

# Columba: Co-Layout Synthesis for Continuous-Flow Microfluidic Biochips

Tsun-Ming Tseng<sup>†</sup>, Mengchu Li<sup>†∇</sup>, Bing Li<sup>†</sup>, Tsung-Yi Ho<sup>\*◇</sup>, and Ulf Schlichtmann<sup>†</sup>

{tsun-ming.tseng, b.li, ulf.schlichtmann}@tum.de, mengchu.li@campus.lmu.de, tyho@cs.nctu.edu.tw

<sup>†</sup>Institute for Electronic Design Automation, Technische Universität München, Arcisstraße 21, 80333 München, Germany

<sup>∇</sup>Ludwig-Maximilians-Universität München, Geschwister-Scholl-Platz 1, 80539 München, Germany

<sup>\*</sup>Department of Computer Science, National Tsing Hua University, No. 101, Section 2, Kuang-Fu Road, 30013 Hsinchu, Taiwan

<sup>◇</sup>Institute for Advanced Study, Technische Universität München, Lichtenbergstrasse 2 a, 85748 Garching, Germany

## ABSTRACT

Continuous-flow microfluidics have evolved rapidly in the last decades, due to their advantages in effective and accurate control. However, complex control results in complicated valve actuations. As a result, sophisticated interactions between control and flow layers substantially raise the design difficulty. Previous work on design automation for microfluidics neglects the interactions between the control and flow layers and designs each layer separately, which leads to unrealistic designs. We propose the first planarity-guaranteed architectural model, and the first physical-design module models for important microfluidic components, which have modelled the interactions between both control and flow layers, while reducing the design difficulty. Based on the above, we propose the co-layout synthesis tool called **Columba**, which considers the pressure sharing among different valves, and routes channels in an any-angled manner. Experimental results show that complicated designs considering layer interactions can be synthesized for the first time.

## 1 Introduction

Continuous-flow microfluidics have evolved rapidly in the last decade thanks to their efficiency in high-throughput applications [1] [2] and convenience for operations requiring accurate reagent amounts [3] [4]. These advantages owe to their effectiveness in valve control. Valves are formed by segments of control and flow channels in corresponding layers. As shown in Figure 1(a)(b), manipulating the pressure in the control channel results in a shape change of the membrane between the control layer and the flow layer, which enables the control of the fluid flow. With this mechanism, switches guiding flow directions as shown in Figure 1(c), highly efficient rotary mixers as shown in Figure 1(d), and reaction chambers performing operations such as *metering* as shown in Figure 1(e), can easily be built.

The maximum number of valves in a single chip supported by current technology has already surpassed 1 million [5].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

DAC 2016, DOI: [10.1145/2897937.2897997](https://doi.org/10.1145/2897937.2897997)  
<http://ieeexplore.ieee.org/document/7544389/>  
<http://dl.acm.org/citation.cfm?id=2897997>

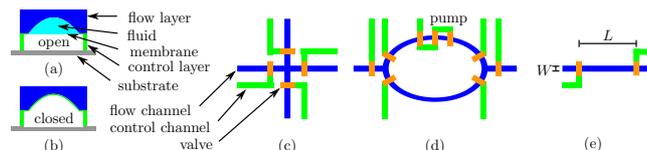


Figure 1: (a)(b) Two push-up valves with different pressure. (c) A four-end switch. (d) A rotary mixer. (e) A reaction chamber for  $W \times L$  volume metering.

However, valves are controlled by the pressure from chip ports, the number of which is limited due to their area cost (a port occupies up to  $1 \text{ mm}^2$  chip area [6]). Therefore, designs involving a large number of valves require port-sharing among those valves, and are thus used for biochemical assays consisting of similar sub-tasks such as protein crystallography [7] and high-throughput screening [8] [9].

For complex bioassays requiring precise control of each valve, more chip ports are needed to convey different pressure levels, which results in a limited number of valves. However, the design difficulty is not reduced with the reduction of the number of valves, since precise control results in complex valve actuations and therefore sophisticated interactions between control layer and flow layer. Manual designs need to ensure the coordination of both layers and usually take an experienced specialist several months.

