PathDriver+: Enhanced Path-Driven Architecture Design for Flow-Based Microfluidic Biochips

Xing Huang, Youlin Pan, Grace Li Zhang, Bing Li, *Senior Member, IEEE*,
Wenzhong Guo, *Member, IEEE*, Tsung-Yi Ho, *Senior Member, IEEE*,
and Ulf Schlichtmann, *Senior Member, IEEE*

***Abstract*—Continuous-flow microfluidic biochips have attracted high research interest over the past years. Inside such a chip, fluid samples of milliliter volumes are efficiently transported between devices (e.g., mixers, heaters, etc.) to automatically perform various laboratory procedures in biology and biochemistry. Each transportation task, however, requires an exclusive flow path composed of multiple contiguous microchannels during its execution period. Excess/waste fluids, in the meantime, should be discarded by independent flow paths connected to waste ports. All these paths are etched in a very tiny chip area using multilayer soft lithography and driven by flow ports connecting with external pressure sources, forming a highly integrated chip architecture that determines the final performance of biochips. In this article, we propose a new and practical design flow called PathDriver+ (PD+) for the architecture design of microfluidic biochips, integrating the actual fluid manipulations into both high-level synthesis and physical design, which has never been considered in prior work. With this design flow, highly efficient chip architectures with a flow-path network that enables the actual fluid transportation and removal can be constructed automatically. Meanwhile, fluid volume management between devices and flow-path minimization are realized for the first time, thus, ensuring the correctness of assay outcomes while reducing the complexity of chip architectures. Additionally, diagonal channel routing is implemented to fundamentally improve the chip performance. The tradeoff between the numbers of channel intersections and fluidic ports is evaluated to further reduce the fabrication cost of biochips. The experimental results on multiple benchmarks confirm that the proposed design flow leads to high assay execution efficiency and low overall chip cost.**

***Index Terms*—Architecture design, diagonal routing, flow-path planning, fluid manipulation, fluidic port, microfluidics, volume management.**

I. Introduction

Advances in continuous-flow microfluidics technology have led to the emergence of "lab-on-a-chip" devices for automating laboratory procedures in biology and biochemistry. On such a micrometerscale platform, various biochemical applications, such as nanoparticle synthesis [2], drug discovery [3], etc., can be performed automatically with the advantages of high efficiency and high precision [4].

Structurally, a biochip is composed of two elastomer layers as shown in Fig. 1(a) [5], [6], where each layer consists of a set of microscale channels (note that there is another class of continuous-flow biochips that comprise only a single layer [7]). Microchannels in the flow layer, also called flow channels, are connected to external pressure sources through flow ports to transport samples and reagents. Microchannels in the control layer, also called control channels, are connected to control ports to conduct air pressure, so that the membrane in the overlapping area of the two channels can be pushed down into the flow channel to block the transportation of fluids, essentially forming a valve as illustrated in Fig. 1(b).

With valves as the basic control units, complex microfluidic devices, such as multiplexers and mixers, can be constructed [8], and biochemical applications can be executed automatically with a precustomized assay plan [9]. Fig. 1(c) shows a part of the layout of a cell-culturing biochip, where the reaction loop at the center is a mixer constructed with nine valves. Valves a_1–a_6 are used for directing the loading of fluid samples. For example, by closing a_5–a_6 and opening other valves, a sample can be loaded into the upper half of the mixer. Once the loading of samples is completed, the mixer can be sealed by closing a_1 and a_2 and the peristaltic valves a_7–a_9 are actuated sequentially to force the liquid inside to rotate.

When executing bioassays on a biochip, fluid samples and reagents need to be transported between devices to execute various biochemical operations. Each transportation task, however, requires an exclusive flow path during its execution period. Specifically, to drive the movement of fluid flow, a flow port that connects to an external pressure source should be placed at the start of the path to conduct thrust. Furthermore,

Manuscript received April 4, 2021; revised July 5, 2021; accepted August 2, 2021. Date of publication August 10, 2021; date of current version June 20, 2022. This work was supported in part by the Deutsche Forschungsgemeinschaft (DFG) through TUM International Graduate School of Science and Engineering (IGSSE); in part by the Natural Science Foundation of Fujian Province under Grant 2018J07005; and in part by the Humboldt Research Fellowship from the Alexander von Humboldt Foundation. A preliminary version of this article was published in the Proceedings of IEEE/ACM International Conference on Computer-Aided Design (ICCAD) [1]. This article was recommended by Associate Editor S. Chakraborty. *(Corresponding author: Grace Li Zhang.)*

Xing Huang, Grace Li Zhang, Bing Li, and Ulf Schlichtmann are with the Chair of Electronic Design Automation, Technical University of Munich, 80333 Munich, Germany (e-mail: xing.huang@tum.de; grace-li.zhang@tum.de; b.li@tum.de; ulf.schlichtmann@tum.de).

Youlin Pan and Wenzhong Guo are with the College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350116, Fujian, China (e-mail: panyoulin@163.com; guowenzhong@fzu.edu.cn).

Tsung-Yi Ho is with the Department of Computer Science, National Tsing Hua University, Hsinchu 30013, Taiwan (e-mail: tyho@cs.nthu.edu.tw).

Digital Object Identifier 10.1109/TCAD.2021.3103832

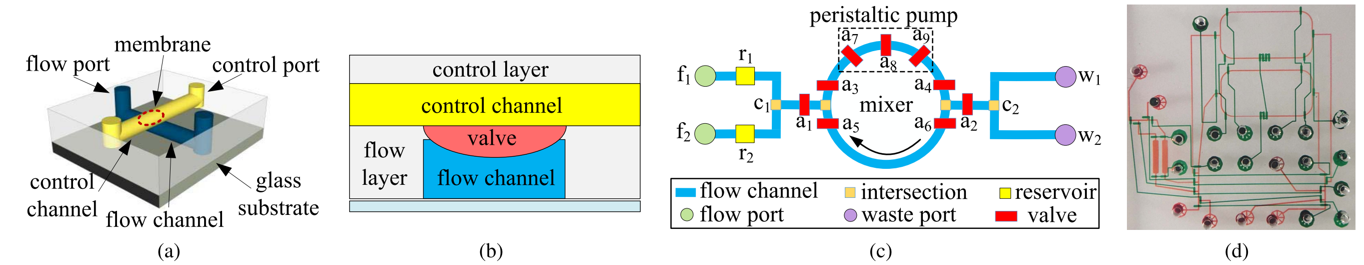


Fig. 1. (a) Schematic of a biochip [5]. (b) Front view of (a) [5]. (c) Part of the layout of a cell-culturing biochip [10]. (d) Fabricated chip with diagonal channels for kinase activity application [11].

since the initial status of a channel/device is not a vacuum but filled with air, to avoid channel blockage, a waste port needs to be connected to the end of the path to release air pressures [12]. As a consequence, the flow port, source and target devices, waste port, as well as flow channels connecting them form a complete flow path of the transportation task. For example, when transporting the reagent stored in reservoir r_1 to the mixer in Fig. 1(c), a valid flow path should be $\lfloor f_1 \rightarrow r_1 \rightarrow c_1 \rightarrow \text{mixer} \rightarrow c_2 \rightarrow w_1 \rfloor$.

Besides fluid transportation, fluids that are not needed anymore during the execution of bioassays should be removed to waste ports by independent flow paths. There are two categories: 1) excess fluids left in flow channels after loading samples into devices and 2) waste fluids that occupy devices after completing the execution of biochemical operations. Since fluid transportation and removal need to be performed frequently when executing bioassays [8], this leads to a large number of flow paths that need to be planned and constructed during chip design. These paths intersect with each other within a very small region, forming a highly integrated chip architecture that dominates the performance of the biochip.

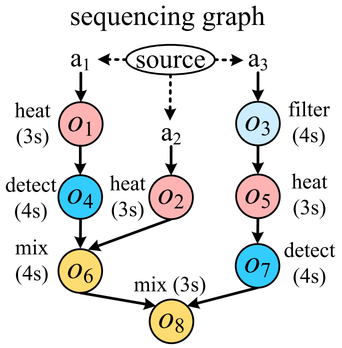
To fundamentally improve the performance of biochips while reducing the corresponding fabrication cost, the following issues should be considered during chip design: 1) the number of flow paths constructed on the chip should be minimized to reduce the complexity of chip architecture; 2) volumes of fluids transported between devices need to be controlled strictly, so that the correctness of assay outcomes can be ensured; 3) fluidic ports deployed in the chip, i.e., flow ports and waste ports, should be minimized. This is mainly because ports are holes punched on the elastomer material [4], biochips with many ports are more vulnerable to physical defects [13], [14]. Moreover, ports occupy extra chip area and need to be connected to external pressure sources, thus, increasing the overall chip cost; and 4) diagonal channels should be introduced so that the execution efficiency of biochips can be further improved [15]. Note that diagonal routing has already been supported by the current fabrication technology as illustrated in Fig. 1(d) [11], [15], where flow channels in orange color are implemented on the chip using the well-known X (octilinear) architecture—a routing model with 0°, ±45°, ±90°, and ±135° directions [16]–[21].

In recent years, a number of methods have been proposed to deal with the architecture design of biochips [6], [13], [22]–[26]. In [22], a top-down synthesis methodology is proposed to generate optimized biochip architectures while minimizing the total amount of valve-switching. In [23], physical design of biochips is investigated using a sequence-pair-based method, and it is formulated as a SAT problem in [24] to generate a close-to-optimal solution. In [13], a design flow is proposed to generate chip architectures under strictly constrained control ports. In [25], a timing-aware flow-channel routing method is proposed to optimize the timing behaviors of biochips. Moreover, in [6], a design flow considering wash optimization and channel storage is proposed to generate chip architectures with high storage efficiency. These methods, however, do not take the aforementioned issues into account and most of them assume that fluidic ports are implicitly connected to flow channels. The vast flexibility in the flow-path network construction has so far been ignored.

To fundamentally solve the problems above, we propose PathDriver+ (PD+), a practical path-driven design flow for microfluidic biochips. Our contributions are listed as follows.

1) We propose the first design flow that takes actual fluid manipulations into account for the design automation of microfluidic biochips. With the proposed method, practical biochip architectures with both high efficiency and low cost can be generated automatically.
2) We take both fluid volume management and excess/waste fluid removal into consideration, so that biochemical operations can be executed accurately and efficiently, thus, ensuring the correctness of assay outcomes with minimized completion time.
3) We take the tradeoff between the numbers of fluidic ports and channel intersections into account. In this way, chip cost and control complexity can be minimized simultaneously.
4) We integrate flow-path planning with diagonal channels into the physical design of biochips, so that the efficiency of fluid manipulations can be further improved, while the overall chip cost can be reduced significantly.
5) We propose an effective application remapping method to generate exact execution procedures of bioassays. Moreover, experimental results on five real-life bioassays and five synthetic benchmarks demonstrate the effectiveness of the proposed design flow.

The remainder of this article is organized as follows. In Section II, we briefly introduce the common design flow of microfluidic biochips and analyze the design challenges in path-driven architectural design. Section III formulates the



device library

resource	size (mm^2)	capacity
heater$_1$	15 × 10	35
heater$_2$	10 × 10	30
mixer	10 × 10	45
filter$_1$	5 × 5	20
filter$_2$	10 × 10	35
detector$_1$	10 × 10	20
detector$_2$	15 × 10	25

Fig. 2. Sequencing graph of a bioassay and the given device library.

problem addressed in this article. Section IV discusses the proposed design flow in detail. The experimental results are shown in Section V to demonstrate the effectiveness of PD+, and conclusions are drawn in Section VI.

