ILP-Based Pin-Count Aware Design Methodology for Microfluidic Biochips *

Cliff Chiung-Yu Lin[1], and Yao-Wen Chang[1,2]
[1]Graduate Institute of Electronics Engineering, National Taiwan University, Taipei 106, Taiwan
[2]Department of Electrical Engineering, National Taiwan University, Taipei 106, Taiwan
chiungyu@eda.ee.ntu.edu.tw; ywchang@cc.ee.ntu.edu.tw

ABSTRACT

Digital microfluidic biochips have emerged as a popular alternative for laboratory experiments. To make the biochip feasible for practical applications, pin-count reduction is a key problem to higher-level integration of reactions on a biochip. Most previous works approach the problem by post-processing the placement and routing solutions to share compatible control signals; however, the quality of such sharing algorithms is inevitably limited by the placement and routing solutions. We present in this paper a comprehensive pin-constrained biochip design flow that addresses the pin-count issue at all design stages. The proposed flow consists of three major stages: (1) pin-count aware stage assignment that partitions the reactions in the given bioassay into execution stages, (2) pin-count aware device assignment that determines a specific device used for each reaction, and (3) guided placement, routing, and pin assignment that utilize the pin-count saving properties from the stage and device assignments to optimize the assay time and pin count. For both the stage and device assignments, exact ILP formulations and effective solution-space reduction schemes are proposed to minimize the assay time and pin count. Experimental results show the efficiency of our algorithms/flow and a 55–57% pin-count reduction over the state-of-the-art algorithms/flow.

Categories and Subject Descriptors

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

General Terms

Algorithms, Performance, Design

Keywords

Microfludics, biochip, design methodology, integer linear programming

1. INTRODUCTION

Digital microfluidic biochips, also referred to as lab-on-a-chip or biochips, have emerged as an alternative for conventional laboratory experiments. With lower cost and higher immunity to human errors, the technology is gaining increasing applications including DNA sequencing, immunoassays, environmental toxicity monitoring, and point-of-care diagnosis of diseases [8].

Recently, the second-generation (digital) microfluidic biochips have been proposed [5,12]. Typically, a digital microfluidic biochip consists of a two-dimensional (2D) electrode array and peripheral devices (optical detection sites, dispensing ports, etc. [5]. On a digital microfluidic biochip, movements of the *droplets* are controlled by the electrohydrodynamic force generated by the electrodes. By assigning time-varying voltage values to turn on/off the electrodes on the digital microfluidic biochip, we can move the droplets around the entire 2D array and perform fundamental microfluidic operations (i.e., mixing reactions) for different bioassays. These operations performed under the control of the electrodes are also called *reconfigurable* operations because of their flexibility in area (electrodes involved) and in execution time. A reconfigurable operation can be carried out anywhere on the 2D plane. And it can be completed, for example, slowly with 4 electrodes, or faster with 8 electrodes.

The reconfigurability of the operations, as it brings about the freedom in design, raises issues both in scheduling and in electrode control. For a given bioassay, different completion time and resource requirements can be achieved by changing the time and location schedule of the involved operations. Therefore, it has raised active design automation discussions in the past few years [10,11,18]. On the other hand, as the chip size grows, it becomes necessary to restructure the electrode control mechanism, or the unlimited number of control signals will be impossible to be implemented.

Originally, the electrodes are addressed and controlled independently, that is, each electrode is assigned a dedicated control pin. This kind of biochips, which are also referred to as *direct-addressing biochips*, provide great flexibility for droplet movement; yet they suffer from the increasing design complexity. Specifically, the routing problem for the large number of control pins has made this architecture only applicable to small-scale biochips [14].

Lately, alternative driving schemes have been proposed to alleviate the growth of the required number of control pins. Pin-constrained digital microfluidic biochips [8], one of the major genres of the pin-count reduction approaches, reduce the number of pins to be routed to the electrodes by assigning each control pin to multiple electrodes; that is, multiple electrodes are controlled by a single control signal, and thus they are turned on/off simultaneously. Regarding this, efforts have been made to cluster the electrodes that can be controlled together without introducing unexpected droplet behaviors [14,16].

However, these currently available pin-count reduction methods focus on the electrode partitioning, control signal merging, and pin assignment as the last step of the design flow, while the feasibility and effectiveness of such methods actually depend on the scheduling, placement, and routing results of the given design. Therefore, it is desirable to consider the pin-count constraint at earlier stages of the pin-constrained biochip design flow. Consequently, we propose in this paper a novel pin-count aware design flow for pin-constrained biochips and efficient algorithms for the corresponding steps in the flow.