Design automation approaches have thus been needed to alleviate the design difficulty, and several approaches have been devoted to generating place-and-route solutions for flow-based microfluidics automatically. However, current work performs separate design processes for flow layer and control layer, which make it impractical to treat the interactions between control and flow layer properly. For example, [10] together with [11] first generate device and channel-layout of the flow-layer, the side-effect of which is that the placement of all valves is decided without considering any constraints in control channel routing, thus the planarity of the control layer is neglected, which makes the design unrealistic. [12] also performs independent flow-layer design at first, but when a conflict occurs in the control-layer design, it falls back to the previous phase and performs the flow-layer design again. This tuning process will be performed iteratively until a feasible solution is found, but the feasible solution could be far from optimality.

Interactions between control and flow layers appear with the actuation of valves, which means that the interaction needs to be considered wherever a valve is implemented. In this work, we include all the valve implementation into *module models*, which can be regarded as the fundamental

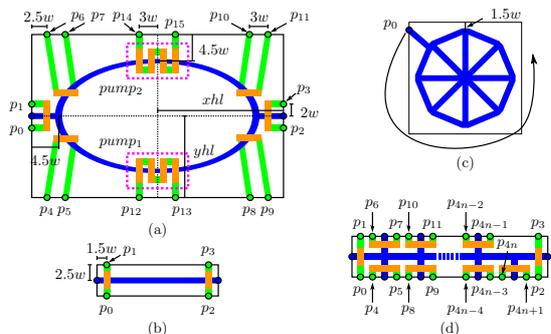


Figure 2: Physical-design module models for (a) Mixer. (b) Reaction Chamber. (c) Fluid port. (d) Switch.

unit of a planarity guaranteed architecture model. Based on this, we propose a co-layout synthesis tool named *Columba* that performs simultaneous place-and-route for both control and flow layers for the first time in the design automation field. Our contributions include:

- We propose the first physical-design module models for mixers, reaction chambers, switches, and ports, which include all the valve implementation on a chip and thus provide a basis for modelling all the interactions between control and flow layers.
- We propose the first planarity-guaranteed architectural model which ensures the feasibility of place-and-route solutions without unexpected overlapping among devices and channels.
- We propose the co-layout synthesis tool *Columba*, which consists of four design phases and focuses on the whole layout containing both layers from the very beginning, thus enabling designs with global view.
- We propose the first pressure sharing method in *Columba*, thus reducing the number of control ports.
- We route control and flow channels in an any-angled manner, as required for realistic designs.

## 2 Layout Abstraction

### 2.1 Physical-design Module Models

In our architectural model, mixers, reaction chambers, switches, and ports are defined as modules. A physical-design module model can be regarded as a bounding box with pins on its boundary for the connection with other modules. The layout inside of each module model is mostly pre-defined, but for some pins, alternative positions are provided, which ensure the flexibility of the global layout.

Figure 2 shows the inner structure of our module models. Flow channels are indicated by blue lines, control channels by green lines and valves by orange rectangles. When the orientation of a module is determined, the corresponding potential placement of valves and pins can be fixed. Most of the valves inside the module models are connected with two pins, one of which is connected with pressure sources, while the other is not necessarily enabled or may be connected with valves from other modules for pressure sharing (see Section 3.1.2). An exception are valves forming peristalsis pumps in the mixer module models, which are sequentially connected with each other as shown in Figure 2(a) and share the corresponding pins.

Figure 2(a) shows the module model for a mixer consisting of 4 valves forming a peristalsis pump and 6 valves con-

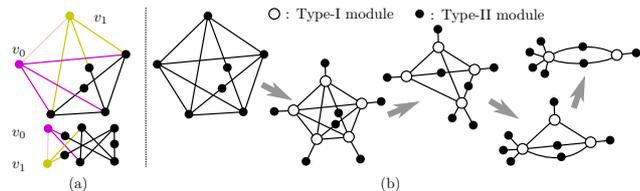


Figure 3: (a) Kuratowski graphs  $K_5$  and  $K_{3,3}$ . (b) An example of introducing switches to planarize the architectural model in our method.

trolling fluid flow. There are two possible locations for a pump in our module model (*pump1* and *pump2*).

Figure 2(b) shows the module model for a reaction chamber, which is simple but fundamental and can be used for performing several kinds of operations [13] [14] [15].

Figure 2(c) shows the module model for a fluid port with the shape from [6]. For the sake of planarity, each port possesses only one pin, which can be located anywhere on the module boundaries.

Figure 2(d) shows the module model for a switch, the number of valves and corresponding pins of which is determined by its degree.

### 2.2 Planarity-guaranteed Architectural Model

If we regard an architectural model as a graph, its modules as vertices, and the flow channels between modules as edges, then the edge crossing in this graph means the crossing of channels without a proper switch, which requires extra control effort that is unexpected. Therefore, we need to guarantee the planarity of the graph representing our architectural model.