II. Design Flow and Challenges

A. Design Flow of Microfluidic Biochips

The protocol of a bioassay is usually modeled as a directed acyclic sequencing graph $G(O, E)$. A vertex $o_i \in O$ represents a biochemical operation, e.g., mixing and heating, and is associated with a parameter indicating the corresponding execution time. An edge in G defines the dependency between operations, i.e., $e_{i,j} \in E$ represents that the output fluid of o_i is an input of o_j. For example, Fig. 2 shows the sequencing graph of a bioassay, which takes three fluid samples (a_1–a_3) as inputs and processes them with eight biochemical operations (o_1–o_8). To realize the execution of these operations, a device library D is provided, where each device $d_i \in D$ is represented by a rectilinear box with input/output port on its boundary [11], [13], [23], [24]. Moreover, each device is associated with a parameter indicating its capacity in milliliter. In Fig. 2, there are seven devices available for realizing the bioassay above.

Given the sequencing graph of a bioassay and the corresponding device library, the synthesis of microfluidic biochips is usually divided into two major stages: 1) high-level synthesis and 2) physical design.

Unlike the high-level synthesis performed in traditional very large-scale integration (VLSI) circuits, in microfluidics, this stage aims to realize the following goals: 1) find a binding function $\phi : O \rightarrow D$ such that each operation is bound to a specific device to execute and 2) generate a scheduling scheme such that all the operations are executed satisfying the given dependencies while minimizing the completion time of the bioassay. Fig. 3 shows a high-level synthesis result of the bioassay described in Fig. 2, where five devices are allocated to execute operations o_1–o_8 and the assay is completed in 19 s with nine transportation tasks (#$_1$–#$_9$).

Afterward, physical design is performed to assign the allocated devices to exact locations on the chip while establishing efficient connections among them. Correspondingly, Fig. 4 shows a chip layout generated by the existing design framework [13], [23], [24], where samples a_1–a_3 are injected into the chip through three flow ports f_1–f_3 and the outcome of the assay is collected by a reservoir r_1.

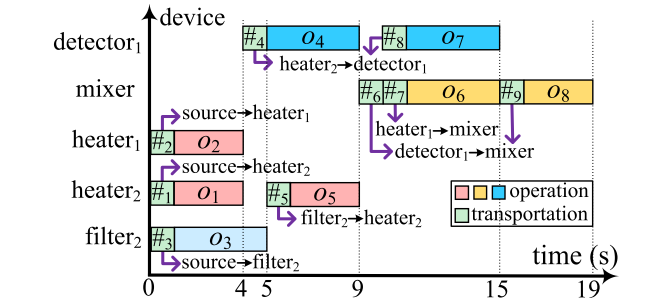


Fig. 3. Scheduling and binding scheme corresponding to the bioassay described in Fig. 2.

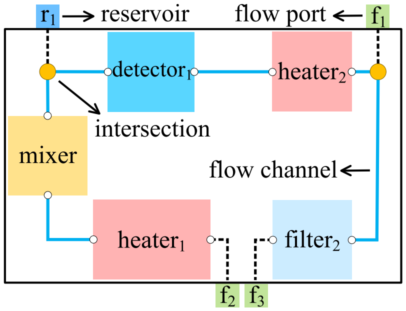


Fig. 4. Chip layout constructed for the scheduling described in Fig. 3 using the existing design framework.

Note that signal propagation in VLSI circuits is electronic [27]. In microfluidics, however, fluid samples are driven by the air pressure injected from an external pressure source and transported in flow channels with a relatively low velocity (typically 10 mm/s [28]). Consequently, transportation delays occur extensively in biochips during the execution of bioassays. Since the final chip layout has not been determined during the high-level synthesis of biochips, these delays are usually set to a predefined constant [6], [9], [22], [29]. For example, the delay of the transportation tasks specified in Fig. 2 is set to 1 s. Accordingly, after completing the physical design, bioassays need to be remapped to the generated chip architectures, so that real transportation delays of fluids can be incorporated into the scheduling, thus, generating exact execution procedures of bioassays. Note that application remapping needs to be performed only once without iteration.

B. Motivation and Design Challenges

During the architectural design above, several key parameters, such as completion time of the assay, total length of channels, and the number of intersections, etc., have been optimized systematically. Flow channels for all the transportation tasks (i.e., #$_1$–#$_9$) specified in Fig. 3 have been constructed during physical design. The chip layout shown in Fig. 4, however, is still not practical since the following key problems are ignored in the existing design framework: 1) fluid volume management between devices; 2) removal of excess/waste fluids; and 3) flow-path planning that enables the actual manipulation of fluid transportation/removal. Furthermore, chip performance

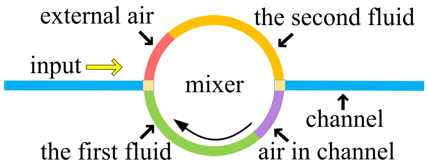
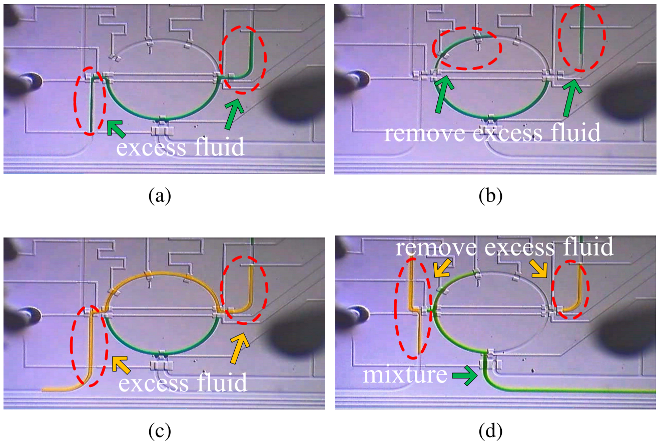


Fig. 5. Failure of fluid volume control when transporting fluids to a mixer.



(a) (b) (c) (d)

Fig. 6. Snapshots of the mixing operation using a fabricated biochip [30]. (a) Step 1: loading the first fluid flow into the lower half of the mixer. (b) Step 2: removing the excess fluids left behind by the first fluid. (c) Step 3: loading the second fluid flow into the upper half of the mixer and starting the mixing operation. (d) Step 4: recovering the resulting mixture and removing the corresponding excess fluids.

and chip cost can be further optimized by taking the following issues into consideration: 1) implementation of diagonal channels and 2) fluidic-port minimization.

1) Volume Management Between Devices: In microfluidic biochips, as mentioned before, the initial status of a flow channel/device is not a vacuum but filled with air [12]. When executing biochemical operations, the air already present should be completely pushed out of devices, so that the operations can be executed correctly while ensuring the purity of samples/reagents. On the other hand, since the movement of fluids in flow channels is driven by the air pressure injected from external pressure sources [30], the air should avoid entering into the target device during fluid transportation.

Consider the mixing operation shown in Fig. 5 as an example, where two fluid flows have been loaded into the mixer, respectively. However, due to insufficient fluid volumes, external air enters into the upper half of the mixer and a small amount of air still remains in the lower half of the mixer. As a consequence, the two fluids that need to be mixed are separated during the mixing process, leading to execution failure of the operation. In contrast, Fig. 6 shows the snapshots of fluid manipulation when the mixing operation is performed in a wet laboratory. In Fig. 6(a), the lower half of the mixer is filled with fluid flow when loading the first sample. After removing the excess fluids cached at the two ends of the mixer in Fig. 6(b), the second fluid is loaded into the mixer in a similar manner, as shown in Fig. 6(c). Because no extra air is kept inside the mixer, the mixing of two fluids is completed successfully, as shown in Fig. 6(d). Note that since an extra output port is added to the bottom of the mixer, excess fluids of the second fluid flow and the resulting mixture can be removed simultaneously in Fig. 6(d).

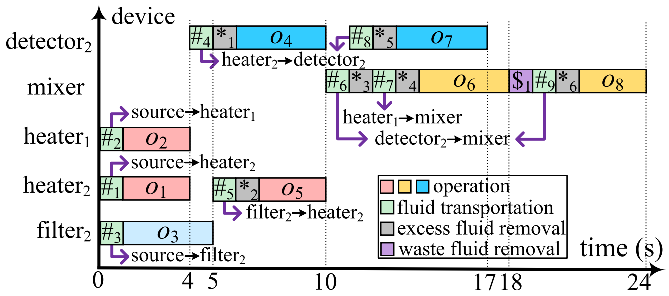


Fig. 7. Updated scheduling corresponding to Fig. 3 after considering fluid volume management.

To avoid execution failure of operations, volumes of input fluids should be controlled strictly according to the capacities of devices. More specifically, each input of a device should be bound by a minimum volume, so that air can be pushed out of the device while avoiding the entering of external air. For example, the minimum input volume of a detector corresponds directly to its capacity since a detecting operation has only one input. Moreover, for a rotary mixer with two inputs, the volume of each input should be at least half of its capacity. Accordingly, when binding operations of a bioassay to specific devices in the high-level synthesis stage, volume constraints between devices should be considered carefully based on the dependencies specified in the sequencing graph.

Based on the analysis above, let us revisit the scheduling described in Fig. 3, where o_4 and o_7 are bound to detector$_1$ and the resulting fluids of both operations need to be transported to mixer. Since the maximum output volume of detector$_1$ is only 20 ml, which is smaller than the minimum input volume of mixer (i.e., 22.5 ml), this results in an invalid binding scheme. In contrast, if o_4 and o_7 are bound to detector$_2$, the volume constraint between the two devices can be satisfied.

2) Removal of Excess/Waste Fluids: Besides volume management between devices, another key issue that has been ignored in the prior work is the removal of excess/waste fluids. After completing a fluid transportation task during the execution of a bioassay, as illustrated in Fig. 6, excess fluids cached at the two ends of a device should be removed from the chip. Additionally, fluids that are not needed anymore after completing the execution of operations should also be discarded. For example, since operations o_6 and o_8 are bound to the same mixer in Fig. 3, before starting the execution of o_8, one half of the resulting fluid of o_6 should be removed so that the output fluid of o_7 can be loaded into the mixer.

Correspondingly, Fig. 7 shows a modified scheduling corresponding to the bioassay described in Fig. 2, where both volume management and fluid removal have been considered. Note that since volumes of source samples (i.e., a_1–a_3 in Fig. 2) can be controlled precisely to be equal to the minimum input volumes of the corresponding devices, there is no excess fluid after completing transportation tasks $\#_1$–$\#_3$.