1.1 Previous Work

Previous works on the pin-constrained biochip problem generally address the problem after the biochip is placed and routed, and its design flow is illustrated in Figure 1(a). The flow consists of three major stages. The first stage, referred to as scheduling or placement, assigns the time slots and electrodes to each reactions in the bioassay and are resolved in [4,7,11,18]. The second stage, routing, determines the paths that the droplets move around the biochip and are worked out in [2,3,6,13,17]. Note that the solutions for these two stages are designed for direct-addressing biochips and do not address the pin-count constraint. Finally the electrode partitioning or control signal merging, followed by pin assignment, in [14,16], is used to reduce the final pin count. A fundamental problem with the previous flow is that only the last stage of the flow is pin-count aware, while the properties of the scheduling, placement, and routing results do affect the space for pin-count reduction. Currently, due to the lacking of specialized front-end design automation methods for pin-constrained biochips, these properties cannot be maintained/utilized well, and thus the quality of the pin-reduction will inevitably be restricted.
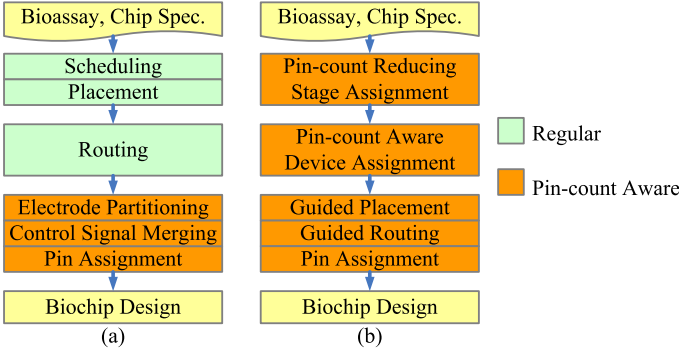

Figure 1: Comparison of the previous design flow and our proposed one: (a) Previous design flow, in which pin count is considered only during sequence merging and pin assignment. (b) Our proposed pin-count aware design flow.

1.2 Our Contributions

Unlike the previous flow that addresses the pin-constrained issue at the last stage, we propose in this paper a novel comprehensive pin-count aware design flow. The flow is summarized in Figure 1(b) and will be detailed in Section 2. It consists of three major stages: (1) pin-count aware stage assignment, (2) pin-count aware device assignment, and (3) guided placement, routing, and pin assignment. Besides adding the pin-count considerations into the flow, we expand the scheduling stage into *stage assignment* and *device assignment* to reflect their criticality in pin-count reduction. For both stages, we propose exact ILP formulations and effective reduction methods to minimize the assay completion time and the pin count. As mentioned earlier, pin-count saving properties should be maintained along the design flow. We also present guidelines for the placement, routing, and pin assignment to maintain the low pin-count properties obtained earlier at the assignment stage.

With the flow and the algorithms for realizing these properties, our contributions can be summarized as follows.

- We propose a dedicated pin-count aware design flow for pin-constrained biochips to consider pin count throughout all stages of the flow.
- We identify the factors that would affect the pin-count along the design and explore the properties that are favorable for pin-count reduction. The properties are universal to the pin-constrained biochips and would be helpful for future development with our and other design flows.
- We derive an exact ILP formulation for stage assignment that models the synchronous control of reactions and minimize the assay completion time. We also provide an effec-

tive scheme to reduce the problem size and an approximation to reduce the assay time to speed up the process.

- We derive an exact ILP formulation for device assignment that minimizes the number of branches from a device fanout, which reduces the corresponding pin-count demand, potential routing complexity, and fault tolerance. A corresponding problem size reduction method is also provided to lower the runtime.
- We present the guidelines for placement, routing, and pin assignment that maintain the minimized pin count derived by the stage and device assignments.

The experimental results show the efficiency of our algorithms for different bioassays (and with different device selections). As our method provides more flexible device count/type choices, we also achieve 55–57% pin-count reductions over the previous works [14,16], which is a very significant improvement and justifies the effectiveness of our design flow/algorithms.

The remainder of this paper is organized as follows. Section 2 analyzes the control-pin demand and gives the overall design flow. Sections 3 and 4 detail the ILP formulation and problem reduction for stage assignment and device assignment, respectively. Section 5 provides the guidelines for follow-up placement, routing, and pin assignment. Finally, the experimental results and conclusions are given in Sections 6 and 7, respectively.
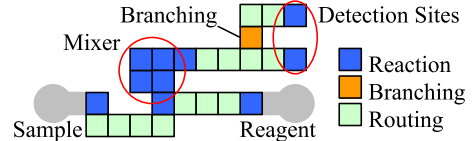
2. PIN DEMAND AND PROPOSED FLOW

Figure 2: Classification of electrodes.

We classify the demand of pins N_p into three categories:

$$N_p = P_{reaction} + P_{branching} + P_{routing}, \qquad (1)$$

where $P_{reaction}, P_{branching}$, and $P_{routing}$, are the the numbers of pins needed to control the electrodes for reactions, for output branchings from devices, and for routing, respectively. Figure 2 illustrates the classification of electrodes. Although the control signals of these three categories of pins may be further merged, we shall focus on minimizing the three terms before the merging.