Planarity is well-defined in Kuratowski's theorem [16]: *a graph is planar if and only if the graph does not contain a subgraph that is a subdivision of  $K_5$  or of  $K_{3,3}$* . Based on this theorem, we give the following lemma:

**Lemma.** *For a graph  $G=(V,E)$  that does not contain a barycentric subdivision of  $K_5$  or  $K_{3,3}$ ,  $G$  is non-planar only if there exist two adjacent vertices  $v_0, v_1 \in V$  both with degrees larger than or equal to 3.*

To verify this lemma, we aim to show that a non-barycentric subdivision of  $K_5$  or  $K_{3,3}$  contains two adjacent vertices  $v_0, v_1 \in V$  both with degrees larger than or equal to 3. A barycentric subdivision is a special subdivision that subdivides each edge of the concerning graph. As shown in Figure 3(a), a non-barycentric subdivision of  $K_5$  contains at least two adjacent vertices both with degree 4, and a non-barycentric subdivision of  $K_{3,3}$  contains at least two adjacent vertices both with degree 3.

**Corollary.** *In the graph  $G=(V,E)$  representing our architectural model, if  $G$  contains no barycentric subdivision of  $K_5$  or of  $K_{3,3}$  as its subgraph, and for every vertex with a degree larger than or equal to 3, each of its adjacent vertices has a degree no more than 2, the graph is planar.*

Based on the derived corollary, we classify our modules into two types: Type-I includes switch modules with unconstrained degree, and Type-II includes mixers, reaction chambers and port modules with a degree limited to no more than 2. As shown in Figure 3(b), white circles represent Type-I modules and black circles represent Type-II modules. If a Type-II module needs to receive inputs from or deliver outputs to multiple other Type-II modules, instead of connecting these modules directly with each other, we connect

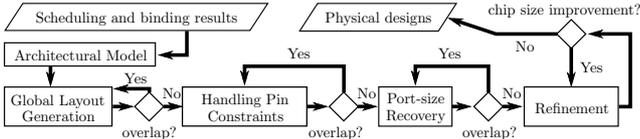


Figure 4: The flowchart of our method.

them with Type-I modules as transfer station to meet the degree limitations. If two Type-I modules are connected directly with each other, we will merge them together to form a new Type-I module. Therefore, the minimum degree of two adjacent modules is restricted to no more than 2, which means the the graph representing our architectural model must be planar.

When no new Type-I module is introduced and no more modules are merged, a check for the existence of barycentric subdivision will then be performed. If a barycentric subdivision of  $K_5$  or  $K_{3,3}$  is ascertained, we will arbitrarily pick 2 Type-I modules from this subdivision and merge them together until no more barycentric subdivision exists, so that the planarity of the architectural model can be guaranteed.

## 2.3 Problem Formulation

The problem formulation of this work is defined as:

### Input:

High-level-synthesis results of a biochemical assay indicating scheduling and binding solutions.

### Objective:

Minimize chip area, channel length, and the number of control ports.

### Output:

Physical design results for both control and flow layers.

## 3 Co-Layout Synthesis

The proposed physical design problem deals with both control and flow layers. In contrast to the simple combination of two general single-layer sub-problems, the interaction between both layers plays a key role in this problem, which means that well-researched methods for single-layer problems of printed circuit boards (e.g., [17] [18]) cannot be applied. In this work, we propose a four-phase co-layout synthesis method, the flowchart of which is shown in Figure 4.

### 3.1 Global Layout Generation

#### 3.1.1 Channel model

As the backbone of the global layout, we propose a directed channel model which consists of several bending points and channel segments. As shown in Figure 5(a), the bending points of the  $i$ -th channel  $c_i$  are indexed as  $b_{i,0}, \dots, b_{i,n-1}$ , and the channel segments are indexed as  $seg_{i,0}, \dots, seg_{i,n-2}$ , thereof  $b_{i,k}$  and  $b_{i,k+1}$  are also called the terminals of  $seg_{i,k}$ , and  $b_{i,0}$  and  $b_{i,n-1}$  are also called the terminals of  $c_i$ .  $n \in \mathbb{N}$  indicates the number of bending points.