TABLE I
FLOW PATHS OF THE FLUID MANIPULATIONS IN THE CHIP LAYOUT SHOWN IN FIG. 8

flow paths of fluid transportation
$\#_4, \#_8$: $f_1 \rightarrow c_1 \rightarrow$ heater$_2 \rightarrow c_2 \rightarrow$ detector$_2 \rightarrow c_3 \rightarrow w_4$
$\#_5$: $f_3 \rightarrow$ filter$_2 \rightarrow c_5 \rightarrow c_1 \rightarrow$ heater$_2 \rightarrow c_2 \rightarrow w_1$
$\#_6, \#_9$: $f_5 \rightarrow c_2 \rightarrow$ detector$_2 \rightarrow c_3 \rightarrow$ mixer$\rightarrow c_4 \rightarrow w_2$
$\#_7$: $f_2 \rightarrow$ heater$_1 \rightarrow c_4 \rightarrow$ mixer$\rightarrow c_3 \rightarrow w_4$
flow paths of excess fluid removal
$*_1, *_5$: $f_4 \rightarrow s_1 \rightarrow c_3 \rightarrow w_4$; $f_4 \rightarrow s_2 \rightarrow c_2 \rightarrow w_1$
$*_2$: $f_5 \rightarrow s_3 \rightarrow c_2 \rightarrow w_1$; $f_5 \rightarrow s_4 \rightarrow c_1 \rightarrow c_5 \rightarrow w_3$
$*_3, *_4, *_6$: $f_6 \rightarrow s_5 \rightarrow c_3 \rightarrow w_4$; $f_6 \rightarrow s_6 \rightarrow c_4 \rightarrow w_2$
flow paths of waste fluid removal and external inputting
$\$_1$: $f_4 \rightarrow c_3 \rightarrow$ mixer$\rightarrow c_4 \rightarrow w_2$
$\#_1$: $f_1 \rightarrow c_1 \rightarrow$ heater$_2 \rightarrow c_2 \rightarrow w_1$
$\#_2$: $f_2 \rightarrow$ heater$_1 \rightarrow c_4 \rightarrow w_2$
$\#_3$: $f_3 \rightarrow$ filter$_2 \rightarrow c_5 \rightarrow w_3$

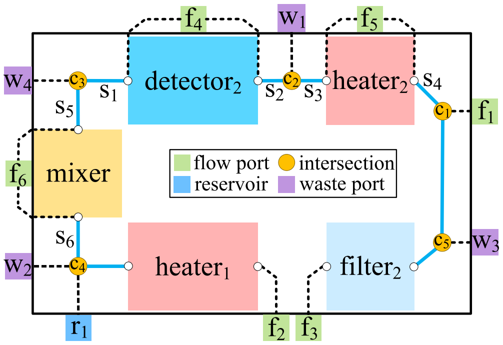


Fig. 8. Updated chip layout corresponding to Fig. 4 based on the scheduling described in Fig. 7, where diagonal channels are introduced to construct the complete flow paths of fluid manipulation tasks.

3) Flow-Path Planning: With a practical scheduling scheme, flow-path planning should be integrated into the physical design of biochips, so that the actual manipulations of fluid transportation and removal can be realized. Note that a valid flow path is not a simple connection of several channel segments. As discussed previously, flow ports that connect with external pressure sources should be deployed to provide impetus for fluid movement. Moreover, waste ports should be connected to the end of flow paths to release the air in flow channels while recovering the excess/waste fluids. As a result, a complete flow path should be composed of fluidic ports, device(s)/caching channels, as well as flow channels. Accordingly, the construction of flow paths during architectural design can be divided into three types.

1) *Type 1 (Fluid Transportation):* When performing a fluid transportation task between devices, the corresponding flow path should be ⌈flow port → source device → target device → waste port⌋.
2) *Type 2 (Excess Fluid Removal):* When removing excess fluids cached at the two ends of a device, the corresponding flow paths should be ⌈flow port → caching position → waste port⌋.
3) *Type 3 (Waste Fluid Removal and External Inputting):* When removing waste fluids from a device or inputting a sample/reagent from an external source, the corresponding flow path should be ⌈flow port → target device → waste port⌋.

The flow paths above should be planned and constructed systematically during the architectural design of biochips, so that all the specified fluid transportation/removal tasks can be completed efficiently. Furthermore, since flow-path planning directly determines the final chip architecture, chip costs such as channel length and chip performances such as execution efficiency, should also be optimized in this stage. To this end, diagonal channels can be implemented on the chip as discussed before. Accordingly, Fig. 8 shows an updated chip layout based on the scheduling described in Fig. 7, where flow paths of all the specified fluid manipulations have been constructed in the X architecture as shown in Table I. Note that since extra space has already been reserved inside the bounding boxes of devices [13], flow channels can be line touched with their boundaries.

Also, fluidic ports have been shared among flow paths in Fig. 8, e.g., port w_2 is shared by the flow paths of $\#_2$, $\#_6/\#_9$, $*_3/*_4/*_6$, and $\$_1$, etc. In this way, the overall chip cost can be reduced accordingly. Moreover, as mentioned before, since biochips with a large number of ports are more vulnerable to physical defects [13], [14], sharing ports can also prolong the lifetime of biochips. Note that sharing ports may introduce extra channel intersections and valves, thereby increasing the control complexity of biochips [13], [14]. Thus, port sharing and channel intersection should be considered jointly during flow-path planning, thereby achieving a tradeoff between the numbers of fluidic ports and intersections.

III. PROBLEM FORMULATION

Based on the analysis above, the system-architectural design problem considered in this article for continuous-flow microfluidic biochips can be formulated as follows.

Inputs:

1) A bioassay modeled as a sequencing graph $G(O, E)$.
2) A device library D indicating the number of available devices of each type and the capacity of each device.

Outputs:

1) An efficient binding and scheduling scheme:
   a) indicating an exact execution procedure of the bioassay;
   b) satisfying the volume constraints between allocated devices;
   c) considering both fluid transportation and waste/excess fluid removal.
2) An optimized biochip architecture:
   a) adopting diagonal channels;
   b) including the complete flow paths for all the fluid manipulation tasks specified in the scheduling.

Objectives:

1) Minimizing the completion time of the bioassay.
2) Minimizing the overall chip cost, including the following indicators.
   a) The number of flow paths constructed on the chip.
   b) The number of channel intersections (extra valves).

TABLE II
SYMBOLS FREQUENTLY USED IN PD+

- $b_{i,k}$: A 0-1 variable representing whether operation o_i in the sequencing graph is bound to a device d_k in D.
- $d(o_i)/out(o_i)$: The allocated device/the resulting fluid of operation o_i;
- $p_{j,i,k}$, k=1,2,3: 1) When a storage is needed, $p_{j,i,1}$ and $p_{j,i,2}$ represent the transportation tasks of $out(o_j)$ from $d(o_j)$ to a storage and from the storage to $d(o_i)$, respectively. 2) When storage is not needed, $p_{j,i,3}$ represents the transportation task of $out(o_j)$ from $d(o_j)$ to $d(o_i)$;
- $z_{i,h,k}$: A 0-1 variable representing whether operations o_i and o_h are bound to the same device d_k;
- $t^s_{p_{j,i,k}}/t^e_{p_{j,i,k}}$: The start time/end time of transportation task $p_{j,i,k}$;
- t^s_i/t^e_i: The start time/end time of operation o_i in the generated scheduling;
- $w_{j,i}$: Waste-fluid removal of $d(o_i)$ when transporting $out(o_j)$ to $d(o_i)$;
- $t^s_{w_{j,i}}/t^e_{w_{j,i}}$: The start time/end time of task $w_{j,i}$;
- $r_{j,i}$: Excess-fluid removal after transporting $out(o_j)$ to $d(o_i)$;
- $t^s_{r_{j,i}}/t^e_{r_{j,i}}$: The start time/end time of the task $r_{j,i}$;
- $u^{d_i}_{x,y}$: A 0-1 variable representing whether a cell (x,y) is occupied by device d_i;
- $u^{f_i}_{x,y}/u^{w_i}_{x,y}$: A 0-1 variable indicating whether a cell (x,y) is occupied by the flow port/the waste port of flow path l_i;
- $u^{n_j}_{x,y}$: A 0-1 variable representing whether a cell (x,y) is on the routing path of net n_j;
- $u^{e_{l_i}}_{x,y}$: A 0-1 variable representing whether a cell (x,y) is used for caching excess fluid e_{l_i};
- $S^c_{x,y}/S^p_{x,y}/S^g_{x,y}$: A 0-1 variable representing whether a cell (x,y) is occupied by a flow channel/fluidic port/channel intersection;
- $S^U_{x,y}/S^R_{x,y}/S^{RU}_{x,y}/S^{RD}_{x,y}$: A 0-1 variable representing whether a cell (x,y) has a upward/rightward/right upward/right downward routing channel;

c) The number of fluidic ports used in the chip.
d) The total length of flow channels.

IV. DETAILS OF THE PROPOSED DESIGN FLOW

In this section, we discuss the proposed PD+ in detail, which integrates the challenges discussed in Section II-B into both high-level synthesis and physical design. The proposed synthesis flow is formulated into integer linear programming (ILP) problems in the following and solved by a solver accordingly. Symbols that are frequently used in the proposed method are listed in Table II.

A. High-Level Synthesis With Volume Management, Fluid Removal, and Flow-Path Minimization

Given the sequencing graph $G(O,E)$ of a bioassay and a device library D, the target of high-level synthesis in PD+ is to generate a practical and optimized binding and scheduling scheme such that: 1) the completion time of the assay is minimized; 2) volume constraints between devices are satisfied; 3) excess/waste fluids are removed efficiently; and 4) the number of required flow paths is minimized.

To complete the execution of the assay, each operation in O should be bound to a specific device in D, which can be formulated as

$$\sum_{d_k \in D} b_{i,k} = 1 \quad \forall o_i \in O \tag{1}$$

where $b_{i,k}$ is a 0-1 variable representing whether operation o_i is bound to device d_k. For the sake of simplicity, hereinafter we denote the output fluid and the allocated device of an operation o_i by out(o_i) and $d(o_i)$, respectively.

Before starting the execution of o_i, for each dependency $e_{j,i} \in E$, out(o_j) should be transported from device $d(o_j)$ to $d(o_i)$. We use a 0-1 variable $c_{i,j}$ representing whether o_j is a child operation of o_i in the sequencing graph and assume that the resulting fluid of an operation is distributed equally to its child operations. Then, the volume constraint between $d(o_j)$ and $d(o_i)$ can be formulated as

$$\frac{\sum_{d_k \in D} b_{j,k} \times v_k}{\sum_{o_k \in O} c_{j,k}} \geq \frac{\sum_{d_k \in D} b_{i,k} \times v_k}{\sum_{o_k \in O} c_{k,i}} \quad \forall e_{j,i} \in E \tag{2}$$

where v_k is the capacity of device d_k.