Our design flow, shown in Figure 1(b), consists of the following three major stages:

1. *Stage assignment:* This stage minimizes $P_{reaction}$ by aligning reactions to execution stages and thus enables the synchronous control of the reactions. For the example bioassay in Figure 3(a), a possible stage assignment result is shown in Figure 3(b). Note that although we represent the generation and optical detection as black boxes in the simplified figure, we also assign corresponding stages to them. In the stage assignment, except for sharing the control pins, we also have to minimize the assay completion time to reduce the execution time overhead introduced during this stage. The constraints and considerations for stage assignment will be detailed in Section 3.

2. *Device assignment:* This stage minimizes $P_{branching}$ by matching the reactions to specific devices. Device assignment is important because different matchings can lead to different feeding relationships between devices and thus affects the number of independent control pins needed for the branchings between devices. A possible device assignment for Figure 3(b) is shown in Figure 3(c), and the device assignment problem will be addressed in Section 4.

3. *Guided placement, routing, and pin assignment:* This stage transforms the stage and device assignment result to actual placement and routing solutions. Following the proposed guideline with careful pin assignment can maintain $P_{routing}$ as a constant.
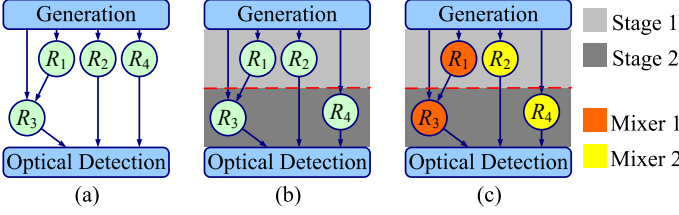
Figure 3: (a) An example bioassay with four mixing reactions. (b) The bioassay after stage assignment. (c) The bioassay after device assignment.

The differences of the proposed flow from the previous one are that the pin-count concerns are considered from the early stages to the end of the flow, and the scheduling problem is expanded to address two major sources of control pin demand.

3. STAGE ASSIGNMENT

This section first introduces the idea of synchronous reactions and the corresponding tradeoff between execution time and pin count. Next, we describe the stage assignment problem that attempts to minimize the assay completion time with the synchronously controlled reactions. Then we present the exact ILP formulation for the stage assignment. Finally, the corresponding solution space reduction schemes and an approximation for execution time are provided to speed up the runtime.

3.1 Advantage of Synchronous Reactions

The motivation for stage assignment is to reduce $P_{reaction}$ by synchronously controlling the reactions assigned to the same stage, while keeping the assay completion time minimized.
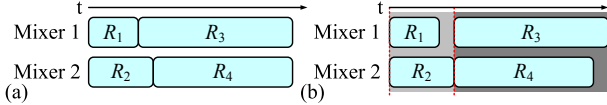

Figure 4: Comparison of asynchronous and synchronous control of two mixers with four reactions R_1, R_2, R_3, R_4: (a) Asynchronous: A shorter completion time is achieved, but the mixers do not share the control signals. (b) Synchronous: Mixers 1 and 2 share their control pins, but the completion time is longer.

The differences between asynchronous and synchronous control can be explained by Figure 4. In Figure 4(a), Mixers 1 and 2 are controlled separately and do not share a control pin; that is, the mixers can decide when to begin and cease their mixing reactions independently. We can see that R_3 can use Mixer 1 right after R_1 is completed, and thus the completion time is shorter. However, Mixers 1 and 2 cannot share a control pin in this way.

On the other hand, in Figure 4(b), Mixers 1 and 2 are controlled together, which means that the mixers must begin and cease their mixing reactions synchronously. Therefore, although R_1 and R_2 take different completion times, the synchronized control signals will cease the reactions only when the slower reaction ends. Thus, R_3 can begin only after R_2 is completed, at the same time as R_4. Despite the execution time overhead, in this case, Mixers 1 and 2 can share their control pins, which is more favorable for pin-constrained biochips.

3.2 The Stage Assignment Problem

Given a bioassay and a chip specification, stage assignment divides the reactions into a set of execution stages, and each stage is dedicated to a single category of reactions (e.g., generation of certain sample/reagent, mixing, optical detection, etc.). Besides, the working time of each stage is also determined in stage assignment. Following are the constraints for the stage assignment.

1. *Capacity constraints:* The number of reactions in a stage is upper-bounded by the number of the device belonging to the category of the stage.

2. *Uniqueness constraints:* A reaction exists in exactly one stage.
3. *Duration constraints:* The duration of a stage is the duration of the slowest reaction assigned to the stage.
4. *Sequence constraints:* Stages that belong to the same category are sorted and executed sequentially without overlapping.
5. *Precedence constraints:* If there are data dependencies between reactions described in the bioassay such as "reaction R_i must happen before reaction R_j," then the stage that includes R_j can begin only after the stage of R_i ends.

