 | 191.26 | 239.64 |
| random-5 | 37.95 | 30.37 | 100.00 | 100.00 | 13 | 12 | 73 | 69 | 6154 | 6583 | 18 | 17 | 1201.54 | 982.47 |
| random-6 | 38.81 | 39.77 | 100.00 | 100.00 | 22 | 12 | 80 | 75 | 7455 | 8054 | 18 | 12 | 429.61 | 314.13 |
| random-7 | 25.13 | 39.21 | 100.00 | 100.00 | 26 | 27 | 118 | 115 | 12331 | 11398 | 17 | 18 | 1064.34 | 463.60 |
| Avg. | 56.41 | 61.16 | 100.00 | 100.00 | 9 | 9 | 46 | 47 | 2991 | 3043 | 16 | 15 | 225.53 | 155.44 |


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

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

[24] 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)*, vol. 3, no. 3, Artical No. 13, 2007.

[25] 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 Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 11, pp. 1928-1941, 2008.

[26] T.-W. Huang and T.-Y. Ho, "A Two-Stage Integer Linear Programming-Based Droplet Routing Algorithm for Pin-Constrained Digital Microfluidic Biochips," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 2, pp. 215-228, 2011.

[27] P.-H. Yuh, S. S. Sapatnekar, C.-L. Yang, and Y.-W. Chang, "A Progressive-ILP-Based Routing Algorithm for the Synthesis of Cross-Referencing Biochips," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 9, pp. 1295-1306, 2009.

[28] Z. Xiao and E. F. Y. Young, "CrossRouter: A Droplet Router for Cross-Referencing Digital Microfluidic Biochips," *Proc. of Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2010, pp. 269-274.

[29] M. Campàs and I. Katakis, "DNA Biochip Arraying, Detection and Amplification Strategies," *TrAC Trends in Analytical Chemistry*, vol. 23, no. 1, pp. 49-62, 2004.

[30] Y. Zhao and K. Chakrabarty, "Cross-contamination Avoidance for Droplet Routing in Digital Microfluidic Biochips," *Proc. Design, Automation and Test in Europe (DATE)*, 2009, pp. 1290-1295.

[31] T.-W. Huang, C.-H. Lin, and T.-Y. Ho, "A Contamination Aware Droplet Routing Algorithm for the Synthesis of Digital Microfluidic Biochips," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 29, no. 11, pp. 1682-1695, 2010.

[32] Y. Zhao and K. Chakrabarty, "Synchronization of Washing Operations with Droplet Routing for Cross-contamination Avoidance in Digital Microfluidic Biochips," *Proc. of Design Automation Conference*, 2010, pp. 635-640.

[33] C. C. Y. Lin and Y.-W. Chang, "Cross-Contamination Aware Design Methodology for Pin-Constrained Digital Microfluidic Biochips," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 6, pp. 817-828, 2011.

[34] D. Mitra, S. Ghoshal, H. Rahaman, K. Chakrabarty, B. B.Bhattacharya, K. Chakrabarty, and B. B. Bhattacharya, "On Residue Removal in Digital Microfluidic Biochips," *Proc. of the Great Lakes Symposium on VLSI*, pp. 1-4, 2011.

[35] 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. of Design Automation Conference (DAC)*, 2014, pp. 1-6.

[36] T. Xu and K. Chakrabarty, "Broadcast Electrode-Addressing for Pin-Constrained Multi-Functional Digital Microfluidic Biochips," *Proc. of IEEE/ACM Design Automation Conference*, 2008, pp. 173-178.

[37] H. J. J. Verheijen and M. W. J. Prins, "Reversible Electrowetting and Trapping of Charge: Model and Experiments," *Langmuir*, vol. 15, no. 20, pp. 6616-6620, 1999.

[38] A. I. Drygiannakis, A. G. Papathanasiou, and A. G. Boudouvis, "On the Connection Between Dielectric Breakdown Strength, Trapping of Charge, and Contact Angle Saturation in Electrowetting," *Langmuir*, vol. 25, no. 1, pp. 147-152, 2009.

[39] 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. of IEEE/ACM International Conference on Computer-Aided Design*, 2012, pp. 353-360.

[40] R. Fair, "Digital Microfluidics: Is a True Lab-on-a-Chip Possible?" *Microfluid Nanofluidics*, vol. 3, no. 3, pp. 245-281, 2007.

[41] K. Chakrabarty, "Towards Fault-Tolerant Digital Microfluidic Lab-on-Chip: Defects, Fault Modeling, Testing, and Reconfiguration," *Transactions of the IRE Professional Group on Audio*, pp. 329-332, 2008.

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

[43] J.-W. Chang, T.-W. Huang, and T.-Y. Ho, "An ILP-Based Obstacle-Avoiding Routing Algorithm for Pin-Constrained EWOD Chips," *Proc. of Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2012, pp. 67-72.

[44] N. Cristianini and J. Shawe-Taylor, An Introduction to Support Vector Machines, *Cambridge University Press*, 2000.

[45] T. Joachims, "Making Large-Scale SVM Learning Practical," *B. Scholkopf and C. Burges, and A. Smola, Advances in Kernel Methods - Support Vector Learning, MIT-Press*, 1999.

[46] P. E. Hart, N. J. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100-107, 1968.