When transporting out(o_j) to $d(o_i)$, if $d(o_i)$ is still occupied by another operation ($d(o_i) \neq d(o_j)$), out(o_j) should be cached temporarily in a storage, leading to two subtransportation tasks: 1) $p_{j,i,1}$: from $d(o_j)$ to the storage and 2) $p_{j,i,2}$: from the storage to $d(o_i)$. Note that o_i and o_j may be bound to the same device ($d(o_i) = d(o_j)$). In this case, after completing o_j, the device is allocated to execute another operation before o_i. We use variables t^s_i and t^e_i to represent the start time and end time of o_i, and the durations of the two transportation tasks above can be constrained as

$$t^e_{p_{j,i,l}} - t^s_{p_{j,i,l}} = \begin{cases} T_p, & \exists o_h \in O, \sum_{d_k \in D} z_{i,h,k} > 0 \quad \forall e_{j,i} \in E \\ & \wedge t^e_h < t^s_i \wedge t^e_h > t^e_j \qquad\qquad l = 1,2 \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where $t^s_{p_{j,i,l}}$ and $t^e_{p_{j,i,l}}$ represent the start time and end time of the tasks $p_{j,i,l}$ ($l = 1, 2$), respectively. $z_{i,h,k}$ is a 0-1 variable representing whether o_i and o_h are bound to device d_k. Since the layout of the chip remains undetermined during high-level synthesis, we use a constant T_p to represent the duration of a fluid transportation/removal task. Note that $p_{j,i,2}$ can only be started after the completion of $p_{j,i,1}$ and thus, can be constrained as

$$t^e_{p_{j,i,1}} \leq t^s_{p_{j,i,2}} \quad \forall e_{j,i} \in E. \tag{4}$$

In contrary, if $d(o_i)$ is not occupied when o_j is completed, there is only one transportation task, denoted by $p_{j,i,3}$, directly from $d(o_j)$ to $d(o_i)$. Note that if the two operations are bound to the same device, the transportation above can be avoided accordingly. Thus, the duration of $p_{j,i,3}$ can be constrained as

$$t^e_{p_{j,i,3}} - t^s_{p_{j,i,3}} = \begin{cases} T_p, & \sum_{d_k \in D} z_{i,j,k} \leq 0 \quad \forall e_{j,i} \in E. \\ 0, & \sum_{d_k \in D} z_{i,j,k} > 0. \end{cases} \tag{5}$$

Moreover, as discussed in Section II-B2 and Fig. 7, when transporting out(o_j) to $d(o_i)$, if o_i is a multiinput operation, e.g., a mixing operation, and $d(o_i)$ is filled with the resulting fluid of another father operation of o_i, a part of the fluid should first be removed from $d(o_i)$, leading to a waste-fluid removal task, denoted by $w_{j,i}$, whose duration can be constrained as

$$t^e_{w_{j,i}} - t^s_{w_{j,i}} = \begin{cases} T_p, & \exists e_{h,i} \in E, \qquad \sum_{d_k \in D} z_{i,h,k} > 0 \\ & \wedge t^e_{p_{h,i,1}} - t^s_{p_{h,i,1}} = 0 \quad \forall e_{j,i} \in E' \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

$$t^e_{w_{j,i}} \leq t^s_{p_{j,i,k}} \quad \forall e_{j,i} \in E, k = 2, 3 \tag{7}$$

where $t^s_{w_{j,i}}$ and $t^e_{w_{j,i}}$ represent the start time and end time of the task $w_{j,i}$, respectively.

After completing the transportation of out(o_j), if its volume is larger than the minimum input volume of $d(o_i)$, excess fluids cached at the two ends of $d(o_i)$ should be removed, leading to an excess fluid removal task, denoted by $r_{j,i}$, whose duration can be constrained as

$$t^e_{r_{j,i}} - t^s_{r_{j,i}} = \begin{cases} T_p, & \frac{\sum_{d_k \in D} b_{j,k} \times v_k}{\sum_{o_k \in O} c_{j,k}} > \frac{\sum_{d_k \in D} b_{i,k} \times v_k}{\sum_{o_k \in O} c_{k,i}} \\ & \wedge 1 - \sum_{d_k \in D} z_{i,j,k} + \rho_{j,i} > 0 \quad \forall e_{j,i} \in E \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

$$t^s_{r_{j,i}} \geq t^e_{p_{j,i,k}} \quad \forall e_{j,i} \in E, k = 2, 3 \tag{9}$$

where $t^s_{r_{j,i}}$ and $t^e_{r_{j,i}}$ represent the start time and end time of the task $r_{j,i}$ and $\rho_{j,i}$ is a 0-1 variable representing whether a storage is introduced in the transportation from $d(o_j)$ to $d(o_i)$. Moreover, if o_i is a multiinput operation, the next input should be started after $r_{j,i}$ is completed. Note that the input fluids of o_i should be loaded separately. Thus, we have the following constraints:

$$\begin{aligned} (1 - \lambda_{j,h,i})M + t^s_{p_{h,i,k}} &\geq t^e_{p_{j,i,k}} \\ \lambda_{j,h,i}M + t^s_{p_{j,i,k}} &\geq t^e_{p_{h,i,k}} \quad \forall e_{j,i}, e_{h,i} \in E, k = 2, 3 \end{aligned} \tag{10}$$

$$t^s_{p_{h,i,k}} \geq t^e_{r_{j,i}}, \ \exists e_{h,i} \in E \wedge t^s_{p_{h,i,k}} \geq t^e_{p_{j,i,k}}, k = 2, 3 \tag{11}$$

where $\lambda_{j,h,i}$ is a 0-1 variable representing the input order of out(o_j) and out(o_h) with respect to $d(o_i)$ and M is a very large constant to transform the two situations indicated by $\lambda_{j,h,i}$ into linear constraints.

After completing the transportation of all the input fluids of o_i, the execution of o_i can be started and should last long enough, constrained as

$$t^s_i \geq t^e_{p_{j,i,k}} \quad \forall e_{j,i} \in E, k = 2, 3 \tag{12}$$

$$t^s_i + t(o_i) \leq t^e_i \tag{13}$$

where $t(o_i)$ is the execution time of o_i specified in the sequencing graph.

Only when both the execution of o_i and the removal of corresponding excess fluids are completed, the resulting fluid can be transported to the next operation and thus, we have

$$t^s_{p_{i,h,k}} \geq t^e_i, \ t^s_{p_{i,h,k}} \geq t^e_{r_{j,i}} \quad \forall e_{j,i}, e_{i,h} \in E, k = 1, 3. \tag{14}$$

Similar to the situation when loading fluids into a device, if o_i is a multioutput operation, e.g., a splitting operation, the corresponding output fluids should be removed separately and thus, can be constrained as

$$\begin{aligned} (1 - \mu_{i,h,q})M + t^s_{p_{i,h,k}} &\geq t^e_{p_{i,q,k}} \quad \forall e_{i,h}, e_{i,q} \in E, k = 1, 3 \\ \mu_{i,h,q}M + t^s_{p_{i,q,k}} &\geq t^e_{p_{i,h,k}} \quad \forall e_{i,h}, e_{i,q} \in E, k = 1, 3 \end{aligned} \tag{15}$$

where $\mu_{i,h,q}$ is a 0-1 variable representing the output order of the resulting fluids of o_i.

Additionally, after completing the execution of o_i, if there exists another operation $o_q \in O$ that needs to be executed using the same device ($t^s_q \geq t^e_i$), it can only be started after removing the resulting fluid of o_i. Thus, we have the following constraint:

$$\left(1 - \sum_{d_k \in D} z_{i,q,k}\right)M + t^s_{p_{u,q,k_1}} \geq t^e_{p_{i,h,k_2}}$$
$$\forall e_{u,q}, e_{i,h} \in E, k_1 = 2, 3, k_2 = 1, 3. \tag{16}$$

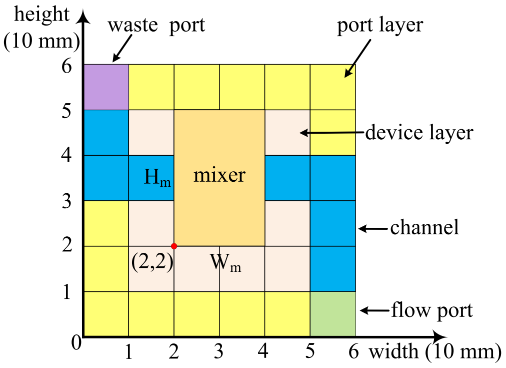


Fig. 9. Illustration of the virtual grid used in PD+.

After completing the execution of all the operations, the complete assay is finished. We use a variable T_{assay} to represent the completion time of the assay, which can be constrained as

$$t^e_i \leq T_{\text{assay}} \quad \forall o_i \in O. \tag{17}$$

Finally, since the duration of the fluid transportation/removal task corresponding to each flow path is set to T_p during high-level synthesis, an optimized binding and scheduling scheme minimizing both completion time of the assay and the number of flow paths can be determined by solving the following optimization problem:

$$\text{minimize} \ \ \alpha \times T_{\text{assay}} + \beta \times \left(\sum_{e_{j,i} \in E} \frac{1}{T_p} \left(\sum_{k=1}^{3} \left(t^e_{p_{j,i,k}} - t^s_{p_{j,i,k}} \right) \right. \right.$$
$$\left. \left. + 2 \times \left(t^e_{r_{j,i}} - t^s_{r_{j,i}} \right) + t^e_{w_{j,i}} - t^s_{w_{j,i}} \right) \right) \tag{18}$$

$$\text{subject to} \ \ (1)\text{–}(17) \tag{19}$$

where α and β are two weighting factors.

B. Physical Design With Flow-Path Planning

After completing the high-level synthesis, a set $D_a \subset D$ of allocated devices and a set L of flow paths that need to be constructed are obtained accordingly. In this section, physical design is performed to generate a chip architecture such that the previously generated scheduling can be realized while minimizing chip costs such as the total length of flow channels, the numbers of channel intersections and fluidic ports, etc.

To generate an optimized chip architecture, PD+ adopts a virtual grid R of size $W \times H$ to realize the placement of devices and the construction of flow paths as illustrated in Fig. 9. Each cell on the grid can be represented by the coordinates of its lower left corner. Moreover, the complete grid is divided into the following two regions.

1) *Port Layer:* The cells on the boundary of the grid are used to deploy fluidic ports and can be mathematically represented as $R_p = \{(x, y) | 0 \leq x < W, y = 0, H - 1\} \cup \{(x, y) | x = 0, W - 1, 0 \leq y < H\}$.

2) *Device Layer:* The cells surrounded by a port layer are used to deploy devices and can be mathematically represented as $R_d = \{(x, y)|0 < x < W - 1, 0 < y < H - 1\}$.

For example, in Fig. 9, a flow port and a waste port are placed at the lower right and upper left corners of the grid, respectively. The two ports are connected with a mixer placed at the center of the device layer using rectilinear channels, forming a complete flow path.

To ensure that a device is placed inside the device layer, we have the following constraints:

$$x_i > 0, y_i > 0, x_i + w_i < W, y_i + h_i < H \; \forall d_i \in D_a \qquad (20)$$

where (x_i, y_i) represents the coordinates of the lower left corner of d_i, and w_i and h_i are the width and height of d_i on the grid, respectively.

Moreover, we use a 0-1 variable $u^{d_i}_{x,y}$ to represent whether a cell (x, y) in the device layer is occupied by device d_i. Since any two devices cannot overlap with each other, we have

$$\sum_{d_i \in D_a} u^{d_i}_{x,y} \le 1 \quad \forall (x, y) \in R_d. \qquad (21)$$

When performing the flow-path planning of each $l_i \in L$, a flow port and a waste port should be placed at the two ends of l_i and thus, can be constrained as

$$\sum_{(x,y)\in R_p} u^{f_i}_{x,y} = 1, \sum_{(x,y)\in R_p} u^{w_i}_{x,y} = 1 \quad \forall l_i \in L \qquad (22)$$

where $u^{f_i}_{x,y}$ and $u^{w_i}_{x,y}$ are 0-1 variables indicating whether a cell (x, y) is occupied by the flow port and the waste port of flow path l_i, respectively.

Similarly, a flow port and a waste port cannot be placed at the same cell on the grid and thus, can be constrained as

$$u^{f_i}_{x,y} + u^{w_i}_{x,y} \le 1, u^{f_i}_{x,y} + u^{w_j}_{x,y} \le 1 \quad \forall l_i, l_j \in L \quad \forall (x, y) \in R_p. \qquad (23)$$

Flow paths can be divided into three types as mentioned in Section II-B3. To construct the corresponding flow channels of l_i, it can be further split into a set of end-to-end routing nets, denoted by N_{l_i}. For example, a *Type 1* path consists of three nets, i.e., (flow port, source device), (source device, target device), and (target device, waste port).