Note that the major differences from the previous scheduling or placement problems in [10, 11, 17] are the duration constraints and sequence constraints superimposed for synchronous control.

Different stage assignment results can lead to different assay completion times. For a given stage assignment, the minimum assay completion time can be calculated with the duration, sequence, and precedence constraints. In our stage assignment, the target is to find the stage assignment with the shortest assay completion time.

3.3 Problem Formulation for Stage Assignment

The stage assignment problem can be formulated as follows.
Given:

1. A set of N reactions $S_r = \{R_1, R_2, \ldots, R_N\}$.
2. A set of M devices categories $S_d = \{D_1, D_2, \ldots, D_M\}$.
3. $V_n \in [1, M], \forall R_n \in S_r$, where $V_n = m$ means that reaction R_n is executed by a device of category D_m.
4. $C_m, \forall D_m \in S_d$, where C_m is the number of the devices of category D_m.
5. $T_{n,m}, \forall R_n \in S_r, D_m \in S_d$, where $T_{n,m}$ is the time that R_n takes with device category m.
6. A set of precedence relationships $S_p \in 2^{S_r \times S_r}$, where $(R_{n_1}, R_{n_2}) \in S_p$ means that R_{n_1} must be completed before R_{n_2} begins.

Find:

1. A partitioning of S_r into independent subsets (stages) $S_{m,1}, S_{m,2}, \ldots, S_{m,I_m}, \forall m \in [1, M]$, where $S_{m,i}$ represents the i-th stage for D_m, and I_m represents the maximum number of stages for D_m.
2. Corresponding start time $B_{m,i}$ and finish time $E_{m,i}$ for these stages.

That minimize: $\max_{\forall m,i} E_{m,i}$
Under the following constraints:

1. *Capacity constraints:* $|S_{m,i}| \le C_m, \forall (m, i)$,
2. *Uniqueness constraints:*
   (a) $\exists (m, i), R_n \in S_{m,i}, \forall R_n \in S_r$, and
   (b) $S_{m_1, i_1} \cap S_{m_2, i_2} = \emptyset, \forall (m_1, i_1) \ne (m_2, i_2)$,
3. *Duration constraints:* $E_{m,i} - B_{m,i} \ge T_{n,m}, \forall R_n \in S_{m,i}$,
4. *Sequence constraints:* $E_{m,i_1} \le B_{m,i_2}, \forall m, i_1 < i_2$,
5. *Precedence constraints:* $E_{m_1,i_1} \le B_{m_2,i_2}, \forall (R_{n_1}, R_{n_2}) \in S_p, R_{n_1} \in S_{m_1,i_1}, R_{n_2} \in S_{m_2,i_2}$.

3.4 ILP Formulation for Stage Assignment

Denote the occurrence of $R_n \in S_{m,i}$ as $g_{n,m,i}$, and follow the notion of the problem formulation, then the stage assignment problem that minimizes assay completion time T_c can be formulated as follows.

Minimize T_c
Subject to

1. *Capacity constraints:*
$$\sum_{\forall n} g_{n,m,i} \le C_m, \forall (m, i). \qquad (2)$$

2. *Uniqueness constraints:*
$$\sum_{\forall (m,i)} g_{n,m,i} = 1, \forall n. \qquad (3)$$

3. *Duration constraints:*
$$E_{m,i} - B_{m,i} - T_{n,m} g_{n,m,i} \ge 0, \forall n, m, i. \qquad (4)$$

4. *Sequence constraints:*
$$E_{m,i_1} \le B_{m,i_2}, \forall m, i_1 < i_2. \qquad (5)$$

5. *Precedence constraints:*
$$E_{m_1,i_1} - B_{m_2,i_2} + T_{max}(g_{n_1,m_1,i_1} + g_{n_2,m_2,i_2} - 2) \le 0,$$
$$\forall (R_{n_1}, R_{n_2}) \in S_p, m_1, i_1, m_2, i_2. \quad (6)$$

6. *Assay start and finish:*
$$0 \le B_{m,i} \le E_{m,i} \le T_c, \forall (m,i). \quad (7)$$

In Inequality (6), T_{max} is a large number used to formulate the *AND* logic. It can also be set to a upper bound of the minimum assay completion time obtained by greedy assignment.

3.5 Solution Space Reduction

The naive formulation shown in the previous subsection is complicated and not efficient to be solved without reduction. The number of variables and constraints are $O(NMI_a)$ and $O(|S_p|M^2I_a^2 + NMI_a)$, where I_a is the max number of stages among the device categories, which can be $O(N)$ without careful bounding.

3.5.1 Reaction Category Mapping

To reduce the problem size, we first note that a reaction is mapped to a specific category of devices, and thus we can eliminate redundant variables and constraints. That is, all $g_{n,m,i}$ with $m \ne V_n$ can be removed, and thus all related constraints can be simplified or even abandoned. Therefore, the numbers of variables and constraints become $O(NI_a)$ and $O(|S_p|I_a^2 + NI_a)$, respectively. Note that a pair in S_p actually indicates two corresponding reactions and thus two specific device categories, so we can drop the M^2 from $|S_p|M^2I_a^2$.