Afterward, for each net $n_j(v^1_j, v^2_j) \in N_{l_i}$, where v^1_j and v^2_j are the two ends of n_j, which can be a fluidic port, a device, or a flow channel occupied by excess fluid, our task is to construct a routing path of n_j such that v^1_j and v^2_j are connected by channels. Correspondingly, one of the adjacent cells of v^1_j and v^2_j on the grid should be the start point and end point of the routing path, respectively. Thus, we have the following constraints:

$$\sum_{(x,y)\in AC_{v^1_j}} u^{n_j}_{x,y} = 1, \sum_{(x,y)\in AC_{v^2_j}} u^{n_j}_{x,y} = 1 \quad \forall n_j \in N_{l_i} \qquad (24)$$

where $u^{n_j}_{x,y}$ is a 0-1 variable representing whether a cell (x, y) is on the routing path of n_j (excluding v^1_j and v^2_j). $AC_{v^1_j}$ and $AC_{v^2_j}$ are adjacent cells of v^1_j and v^2_j on the grid, respectively.

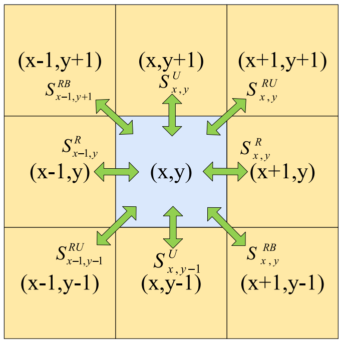


Fig. 10. Adjacent cells of (x, y) in a diagonal routing plane.

Meanwhile, for any cell $(x, y) \in R$ on the routing path of n_j except for the start and end points, to ensure the connectivity of the path, exactly two adjacent cells of (x, y) should also be on the path, constrained as

$$\sum_{(x',y')\in AC_{x,y}} u^{n_j}_{x',y'} = 2$$
$$\forall n_j \in N_{l_i} \; \forall (x, y) \in R \wedge u^{n_j}_{x,y} = 1 \wedge (x, y) \notin AC_{v^1_j} \cup AC_{v^2_j} \qquad (25)$$

where $AC_{x,y}$ is a set of adjacent cells of (x, y) on the grid. Note that when diagonal routing in the X architecture is implemented, (x, y) has eight adjacent cells on the grid as illustrated in Fig. 10, i.e., $|AC_{x,y}| = 8$.

When flow path l_i is used to transport a fluid sample between devices d_j and d_k (i.e., a *Type 1* path), as illustrated in Fig. 6, the corresponding excess fluid should be cached at the two ends of d_k using the channels of l_i. Assuming that e_{l_i} represents the excess fluid generated by this transportation task, we have

$$u^{e_{l_i}}_{x,y} \le u^{n_j}_{x,y} \quad \forall n_j \in N_{l_i} \quad \forall (x, y) \in R \qquad (26)$$

where $u^{e_{l_i}}_{x,y}$ is a 0-1 variable representing whether a cell (x, y) is used for caching e_{l_i}.

For each segment of cached fluid, the corresponding caching channel must start from a grid cell adjacent to d_k and end at a grid cell on l_i. We use a 0-1 variable $\theta^{l_i}_{x,y}$ to represent whether a cell (x, y) is occupied by the end of a caching channel on l_i. Then, we have

$$\sum_{(x,y)\in AC_{d_k}} u^{e_{l_i}}_{x,y} = 1 \qquad (27)$$

$$\theta^{l_i}_{x,y} \le u^{e_{l_i}}_{x,y} \quad \forall (x, y) \in R \qquad (28)$$

$$\sum_{(x,y)\in R} \theta^{l_i}_{x,y} = 1 \qquad (29)$$

$$\sum_{(x',y')\in AC_{x,y}} u^{e_{l_i}}_{x',y'} = 1 \quad \forall (x, y) \in R \wedge \theta^{l_i}_{x,y} = 1. \qquad (30)$$

To ensure the connectivity of a caching channel, for any cell (x, y) on the channel except for the two ends, two adjacent cells of (x, y) must be on the caching channel, constrained as

$$\sum_{(x',y')\in AC_{x,y}} u^{e_{l_i}}_{x',y'} = 2$$
$$\forall (x, y) \in R \wedge u^{e_{l_i}}_{x,y} = 1 \wedge (x, y) \notin AC_{d_k} \wedge \theta^{l_i}_{x,y} \ne 1. \qquad (31)$$

Assuming that the volume of excess fluid e_{l_i} is $V_{e_{l_i}}$, to ensure a proper caching, sufficient channels should be reserved at the two ends of the target device d_k. Accordingly, we have the following constraint:

$$\sum_{(x,y)\in R} u^{e_{l_i}}_{x,y} \geq \frac{V_{e_{l_i}}}{V_u} \tag{32}$$

where V_u is the capacity a grid cell.

Since flow path l_i cannot pass through any cell occupied by a fluidic port, we have

$$u^{n_j}_{x,y} + u^{f_k}_{x,y} + u^{w_k}_{x,y} \leq 1 \quad \forall n_j \in N_{l_i} \quad \forall l_k \in L \quad \forall (x,y) \in R. \tag{33}$$

Furthermore, if there exists another flow path l_j, whose corresponding fluid transportation/removal task needs to be performed concurrently with that of l_i in the scheduling, denoted by $l_i \| l_j$, the two flow paths cannot share a waste port and should avoid forming any channel intersection. Thus, we have following constraints:

$$u^{w_i}_{x,y} + u^{w_j}_{x,y} \leq 1 \quad \forall (x,y) \in R_p \quad \forall l_i, l_j \in L \wedge l_i \| l_j \tag{34}$$

$$\sum_{n_k \in N_{l_i} \cup N_{l_j}} u^{n_k}_{x,y} \leq 1 \quad \forall (x,y) \in R \quad \forall l_i, l_j \in L \wedge l_i \| l_j. \tag{35}$$

Note that if the fluid manipulation performed on either l_i or l_j above is an external inputting task (e.g., a_1–a_3 in Fig. 2), denoted by $l_i \overset{s}{\|} l_j$, the two flow paths cannot share a flow port, constrained as

$$u^{f_i}_{x,y} + u^{f_j}_{x,y} \leq 1 \quad \forall (x,y) \in R_p \quad \forall l_i, l_j \in L \wedge l_i \overset{s}{\|} l_j. \tag{36}$$

We refer to a device, that is, performing either an inputting/outputting task or a biochemical operation as an active device. Since flow path l_i cannot pass through an active device during its execution period, we have

$$u^{n_k}_{x,y} + u^{d_j}_{x,y} \leq 1 \quad \forall n_k \in N_{l_i} \quad \forall (x,y) \in R_d \quad \forall d_j \in D_i \tag{37}$$

where D_i is a set of active devices during the exection of l_i's fluid transportation/removal task.

Additionally, we use 0-1 variables $S^c_{x,y}$ and $S^p_{x,y}$ to represent whether a cell (x,y) is occupied by a flow channel and a fluidic port, respectively, which can be constrained as

$$u^{n_i}_{x,y} \leq S^c_{x,y}, S^c_{x,y} \leq \sum_{n_j \in N} u^{n_j}_{x,y} \ \forall n_i \in N \ \forall (x,y) \in R \tag{38}$$

$$u^{f_i}_{x,y} + u^{w_i}_{x,y} \leq S^p_{x,y}, S^p_{x,y} \leq \sum_{l_j \in L} \left(u^{f_j}_{x,y} + u^{w_j}_{x,y} \right) \quad \forall l_i \in L \quad \forall (x,y) \in R_p \tag{39}$$

where N is the set of the end-to-end nets after spliting all the flow paths in L.

To count the number of channel intersections, we have the following definition. A grid cell (x,y) has an upward routing channel if $\exists n_j(v_j^1, v_j^2) \in N$ such that both (x,y) and $(x,y+1)$ are on the net n_j. In particular, if $(x,y+1) \in v_j^1(v_j^2)$ and $v_j^1(v_j^2)$ occupies more than one cell on the routing grid (e.g., a device), $v_j^1(v_j^2)$ will be seen as a whole placed above the grid cell (x,y). Moreover, a 0-1 variable $S^U_{x,y}$ is used to represent whether grid cell (x,y) has an upward routing channel as illustrated in Fig. 10. Similarly, we use 0-1 variables $S^R_{x,y}$, $S^{RU}_{x,y}$, and $S^{RD}_{x,y}$ to represent whether cell (x,y) has a rightward, a right upward, and a right downward routing channel, respectively. Accordingly, the variables above can be constrained as

$$S^U_{x,y} = \begin{cases} 1, & \exists n_j \in N, u^{n_j}_{x,y} + u^{v_j^1}_{x,y} + u^{v_j^2}_{x,y} = 1 \\ & \wedge u^{n_j}_{x,y+1} + u^{v_j^1}_{x,y+1} + u^{v_j^2}_{x,y+1} = 1 \\ 0, & \text{otherwise} \end{cases}$$
$$0 \leq x < W, 0 \leq y < H - 1 \tag{40}$$

$$S^R_{x,y} = \begin{cases} 1, & \exists n_j \in N, u^{n_j}_{x,y} + u^{v_j^1}_{x,y} + u^{v_j^2}_{x,y} = 1 \\ & \wedge u^{n_j}_{x+1,y} + u^{v_j^1}_{x+1,y} + u^{v_j^2}_{x+1,y} = 1 \\ 0, & \text{otherwise} \end{cases}$$
$$0 \leq x < W - 1, 0 \leq y < H \tag{41}$$

$$S^{RU}_{x,y} = \begin{cases} 1, & \exists n_j \in N, u^{n_j}_{x,y} + u^{v_j^1}_{x,y} + u^{v_j^2}_{x,y} = 1 \\ & \wedge u^{n_j}_{x+1,y+1} + u^{v_j^1}_{x+1,y+1} + u^{v_j^2}_{x+1,y+1} = 1 \\ & \wedge u^{v_j^1}_{x+1,y} + u^{v_j^2}_{x+1,y} \neq 1 \wedge u^{v_j^1}_{x,y+1} + u^{v_j^2}_{x,y+1} \neq 1 \\ 0, & \text{otherwise} \end{cases}$$
$$0 \leq x < W - 1, 0 \leq y < H - 1 \tag{42}$$

$$S^{RD}_{x,y} = \begin{cases} 1, & \exists n_j \in N, u^{n_j}_{x,y} + u^{v_j^1}_{x,y} + u^{v_j^2}_{x,y} = 1 \\ & \wedge u^{n_j}_{x+1,y-1} + u^{v_j^1}_{x+1,y-1} + u^{v_j^2}_{x+1,y-1} = 1 \\ & \wedge u^{v_j^1}_{x+1,y} + u^{v_j^2}_{x+1,y} \neq 1 \wedge u^{v_j^1}_{x,y-1} + u^{v_j^2}_{x,y-1} \neq 1 \\ 0, & \text{otherwise} \end{cases}$$
$$0 \leq x < W - 1, 1 \leq y < H \tag{43}$$

where $u^{v_j^1}_{x,y}$ and $u^{v_j^2}_{x,y}$ are 0-1 variables representing whether (x,y) is on v_j^1 and v_j^2, respectively.

Because having a leftward routing channel for a grid cell (x,y) is equivalent to the case that cell $(x-1,y)$ has a rightward routing channel, the former can be represented by $S^R_{x-1,y}$ as illustrated in Fig. 10. Similarly, variables $S^U_{x,y-1}$, $S^{RU}_{x-1,y-1}$, and $S^{RD}_{x-1,y+1}$ can be used to represent whether (x,y) has a downward, a left downward, and a left upward routing channel, respectively.