3.5.2 Upper Bound for the Stage Number

Then we bound the number of stages used by each device category. The advantage is twofold. First, the effect of the I_a term on the number of variables and constraints can be effectively reduced when the number of stages is tightly bounded. On the other hand, providing a bound for the stage count can restrict the permutations of identical solutions. The exact formulas for the bound will not be presented in this paper due to the page limit.

3.5.3 Lower Bound for Assay Completion Time

Finally, it is helpful to add a lower bound for the assay completion time into the ILP formulation to speed up the runtime of the ILP solver. Sometimes for the case that the assay time is dominated by a few critical paths, the ILP solver may take more time to search for permutations of cells on non-critical paths even when an optimal solution is found. To avoid this situation, we can provide a lower bound by calculating the length values of all paths in advance and use the longest path to bound the assay completion time.

THEOREM 1. *The stage assignment problem can be solved optimally with the ILP formulation in Section 3.4 and the reduction in Section 3.5.*

3.6 Approximation for Assay Completion Time

Even with the reductions in the previous subsection, the formulation is still difficult for ILP solvers. One of the most difficult parts is the precedence constraints around $B_{m,i}$ and $E_{m,i}$ in Equation (6), in which the integer values are conditionally constrained.

To further speed up the ILP solution, we propose to apply ILP only for the stage assignment of mixing reactions, while the stage assignment of the other reactions (i.e., generation, optical detection) are implied by the related mixing reactions called *priors*. By doing so, we can avoid the formulation of the inter-category precedence constraints. Therefore, we only have to formulate the precedence constraints among mixing. To make sure that the minimum assay completion time obtained by the stage assignment of mixing can be used as a good approximate for the minimum execution time for the overall assay, additional constraints must be set on the distribution of the priors. However, the exact constraints for the distribution will not be presented in this paper due to the page limit.

4. DEVICE ASSIGNMENT

This section first explains how device assignment can affect the number of control pins needed for output branchings from devices. Next, the problem description and formulation of device assignment is introduced. Then an exact ILP formulation for the problem is presented. And finally, effective solution-space reduction schemes are proposed to speed up the process.

4.1 Effect of Device Permutation

While $P_{reaction}$ is handled in stage assignment, $P_{branching}$ is to be reduced in device assignment. To illustrate how device assignment can affect the number of branchings, Figure 5(a) shows part of a bioassay, and two possible device assignments for it are shown in Figures 5(b) and (c). In (b), three paths {Mixer 1→ Mixer 1, Mixer 1→ Mixer 2, Mixer 2→ Mixer 1} are used, while in (c), only two paths {Mixer 1→ Mixer 1, Mixer 2→ Mixer 1} are used and thus potentially fewer electrodes are needed for controlling the branchings. Generally, device assignment can affect the routing complexity, the potential pin count, and the number of electrodes used.
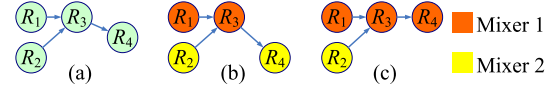

Figure 5: The effect of device assignment. (b)(c) are device assignment results for (a) with different numbers of branchings.

4.2 The Device Assignment Problem

Given a stage assignment result, device assignment decides, for each stage $S_{m,i}$, a 1-1 mapping from the reactions in $S_{m,i}$ to the devices of category D_m, which minimizes the number of branchings introduced by the dependencies in S_p. Note that the sets are obtained by inserting necessary buffers to a direct stage assignment because some droplet may have to wait for a few stages before they are used, and thus the sets may be different from the setup in stage assignment by a buffer category.

4.3 Problem Formulation for Device Assignment

The device assignment problem can be formulated as follows.

Given: $S_r, S_d, V_n, C_m, S_p, S_{m,i}$ same as that from stage assignment after buffer insertion.

Find: $x_{n,z}, z \in [1, C_{V_n}], \forall R_n \in S_r$, where $x_{n,z}$ is the occurrence of the event that R_n is assigned to the z-th device of the category.

That minimize: $\sum_{\forall m_1, z_1, m_2, z_2} p_{m_1, z_1, m_2, z_2}$, where p_{m_1, z_1, m_2, z_2} denotes the existence of a path from the z_1-th device of D_{m_1} to the z_2-th device of D_{m_2}.