Next, we use a 0-1 variable $S^g_{x,y}$ to represent whether a cell (x,y) is occupied by a channel intersection on the chip. There are three possible cases as illustrated in Fig. 11, i.e., channel intersections formed by rectilinear paths [Fig. 11(a) and (b)], channel intersections formed by diagonal paths [Fig. 11(c) and (d)], and channel intersections formed by a rectilinear path and a diagonal path [Fig. 11(e) and (f)]. In each case, a channel intersection has a routing channel in at least three different directions. Thus, we have

$$S^U_{x,y} + S^{RU}_{x,y} + S^R_{x,y} + S^{RD}_{x,y} + S^U_{x,y-1} + S^{RU}_{x-1,y-1} + S^R_{x-1,y} + S^{RD}_{x-1,y+1} < 3 + \left(1 - S^c_{x,y}\right)M + S^g_{x,y}M$$
$$1 \leq x < W, 1 \leq y < H - 1 \tag{44}$$

$$S^U_{x,y} + S^{RU}_{x,y} + S^R_{x,y} + S^{RD}_{x,y} + S^U_{x,y-1} + S^{RU}_{x-1,y-1} + S^R_{x-1,y} + S^{RD}_{x-1,y+1} > 2 - \left(1 - S^c_{x,y}\right)M - \left(1 - S^g_{x,y}\right)M$$
$$1 \leq x < W, 1 \leq y < H - 1 \tag{45}$$

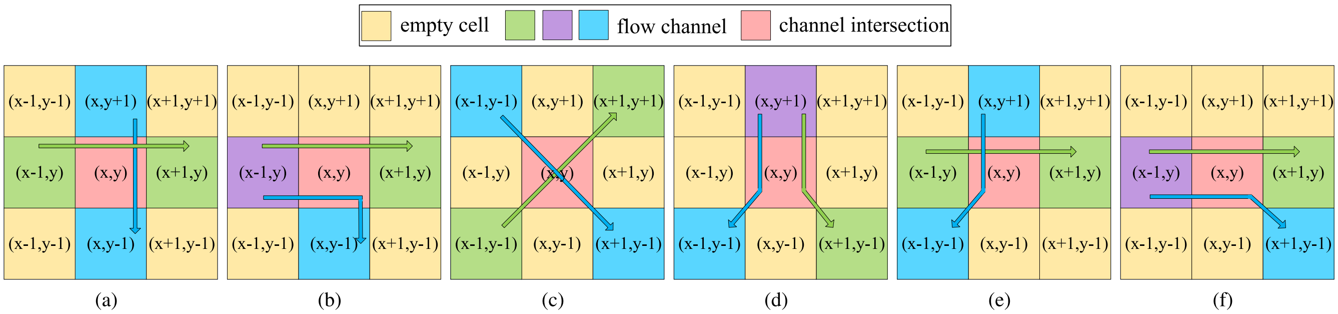


Fig. 11. Channel intersections formed by flow paths. (a) and (b) are intersections formed by rectilinear paths, where (a) is a "†" intersection and (b) is a "⊤" intersection. (c) and (d) are intersections formed by two diagonal paths. (d) and (e) are intersections formed by a rectilinear path and a diagonal path.

$$S^g_{x,y} \le S^c_{x,y}, 0 \le x < W, 0 \le y < H. \tag{46}$$

To reduce the number of extra introduced valves, a grid cell (x, y) is allowed to have a routing channel in at most four different directions, respectively, and thus, can be constrained as

$$\begin{aligned} &S^U_{x,y} + S^{RU}_{x,y} + S^R_{x,y} + S^{RD}_{x,y} + S^U_{x,y-1} + S^{RU}_{x-1,y-1} + S^R_{x-1,y} \\ &\quad + S^{RD}_{x-1,y+1} \le 4 + \left(1 - S^c_{x,y}\right)M \\ &\quad 1 \le x < W, 1 \le y < H-1. \end{aligned} \tag{47}$$

Furthermore, to reduce the complexity of channel routing, we assume that a channel intersection formed by flow paths can only be located on a grid cell. In other words, channel intersections are not allowed to be located at the corner of grid cells. Thus, we have

$$S^{RU}_{x,y} + S^{RD}_{x,y+1} \le 1, \; 0 \le x < W, 0 \le y < H-1. \tag{48}$$

Finally, a biochip architecture with minimized total cost can be generated by solving the following optimization problem:

$$\text{minimize } \gamma \sum_{(x,y)\in R} S^c_{x,y} + \kappa \sum_{(x,y)\in R_p} S^p_{x,y} + \xi \sum_{(x,y)\in R} S^g_{x,y} \tag{49}$$

$$\text{subject to } (20)\text{—}(48) \tag{50}$$

where γ, κ, and ξ are three weighting factors.

C. Application Remapping

As discussed in Section II-A, after completing the physical design, the bioassay should be remapped to the generated chip architecture by computing an accurate delay for each fluid transportation/removal task, thus, generating an exact execution procedure of the assay.

Assume that Ψ represents the set of fluid transportation and removal tasks specified in the high-level synthesis result, i.e., $\Psi = \{p_{j,i,l}, r_{j,i}, w_{j,i} | e_{j,i} \in E, l = 1, 2, 3\}$. For each task $\psi_i \in \Psi$, the duration of ψ_i can be computed as

$$T_{\psi_i} = \frac{\text{len}_{\psi_i} + \text{len}_{\text{flow}_i}}{v_f} \tag{51}$$

where len_{ψ_i}, $\text{len}_{\text{flow}_i}$, and v_f represent the length of the transportation/removal path of ψ_i, the length of the fluid flow of ψ_i, and the flow velocity, respectively.

Afterward, real delays of transportation/removal tasks can be fed back to (3), (5), (6), and (8) in Section IV-A by replacing the predefined constant T_p, thereby generating an updated scheduling matching with the generated chip architecture. Note that if the flow paths of two tasks ψ_i, $\psi_j \in \Psi$ overlap with each other on the chip, ψ_i and ψ_j cannot be performed concurrently in the updated scheduling, constrained as

TABLE III
DETAILS OF BENCHMARKS USED IN THE EXPERIMENTS

Benchmarks $\lvert O\rvert$/($\lvert$mixer$\rvert$, $\lvert$heater$\rvert$, $\lvert$filter$\rvert$, $\lvert$separator$\rvert$, $\lvert$detector$\rvert$, $\lvert$storage$\rvert$)/$\lvert E\rvert$		
PCR 7/(4,0,0,0,0,1)/15	IVD 12/(4,0,0,0,4,1)/24	ProteinSplit 14/(4,0,0,3,3,1)/27
Kinase act-1 4/(4,0,0,4,0,1)/16	Kinase act-2 12/(4,0,0,4,0,1)/48	Synthetic1 10/(4,2,3,0,2,1)/15
Synthetic2 15/(4,3,2,0,3,1)/21	Synthetic3 20/(4,4,3,4,2,1)/28	Synthetic4 25/(4,4,3,0,2,1)/33
Synthetic5 30/(4,3,3,2,4,1)/42	Synthetic6 60/(2,8,8,2,2,1)/63	

$$\epsilon_{i,j} M + t^s_{\psi_i} \ge t^e_{\psi_j} \tag{52}$$

$$\left(1 - \epsilon_{i,j}\right) M + t^s_{\psi_j} \ge t^e_{\psi_i} \tag{53}$$

where $\epsilon_{i,j}$ is a 0-1 variable representing the execution order of ψ_i and ψ_j ($\epsilon_{i,j} = 1$ if ψ_i is first executed. Otherwise, $\epsilon_{i,j} = 0$), $t^s_{\psi_i}$ ($t^s_{\psi_j}$) represents the start time of ψ_i (ψ_j), $t^e_{\psi_i}$ ($t^e_{\psi_j}$) represents the end time of ψ_i (ψ_j), and M is a very large constant.

Let T'_{assay} be the assay completion time after application remapping, which can be constrained as

$$t^e_i \le T'_{\text{assay}} \quad \forall o_i \in O. \tag{54}$$

Then, based on the previously generated binding scheme, the final completion time of the bioassay can be obtained by solving the following optimization problem:

$$\text{minimize } T'_{\text{assay}} \tag{55}$$

$$\text{subject to } (3)\text{–}(16), (51)\text{–}(54). \tag{56}$$

V. EXPERIMENTAL RESULTS

The proposed design flow PD+ was implemented in C++ and tested on a PC with 2.50-GHz CPU and 8-GB memory. The ILP problems formulated in Section IV were solved by the solver Gurobi. Eleven benchmarks were used to verify the proposed method as shown in Table III, in which polymerase chain reaction (PCR) [31], *in-vitro* diagnostics (IVD), Proteinsplit, Kinase act-1, and Kinase act-2 are real-world biochemical applications [25] and the other six assays

TABLE IV
COMPARISON RESULTS BETWEEN PD+ AND LSA REGARDING HIGH-LEVEL SYNTHESIS AND PHYSICAL DESIGN

Benchmark	High-Level Synthesis						Physical Design								
	T_{assay} (s)			N_{path}			N_{port}			$N_{intersection}$			$L_{channel}$ (mm)		
	LSA	PD+	I_m (%)	LSA	PD+	I_m (%)	LSA	PD+	I_m (%)	LSA	PD+	I_m (%)	LSA	PD+	I_m (%)
PCR	25	23	8.00	32	17	46.88	11	6	45.45	11	4	63.64	130	80	38.46
IVD	33	30	9.09	30	18	40.00	11	4	63.64	5	2	60.00	200	60	70.00
ProteinSplit	101	93	7.92	54	25	53.70	11	4	63.64	9	2	77.78	100	60	40.00
Kinase act-1	34	33	2.94	14	10	28.57	10	6	40.00	14	3	78.57	170	60	64.71
Kinase act-2	47	46	2.13	38	25	34.21	13	7	46.15	14	3	78.57	140	60	57.14
Synthetic1	44	39	11.36	31	22	29.03	13	8	38.46	16	5	68.75	210	110	47.62
Synthetic2	53	49	7.55	34	24	29.41	11	4	63.64	8	3	62.50	170	60	64.71
Synthetic3	66	57	13.64	44	24	45.45	9	4	55.56	6	2	66.67	100	70	30.00
Synthetic4	89	76	14.61	72	38	47.22	13	7	46.15	14	4	71.43	190	80	57.89
Synthetic5	72	70	2.78	66	45	31.82	16	9	43.75	12	3	75.00	170	60	64.71
Synthetic6	120	82	31.67	117	60	48.72	9	5	44.44	6	3	50.00	100	80	20.00
Average	–		10.15	–		39.55	–		50.08	–		68.45	–		50.48

are synthetic benchmarks. Parameters $|O|$, $|E|$, and $|device|$ in Table III are the numbers of biochemical operations and edges in the sequencing graph and the number of corresponding devices in the device library, respectively. The parameters used in the experiments are set as follows: $\alpha = 0.3$, $\beta = 0.7$, $\gamma = 0.45$, $T_p = 1$ s, $\kappa = 0.2$, $\xi = 0.35$, and $v_f = 10$ mm/s [28]. Moreover, the runtime on each benchmark was limited to 30 min for the solver to return the best-effort results.

A. Validation of the Proposed PathDriver+

Since there is no existing work targeting the architectural design of microfluidic biochips considering volume management, excess/waste fluid removal, fluidic-port minimization, and flow-path planning simultaneously, we implemented another method, called LSA, to verify the effectiveness of the proposed PD+. LSA integrates volume management and fluid removal into the well-know list scheduling algorithm to generate an efficient scheduling. The latter has been widely adopted in prior high-level synthesis work, such as [6], [9], [13], etc. Afterward, the simulated annealing algorithm [32] is employed to generate placement results of devices and the A*-search method is applied to complete the corresponding flow-path planning. Finally, bioassays are remapped to the constructed chip architectures to generate exact execution procedures.

We ran both algorithms on the aforementioned benchmarks. Table IV shows the corresponding comparison results with respect to both high-level synthesis and physical design, where columns T_{assay}, N_{path}, N_{port}, $N_{\text{intersection}}$, and L_{channel} are the completion time of bioassays, the total number of flow paths, the number of fluidic ports used in the chip, the number of channel intersections, and the total length of flow channels, respectively. I_m (%) in Table IV provides the relative improvement of PD+ over LSA.

In high-level synthesis, it can seen from Table IV that PD+ achieves a 2.13%–31.67% reduction in terms of the completion time of bioassays, with an average reduction of 10.15%. Moreover, the number of flow paths that need to be constructed is reduced by 39.55% on average, thereby reducing the complexity of chip architectures significantly. The following two reasons are responsible for this result: 1) although list scheduling has the advantage of being easy to be implemented, the corresponding solution quality is limited due to the adoption of a greedy strategy, i.e., biochemical operations that are ready to be executed are sequentially bound to an unoccupied device based on the dependencies defined in the sequencing graph. In contrast, by formulating the binding and scheduling into an ILP problem, PD+ considers all the operations and devices in a unified manner, thus generating a highly compact binding and scheduling scheme with minimized completion time of the bioassay and 2) in PD+, by binding operations to appropriate

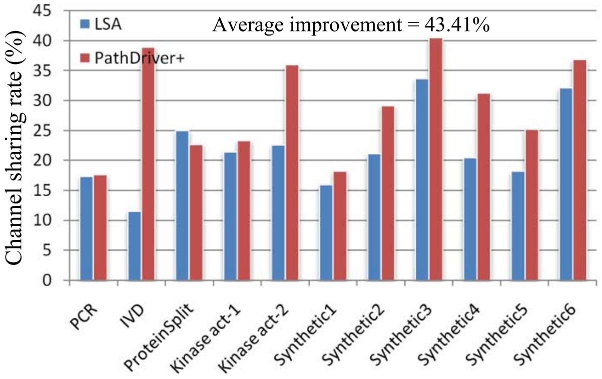


Fig. 12. Comparison results between PD+ and LSA with respect to the channel sharing rate.

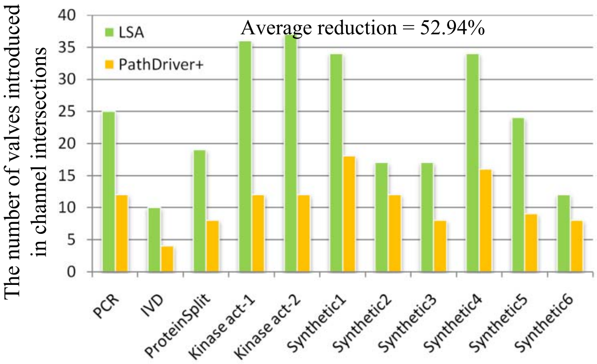


Fig. 13. Comparison results between PD+ and LSA with respect to the number of extra introduced valves.

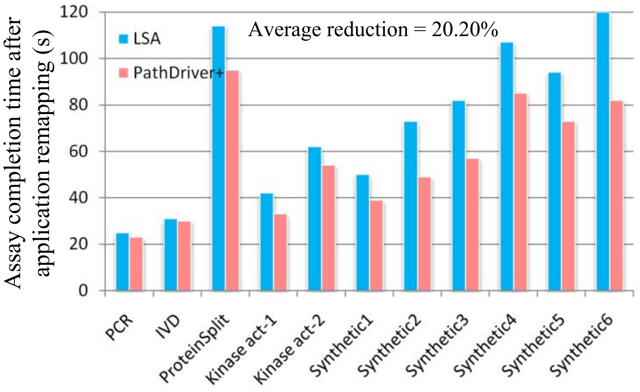


Fig. 14. Comparison results between PD+ and LSA with respect to the completion time of assays after application remapping.

devices and executing them in an optimized order, unnecessary fluid caching in a storage and waste fluid removal can be avoided, thereby reducing the number of flow paths.

Similarly, in physical design, PD+ achieves a 38.46%–63.64% (50.08% on average), a 50.00%–78.57% (68.45% on average), and a 20.00%–70.00% (50.48% on average) reduction across all the benchmarks with respect to the number of fluidic ports, the number of channel intersections, and the total length of flow channels, respectively. This result, once again, proves the strong optimization capability of the proposed design flow. The following reasons are responsible for this result: 1) since the number of flow paths has been reduced significantly during high-level synthesis in PD+, the complexity of physical design is reduced accordingly, thereby avoiding introducing unnecessary fluidic ports and channel intersections; 2) fluidic ports are shared sufficiently among those conflict-free flow paths in PD+, thus, further reducing the number of fluidic ports introduced into the chip; and 3) flow channels are also shared among flow paths as long as no conflict is introduced. As shown in Fig. 12, PD+ improves the channel sharing rate by 43.41% on average compared with LSA, thus reducing the total length of flow channels.

Fig. 13 shows the comparison results on the number of extra valves introduced into the chip. Since two to four valves need to be placed at a channel intersection to direct fluid flow toward the right direction while avoiding conflicts between transportation tasks [25], these valves need to be controlled independently during the execution of bioassays. It can be seen that PD+ achieves 52.94% reduction on average regarding the number of valves introduced in channel intersections and this, undoubtedly, will further reduce the complexity of chip control and the fabrication cost of the corresponding control system.

In addition, as discussed in Section IV-C, after completing the physical design, bioassays are remapped to the constructed chip architectures by computing an accurate duration for each fluid transportation/removal task, thus generating exact execution procedures of bioassays. Accordingly, Fig. 14 shows the comparison results on the completion time of bioassays after application remapping. It can be seen that PD+ outperforms LSA across all the benchmarks with an average reduction of 20.20%. This result further demonstrates that PD+ leads to biochip architectures with higher execution efficiency.

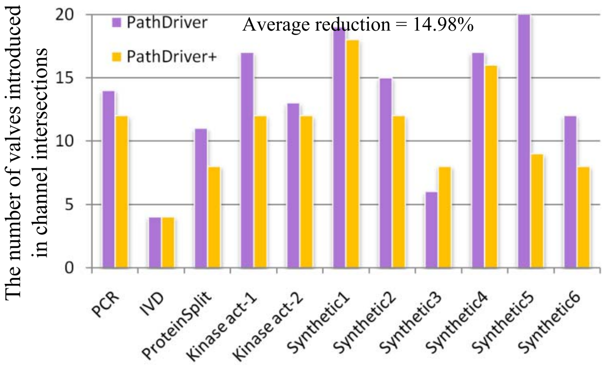


Fig. 15. Comparison results between PathDriver [1] and PD+ with respect to the number of extra introduced valves.

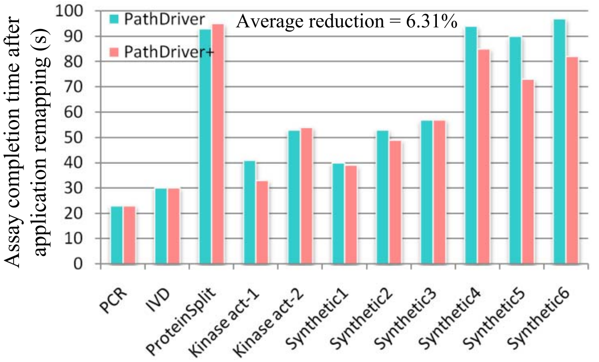


Fig. 16. Comparison results between PathDriver [1] and PD+ with respect to the completion time of bioassays after application remapping.

B. Comparison Between PathDriver [1] and PathDriver+

To further validate the effectiveness of the proposed PD+, in this part, comparisons are made between PD+ and the previous version of PathDriver [1]. Major improvements of PD+ over PathDriver in [1] include three aspects: the minimization of fluidic ports, the introduction of diagonal routing, as well as the implementation of application remapping. These new considerations not only reduce the overall chip cost but also can improve the execution efficiency of bioassays. Moreover, compared with the scheduling generated in high-level synthesis stage, the proposed application remapping method will further result in an exact and practical execution procedure of the complete bioassay.

Table V shows the comparison results with respect to the number of fluidic ports, the number of channel intersections, and the total length of flow channels. As expected, PD+ achieves a 0.00%–37.50% reduction in terms of the number of fluidic ports used in the chip, with an average reduction of 21.89%, due to a further consideration of fluidic-port sharing. As mentioned before, besides the extra overheads, a large number of ports will make biochips more vulnerable to physical defects such as leakage [33], since ports are holes punched on the chip. Hence, the minimization of fluidic ports has a great significance in prolonging the lifetime of biochips. Moreover, compared with PathDriver in [1], the number of channel intersections is also reduced by PD+ on most of the test cases. The maximum reduction rate reaches up to 57.14% with an average reduction of 27.32%. This is mainly because

TABLE V

COMPARISON RESULTS BETWEEN PATHDRIVER [1] AND PD+ IN TERMS OF THE NUMBER OF FLUIDIC PORTS, THE NUMBER OF CHANNEL INTERSECTIONS, AND THE TOTAL CHANNEL LENGTH

Benchmark	N_{port}			$N_{intersection}$			$L_{channel}$ (mm)		
	PathDriver	PathDriver+	I_m (%)	PathDriver	PathDriver+	I_m (%)	PathDriver	PathDriver+	I_m (%)
PCR	8	6	25.00	5	4	20.00	80	80	0.00
IVD	4	4	0.00	2	2	0.00	60	60	0.00
ProteinSplit	6	4	33.33	3	2	33.33	60	60	0.00
Kinase act-1	7	6	14.29	6	3	50.00	70	60	14.29
Kinase act-2	9	7	22.22	4	3	25.00	100	60	40.00
Synthetic1	9	8	11.11	6	5	16.67	110	110	0.00
Synthetic2	6	4	33.33	5	3	40.00	70	60	14.29
Synthetic3	6	4	33.33	2	2	0.00	80	70	12.50
Synthetic4	8	7	12.50	6	4	33.33	100	80	20.00
Synthetic5	11	9	18.18	7	3	57.14	90	60	33.33
Synthetic6	8	5	37.50	4	3	25.00	90	80	11.11
Average	–	–	21.89	–	–	27.32	–	–	13.23

the introduction of diagonal channels increases the flexibility of channel routing significantly. As a consequence, some intersections formed by traditional Manhattan channels can be avoided. On the other hand, compared with the Manhattan metrics, diagonal routing can further reduce the total length of flow channels, and this, can be verified by the results shown in the last three columns of Table V. It can be seen that the total channel length is reduced by 13.23% on average.

Fig. 15 shows the comparison results between PD+ and PathDriver [1] with respect to the number of valves introduced in channel intersections. Consistent with the results shown in Table V, PD+ reduces the number of extra introduced valves by 14.98% on average. Furthermore, since the proposed application remapping method in Section IV-C is fully compatible with PathDriver in [1], we directly apply this method to PathDriver to generate exact execution procedures of bioassays. Correspondingly, Fig. 16 shows comparison results on the completion time of bioassays after application remapping. It can be seen that PD+ leads to higher execution efficiency of assays on most of the test cases, with 6.31% reduction of assay completion time on average. The results above, once again, demonstrate the strong optimization capability of the proposed PD+.