Under the following constraints:
1. *1-1 mapping:* For any stage $S_{m,i}$,
$$OR(x_{n,1}, x_{n,2}, \ldots, x_{n,C_m}) = 1, \forall R_n \in S_{m,i}.$$

$$AND(x_{n,z_1}, x_{n,z_2}) = 0, \forall R_n \in S_{m,i}, z_1, z_2 \in [1, C_m], z_1 \ne z_2.$$

$$AND(x_{n_1,z}, x_{n_2,z}) = 0, \forall R_{n_1}, R_{n_2} \in S_{m,i}, n_1 \ne n_2, z \in [1, C_m].$$

2. *Path usage:* For any $m_1 \ne m_2, z_1 \in [1, C_{m_1}], z_2 \in [1, C_{m_2}]$,
$$p_{m_1, z_1, m_2, z_2} = OR_{\substack{\forall (R_{n_1}, R_{n_2}) \in S_p, \\ V_{n_1} = m_1, V_{n_2} = m_2}} \left(AND(x_{n_1,z_1}, x_{n_2,z_2})\right).$$

4.4 ILP Formulation for Device Assignment

An exact ILP formulation for the device assignment problem can be written as follows.

Minimize $\sum_{\forall m_1, z_1, m_2, z_2} p_{m_1, z_1, m_2, z_2}.$

Subject to
1. *1-1 mapping:* For any stage $S_{m,i}$,
$$\sum_{\forall z \in [1, C_m]} x_{n,z} = 1, \forall R_n \in S_{m,i}. \quad (8)$$

$$\sum_{\forall R_n \in S_{m,i}} x_{n,z} \le 1, \forall z \in [1, C_m]. \quad (9)$$

2. *Path usage:* For any $m_1 \neq m_2, z_1 \in [1, C_{m_1}], z_2 \in [1, C_{m_2}],$

$$x_{n_1,z_1} + x_{n_2,z_2} - p_{m_1,z_1,m_2,z_2} \leq 1,$$
$$\forall (R_{n_1}, R_{n_2}) \in S_p, V_{n_1} = m_1, V_{n_2} = m_2. \quad (10)$$

4.5 Solution-Space Reduction

The above formulation involves $O(N+M^2)$ variables and $O(N+M^2+|S_p|)$ constraints. Since the generation reactions can only precede mixing reactions, and the optical reactions can only follow mixing reactions in practical bioassays, we can cut the $O(M^2)$ feeding relationships between devices down to $O(M)$; thus the numbers of variables and constraints become $O(N + M)$ and $O(N + M + |S_p|)$, respectively. We further propose the following solution-space reduction schemes for the ILP formulation.

4.5.1 Permutation Restriction

First, we eliminate unnecessary permutations of the optimum solutions by fixing, for each device category D_m, the device assignment of one of its stages with $|S_{m,i}| = C_m$. Note that this is always feasible because there must be at least one stage with $|S_{m,i}| = C_m$, or the C_m should be reduced because the bioassay never uses all of the devices. By doing so, assume that the original solution space is U_d, the size of the new solution space after such a reduction is

$$|U_d'| = \prod_{\forall D_m \in S_d} \frac{1}{C_m!} |U_d|. \quad (11)$$

4.5.2 Redundancy Pruning

Then we can further reduce the solution space by removing the paths around *universal* peripheral reactions. Here we define a category of peripheral devices D_u to be universal for mixing devices D_m if and only if it is fed by all mixers of some mixing stage. In other words, each mixer must has a path to an optical detector of D_u, and no matter how the priors are permuted, we need exactly C_m paths for them in the best case. Therefore, we can remove the constraints about these paths. Similarly, this method can be applied to generation categories.

Finally, we can iteratively remove *uniform* mixing stages and related paths from the problem. A stage is uniform if and only if it has one of the following properties.

1. All output paths from the stage are removed and the input devices for all reactions in the stage are identical.

2. All input paths to the stage are removed and the output devices for all reactions in the stage are identical.

Note that the process can be iterative because the removal of one stage could make a consecutive stage uniform.

The universal peripherals and uniform mixing stages are removed during the ILP optimization, and their optimal device assignment can be greedily decided in the reverse order of their removal.

THEOREM 2. *The device assignment problem can be solved optimally with the ILP formulation in Subsection 4.4 and the reduction in Subsection 4.5.*

5. PLACEMENT, ROUTING, PIN ASSIGNMENT

This section introduces the general guidelines for pin-count aware placement and routing. Following the proposed guideline with careful pin assignment can keep the low pin-count obtained by the proposed stage assignment and device assignment algorithms and keep $P_{routing}$ as a constant. Due to the page limit, the pin assignment method will not be covered.

5.1 Pin-Count Saving Guidelines

The major reason that the previous placement and routing methods for direct-addressing biochips do not work appropriately for the design of pin-constrained biochips is that the previous methods only preserve segregation (guarding) cells between electrodes that work during overlapping time periods; in contrast, for pin-count reduction, we also have to provide segregation cells between two electrodes that work in separate time spans. For example, electrodes E_1 and E_2 in Figure 6 are turned on for different routing paths in separate time slots, but they still cannot be controlled by the same pin because they are neighbors.
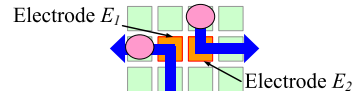

Figure 6: Electrodes used in different time spans that can prevent control pin merging.

Regarding this, we propose the following placement and routing guidelines.

1. The placement and routing for the entire bioassay should be decided simultaneously on a 2D plane.

2. Routing paths should not touch each other, except for necessary crossings or branchings.

3. Segregation (guarding) electrodes should be placed between devices and devices. If we wish to merge the control signals of reactions and routing paths, guarding electrodes are also needed between devices and routing paths.

Note that Items 1 and 2 are not addressed in previous placement and routing works, while similar constraints for Item 3 have been formulated to prevent unexpected merging between droplets [15, 18].

6. EXPERIMENTAL RESULTS

We implemented the proposed design flow in the C++ language with glpk [1] as our ILP solver. All of our experiments were conducted on a Linux machine with two 2.8 GHz AMD-64 CPUs and 8 GB memory.

6.1 Stage Assignment

Two practical bioassays, the *in_vitro* diagnostics used in [10] and the colorimetric protein assay [9], were used to evaluate our stage assignment algorithm. The experimental results for stage assignment are shown in Table 1. Designs "in_vitro 1," "in_vitro 2," and "in_vitro 3" are in_vitro diagnostics with different numbers of samples and reagents and thus different numbers of nodes (reactions) as shown in the table, and Design "protein" is the colorimetric protein assay. For each design, we conducted stage assignment with different mixing device choices to cover both sparse and dense cases, which affect how we approximate the peripheral times. Also, we tested a mini in_vitro diagnostic design "simple" with only 16 nodes just to illustrate the complexity growth of the exact ILP solution.

The runtimes of the ILP solver for both the basic formulation and the formulation after the solution-space reduction and assay time approximation are shown in Table 1. We can see that with the basic formulation, it takes more than 1 day to solve all the cases except for "simple" and "in_vitro 3" with the slowest devices. Note that even for the same design, the choice of devices can greatly affect the runtime. On the other hand, after the solution-space reduction and assay time approximation, the ILP solver took less than one second for all cases. The quality of the assay time approximation is also shown in the table; we can see that the error is smaller than 5% for all cases, with more than half the cases achieving zero errors.

6.2 Device Assignment

We then performed device assignment with the four practical bioassays with the stage assignment results in the previous experiment as reported in Table 2. We can see that the solution-space reduction schemes effectively reduce the number of nodes being formulated by 25% to 84%, and for all cases the optimum device assignment is obtained in less than 0.1 seconds. The results show the effectiveness and efficiency of our device assignment algorithm and the reduction scheme.

6.3 Placement and Routing Quality

With the stage and device assignment results produced by our algorithms, we also generated placement, routing, and pin assignment results for the three designs considered in [14, 16]. Table 3

Table 1: Experimental results for stage assignment.

Design	#nodes	Device Choice	Runtime (sec)		Assay Completion Time		
			Exact	Approx.	Exact	Approx.	Error (%)
simple	16	Fast	0.300	0.032	20	20	0
in_vitro 3	36	Slow	25369.99	0.052	44	46	4.55
		Fast	>1 day	0.036	40	40	0
		Medium	>1 day	0.044	41	43	4.88
in_vitro 2	48	Slow	>1 day	0.044	44	46	4.55
		Fast	>1 day	0.036	40	41	2.50
		Medium	>1 day	0.048	41	43	4.88
in_vitro 1	64	Slow	>1 day	0.540	56	56	0
		Fast	>1 day	0.164	52	52	0
		Medium	>1 day	0.440	53	55	3.77
protein	103	Slow	>1 day	0.116	161	161	0
		Fast	>1 day	0.092	105	105	0
		Medium	>1 day	0.100	121	121	0

Table 2: Experimental results for device assignment.

Design	#nodes		Runtime(sec)
	Original	Reduced	
in_vitro 3	36	27	0.016
in_vitro 2	48	36	0.016
in_vitro 1	64	48	0.020
protein	103	16	0.036

gives the results and comparisons. The results show that our design flow reduces 55–57% of the control pins, which justifies the effectiveness of our design flow and algorithms.

Table 3: Experimental results for placement, routing, and pin assignment.

Design	#control pins		
	[14, 16]	Ours	% improved
multiplexed	25	11	56.00
PCR	14	6	57.14
protein	27	12	55.56

Our result for the 2×2 multiplexed bioassay is shown in Figure 7. In Figure 7(a), a result without control signal merging is illustrated. We can see that even without control signal merging, only 16 pins are required for this bioassay with our algorithms. Pins p, q, r, s, t are shared by the control of the two mixers, while y, z are used to synchronously move the droplets to the two detection sites, and to disposal, respectively. And v, w, x are used to realize the dispensing of the sample/reagent droplets. Pins $1, 2, 3, 4, c$ are used for routing and crossings; b is used to control two branchings.