VI. CONCLUSION

We have investigated the architectural design of continuous-flow microfluidics considering fluid volume management, excess/waste fluid removal, fluidic-port minimization, and flow-path planning with diagonal routing simultaneously. A path-driven design flow named PD+ has been proposed to systematically solve this problem. With the proposed design automation method, actual manipulation of fluid transportation and removal can be realized through an efficient flow-path network with minimized cost. Volumes of the fluids transported between devices are constrained strictly to ensure the correctness of assay outcomes. Furthermore, a tradeoff between the numbers of channel intersections and fluidic ports is realized to reduce the overall chip cost. The experimental results on multiple benchmarks have confirmed that PD+ leads to biochip architectures with both high efficiency and low cost.

REFERENCES

[1] X. Huang *et al.*, "PathDriver: A path-driven architectural synthesis flow for continuous-flow microfluidic biochips," in *Proc. Int. Conf. Comput.-Aided Design*, San Diego, CA, USA, 2020, pp. 1–8.

[2] L.-H. Hung, K. M. Choi, W.-Y. Tseng, Y.-C. Tan, K. J. Shea, and A. P. Lee, "Alternating droplet generation and controlled dynamic droplet fusion in microfluidic device for CdS nanoparticle synthesis," *Lab Chip*, vol. 6, no. 2, pp. 174–178, 2006.

[3] S. Einav *et al.*, "Discovery of a hepatitis C target and its pharmacological inhibitors by microfluidic affinity analysis," *Nat. Biotechnol.*, vol. 26, no. 9, pp. 1019–1027, 2008.

[4] X. Huang, T.-Y. Ho, W. Guo, B. Li, K. Chakrabarty, and U. Schlichtmann, "Computer-aided design techniques for flow-based microfluidic lab-on-a-chip systems," *ACM Comput. Surveys*, vol. 54, no. 5, pp. 1–29, 2021.

[5] Y. Zhu *et al.*, "Multicontrol: Advanced control-logic synthesis for flow-based microfluidic biochips," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 39, no. 10, pp. 2489–2502, Oct. 2020.

[6] X. Huang, W. Guo, Z. Chen, B. Li, T.-Y. Ho, and U. Schlichtmann, "Flow-based microfluidic biochips with distributed channel storage: Synthesis, physical design, and wash optimization," *IEEE Trans. Comput.*, early access, Jan. 26, 2021, doi: 10.1109/TC.2021.3054689.

[7] T. Banerjee, S. Poddar, S. Bhattacharjee, Y.-A. Song, A. Orozaliev, and B. B. Bhattacharya, "Sample preparation with free-flowing biochips using microfluidic binary-tree network," in *Proc. Int. Symp. Circuits Syst.*, Seville, Spain, 2020, pp. 1–5.

[8] T. Thorsen, S. J. Maerkl, and S. R. Quake, "Microfluidic large-scale integration," *Science*, vol. 298, no. 5593, pp. 580–584, 2002.

[9] Z. Chen, X. Huang, W. Guo, B. Li, T.-Y. Ho, and U. Schlichtmann, "Physical synthesis of flow-based microfluidic biochips considering distributed channel storage," in *Proc. Design Autom. Test Eur. Conf.*, Florence, Italy, 2019, pp. 1525–1530.

[10] R. Gómez-Sjöberg, A. A. Leyrat, D. M. Pirone, C. S. Chen, and S. R. Quake, "Versatile, fully automated, microfluidic cell culture system," *Anal. Chem.*, vol. 79, no. 22, pp. 8557–8563, 2007.

[11] T.-M. Tseng *et al.*, "Columba 2.0: A co-layout synthesis tool for continuous-flow microfluidic biochips," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 37, no. 8, pp. 1588–1601, Aug. 2018.

[12] M. Li, T.-M. Tseng, Y. Ma, T.-Y. Ho, and U. Schlichtmann, "VOM: Flow-path validation and control-sequence optimization for multilayered continuous-flow microfluidic biochips," in *Proc. Int. Conf. Comput.-Aided Design*, Westminster, CO, USA, 2019, pp. 1–8.

[13] X. Huang, T.-Y. Ho, W. Guo, B. Li, and U. Schlichtmann, "MiniControl: Synthesis of continuous-flow microfluidics with strictly constrained control ports," in *Proc. Design Autom. Conf.*, Las Vegas, NV, USA, 2019, pp. 1–6.

[14] K. Hu, T. A. Dinh, T.-Y. Ho, and K. Chakrabarty, "Control-layer routing and control-pin minimization for flow-based microfluidic biochips," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 36, no. 1, pp. 55–68, Jan. 2017.

[15] K. Yang, H. Yao, T.-Y. Ho, K. Xin, and Y. Cai, "AARF: Any-angle routing for flow-based microfluidic biochips," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 37, no. 12, pp. 3042–3055, Dec. 2018.

[16] S. L. Teig, "The X architecture: Not your father's diagonal wiring," in *Proc. ACM Int. Conf. Syst. Level Interconnect Prediction*, 2002, pp. 33–37.

[17] X. Huang, W. Guo, G. Liu, and G. Chen, "FH-OAOS: A fast four-step heuristic for obstacle-avoiding octilinear steiner tree construction," *ACM Trans. Design Autom. Electron. Syst.*, vol. 21, no. 3, pp. 1–31, 2016.

[18] X. Huang, G. Liu, W. Guo, and G. Chen, "Obstacle-avoiding octagonal steiner tree construction based on particle swarm optimization," in *Proc. 9th Int. Conf. Natural Comput.*, Shenyang, China, 2013, pp. 539–543.

[19] X. Huang, W. Guo, and G. Chen, "Fast obstacle-avoiding octilinear steiner minimal tree construction algorithm for VLSI design," in *Proc. 16th Int. Symp. Qual. Electron. Design*, Santa Clara, CA, USA, 2015, pp. 46–50.

[20] X. Huang, G. Liu, W. Guo, Y. Niu, and G. Chen, "Obstacle-avoiding algorithm in X-architecture based on discrete particle swarm optimization for vlsi design," *ACM Trans. Design Autom. Electron. Syst.*, vol. 20, no. 2, pp. 1–28, 2015.

[21] X. Huang, W. Guo, G. Liu, and G. Chen, "MLXR: Multi-layer obstacle-avoiding X-architecture steiner tree construction for VLSI routing," *Sci. China Inf. Sci.*, vol. 60, no. 1, pp. 1–3, 2017.

[22] K.-H. Tseng, S.-C. You, J.-Y. Liou, and T.-Y. Ho, "A top-down synthesis methodology for flow-based microfluidic biochips considering valve-switching minimization," in *Proc. Int. Symp. Phys. Design*, 2013, pp. 123–129.

[23] Q. Wang, H. Zou, H. Yao, T.-Y. Ho, R. Wille, and Y. Cai, "Physical co-design of flow and control layers for flow-based microfluidic biochips," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 37, no. 6, pp. 1157–1170, Jun. 2018.

[24] A. Grimmer, Q. Wang, H. Yao, T.-Y. Ho, and R. Wille, "Close-to-optimal placement and routing for continuous-flow microfluidic biochips," in *Proc. Asia South Pacific Design Autom. Conf.*, Chiba, Japan, 2017, pp. 530–535.

[25] X. Huang, T.-Y. Ho, K. Chakrabarty, and W. Guo, "Timing-driven flow-channel network construction for continuous-flow microfluidic biochips," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 39, no. 6, pp. 1314–1327, Jun. 2020.

[26] X. Huang *et al.*, "BigInter: One-pass architectural synthesis for continuous-flow microfluidic lab-on-a-chip systems," in *Proc. Int. Conf. Comput.-Aided Design*, 2021, pp. 1–8.

[27] M. Afghahi and C. Svensson, "Performance of synchronous and asynchronous schemes for VLSI systems," *IEEE Trans. Comput.*, vol. 41, no. 7, pp. 858–872, Jul. 1992.

[28] Y. C. Lim, A. Z. Kouzani, and W. Duan, "Lab-on-a-chip: A component view," *Microsyst. Technol.*, vol. 16, no. 12, pp. 1995–2015, 2010.

[29] C. Liu *et al.*, "DCSA: Distributed channel-storage architecture for flow-based microfluidic biochips," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 40, no. 1, pp. 115–128, Jan. 2021.

[30] *Programmable Microfluidics*. Accessed: 2007. [Online]. Available: http://groups.csail.mit.edu/cag/biostream/

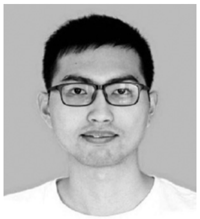
[31] Y. Luo, B. B. Bhattacharya, T.-Y. Ho, and K. Chakrabarty, "Design and optimization of a cyberphysical digital-microfluidic biochip for the polymerase chain reaction," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 34, no. 1, pp. 29–42, Jan. 2015.

[32] K. A. Dowsland and J. Thompson, "Simulated annealing," in *Handbook of Natural Computing*. Heidelberg, Germany: Springer, 2012, pp. 1623–1655.

[33] K. Hu, T. A. Dinh, T.-Y. Ho, and K. Chakrabarty, "Control-layer optimization for flow-based mVLSI microfluidic biochips," in *Proc. Int. Conf. Compilers Archit. Synth. Embedded Syst.*, Uttar Pradesh, India, 2014, pp. 1–10.

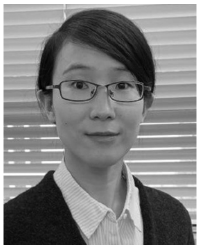
Xing Huang received the Ph.D. degree in electronic science and technology from Fuzhou University, Fuzhou, China, in 2018.

He is currently a Postdoctoral Research Fellow with the Chair of Electronic Design Automation, Technical University of Munich, Munich, Germany, sponsored by the Alexander von Humboldt Foundation. His current research interests include design automation for integrated circuits and microfluidic biochips.



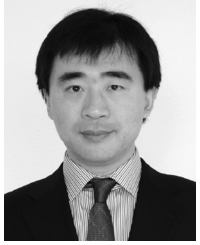
Youlin Pan received the bachelor's degree in computer science and technology from Fuzhou University, Fuzhou, China, in 2019, where he is currently pursuing the Ph.D. degree with the College of Mathematics and Computer Science.

His current research interest includes design automation for microfluidic biochips.



Grace Li Zhang received the Dr.-Ing. degree from the Technical University of Munich, Munich, Germany, in 2018, where she is currently pursuing the Habilitation degree with the Chair of Electronic Design Automation.

Her current research interests include high-performance and low-power design, as well as emerging systems.



Bing Li (Senior Member, IEEE) received the Dr.-Ing. and the Habilitation degrees from the Technical University of Munich (TUM), Munich, Germany, in 2010 and 2018, respectively.

He is currently a Researcher with the Chair of Electronic Design Automation, TUM. His research interests include high-performance and low-power design of integrated circuits and systems.

Wenzhong Guo (Member, IEEE) received the B.S. and M.S. degrees in computer science and the Ph.D. degree in communication and information system from Fuzhou University, Fuzhou, China, in 2000, 2003, and 2010, respectively.

He is currently a Full Professor with the College of Mathematics and Computer Science, Fuzhou University. His research interests include computational intelligence and very large-scale integration physical design.

Tsung-Yi Ho (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, in 2005.

He is a Professor with the Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan. His research interests include design automation and test for microfluidic biochips.

Ulf Schlichtmann (Senior Member, IEEE) received the Dipl.-Ing. and Dr.-Ing. degrees in electrical engineering and information technology from the Technical University of Munich (TUM), Munich, Germany, in 1990 and 1995, respectively.

He is a Professor and the Head of the Chair of Electronic Design Automation, TUM. He joined TUM in 2003, following 10 years in industry. His current research interests include computer-aided design of electronic circuits and systems, with an emphasis on designing reliable and robust systems. Increasingly, he focuses on emerging technologies, such as lab-on-chip and photonics.