Figure 7(b) shows the final result that we report in Table 3. The pin count is further reduced from that in Figure 7(a) by merging pins $1, 2, 3, 4$ with p, q, r, s, and t with z. Consequently, only 11 control pins are needed while [14,16] need to use 25 control pins to realize the same bioassay. Note that we also manage to control two mixers simultaneously and lower the time to complete the bioassay for this design.
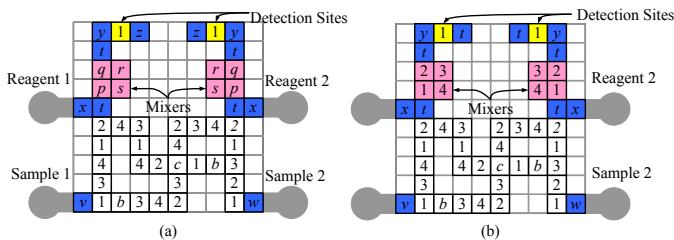

Figure 7: Placement, routing, and pin assignment for 2×2 multiplexed bioassay. (a) Without control signal merging. (b) With control signal merging.

7. CONCLUSIONS

We have presented in this work a pin-count aware design flow for pin-constrained biochips. ILP-based algorithms have been proposed for the stage and device assignments with effective solution-space reductions. Guidelines for placement, routing, and pin-assignment that reduce the pin count have also been discussed.

Experimental results have shown the efficiency of our algorithms and a 55–57% pin-count reduction over the previous flow that only post-processes the placement and routing results.

8. REFERENCES

[1] http://www.gnu.org/software/glpk/.

[2] K. F. Böhringer. Modeling and controlling parallel tasks in droplet-based microfluidic systems. *TCAD*, 25(2):334–344, 2006.

[3] M. Cho and D. Z. Pan. A high-performance droplet router for digital microfluidic biochips. In *ISPD*, pages 200–206, April 2008.

[4] J. Ding, K. Chakrabarty, and R. B. Fair. Scheduling of microfluidic operations for reconfigurable two-dimensional electrowetting arrays. *TCAD*, 20:1463–1468, 2001.

[5] R. B. Fair, V. Srinivasan, H. Ren, P. Paik, V. Pamula, and M. Pollack. Electrowetting-based on-chip sample processing for integrated microfluidics. In *IEDM*, pages 32.5.1–32.5.4, 2003.

[6] E. J. Griffith, S. Akella, and M. K. Goldberg. Performance characterization of a reconfigurable planar-array digital microfluidic system. *TCAD*, 25(2):435–357, 2006.

[7] A. J. Ricketts, K. Irick, N. Vijaykrishnan, and M. J. Irwin. Prtiority scheduling in digital microfluidics-based biochips. In *DATE*, pages 329–334, 2006.

[8] V. Srinivasan, V. Pamula, and R. Fair. An integrated digital microfluidic lab-on-a-chip for clinical diagnostics on human physiological fluids. *LoC*, 4:310–315, 2004.

[9] V. Srinivasan, V. Pamula, P. Paik, and R. Fair. Protein stamping for maldi mass spectrometry using an electrowetting-based microfluidic platform. In *Proceeding of the International Society for Optical Engineering*, pages 26–32, 2004.

[10] F. Su and K. Chakrabarty. Architectural-level synthesis of digital microfluidics-based biochips. In *ICCAD*, pages 223–228, 2004.

[11] F. Su and K. Chakrabarty. Unified high-level synthesis and module placement for defect-tolerant microfluidic biochips. In *DAC*, pages 825–830, June 2005.

[12] F. Su, K. Chakrabarty, and R. B. Fair. Microfluidic-based biochips: Technology issues, implementation platforms, and design-automation challenges. *TCAD*, 25(4):211–223, 2006.

[13] F. Su, W. Hwang, and K. Chakrabarty. Droplet routing in the synthesis of digital microfluidic biochips. In *DATE*, pages 323–328, 2006.

[14] T. Xu and K. Chakrabarty. Droplet-trace-based array partitioning and a pin assignment algorithm for the automated design of digital microfluidic biochips. In *CODES+ISSS*, pages 112–117, Oct. 2006.

[15] T. Xu and K. Chakrabarty. Integrated droplet routing in the synthesis of microfluidic biochips. In *DAC*, pages 948–953, June 2007.

[16] T. Xu and K. Chakrabarty. Broadcast electrode-addressing for pin-constrained multi-functional digital microfluidic biochips. In *DAC*, pages 173–178, June 2008.

[17] P.-H. Yuh, C.-L. Yang, and Y.-W. Chang. BioRoute: A network-flow based routing algorithm for digital microfluidic biochips. In *ICCAD*, pages 752–757, Nov. 2007.

[18] P.-H. Yuh, C.-L. Yang, and Y.-W. Chang. Placement of defect-tolerant digital microfluidic biochips using the T-tree formulation. *JETC*, 3(3):13:1–13:32, Nov. 2007.