We represent the location of the  $i'$ -th module containing valves by its center points  $(m_{i',x}, m_{i',y})$  in a coordinate system, together with  $m_{i',x} \pm \frac{1}{2}m_{i',w}$ ,  $m_{i',y} \pm \frac{1}{2}m_{i',h}$  indicating the location of the module boundaries,  $m_{i',w}$  and  $m_{i',h}$  indicating the width and height of the  $i'$ -th module.

In phase 1, for the sake of model reduction, instead of connecting each channel with specific pins of modules, we connect channels with the center points of modules. The incoming (outgoing) control channels connecting the same

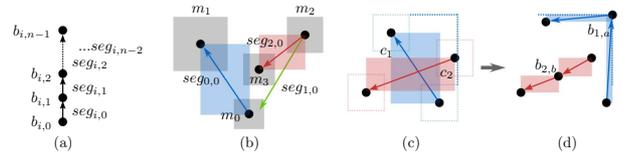


Figure 5: (a) Channel model. (b) An example of initial states of channels in global layout. (c) An example of channel crossing. (d) An example of adding new bending points.

pair of modules are regarded as a whole *channel bundle* sharing the same bending points. We index the  $j$ -th channel bundle as  $cb_j$ , along with a new variable  $ncb_j$  indicating the number of control channels merged by  $cb_j$ . We assign a group of control ports with no area cost (see Section 3.3) to each module containing valves. The number of ports which a port group eventually contains is determined by the number of control channels connected to it.

#### 3.1.2 Pressure Source Sharing

Valves inside the same module are required to work simultaneously with different actuation patterns, and therefore need to be driven by individual pressure sources. However, valves in different modules that do not have any overlapping working period according to the scheduling results can share the same pressure sources to reduce control efforts.

Therefore, besides an incoming control channel connected to a pressure source, every valve  $v_a$  may also possess an outgoing control channel, which is connected with another valve  $v_b$  as the incoming control channel of  $v_b$  in a different module, thus  $v_a$  and  $v_b$  are connected in series to share the same pressure source.

We introduce the following constraints to describe this pressure sharing scenario for each module  $m_{i'} \in M$ :

$$\sum_{\forall cb_j \in C_{in,i'}} ncb_j = \mathcal{V}_{i'}, \quad (1)$$

$$\sum_{\forall cb_j \in C_{in,i'}} ncb_j \geq \sum_{\forall cb_j \in C_{out,i'}} ncb_j \quad (2)$$

where  $M$  is the set of all modules,  $C_{in,i'}$  ( $C_{out,i'}$ ) is the set of all incoming (outgoing) channel bundles of  $m_{i'}$ , and  $\mathcal{V}_{i'}$  is a constant representing the number of valves in  $m_{i'}$ . (1) means that the number of incoming control channels connected to  $m_{i'}$  must equal the number of valves in  $m_{i'}$ , since every valve possesses an incoming control channel as its pressure source. (2) means that the number of the outgoing control channels connected to  $m_{i'}$  should be no more than the number of the incoming control channels, since valves do not all necessarily possess outgoing control channels.

#### 3.1.3 Channel Non-crossing Limitations

As shown in Figure 5(b), each channel possesses in its initial state only one segment formed by two terminals, which are the center points of the two modules connected by this channel. In order to avoid the crossing among channels in the same layer, we view each segment as the diagonal of a rectangle with exclusive chip area, and prohibit overlapping among these areas: suppose that  $b_{i,k}, b_{i,k+1}$  and  $b_{j,l}, b_{j,l+1}$  are the respective terminals of two different segments  $s_{i,k}$  and  $s_{j,l}$ , and the coordinate of a bending point (terminal) is represented as  $(b_{\cdot,\cdot,x}, b_{\cdot,\cdot,y})$ . There are four types of possible locations allowed for these terminals to avoid area overlap-

ping, which can be transformed into (3)(4)(5)(6):

$$\begin{aligned}
b_{i,k,x} &\leq b_{j,l,x} + q_{le}\Phi - w, & b_{i,k,x} &\leq b_{j,l+1,x} + q_{le}\Phi - w, \\
b_{i,k+1,x} &\leq b_{j,l,x} + q_{le}\Phi - w, & b_{i,k+1,x} &\leq b_{j,l+1,x} + q_{le}\Phi - w, & (3) \\
b_{i,k,x} &\geq b_{j,l,x} + q_{ri}\Phi - w, & b_{i,k,x} &\geq b_{j,l+1,x} + q_{ri}\Phi - w, \\
b_{i,k+1,x} &\geq b_{j,l,x} + q_{ri}\Phi - w, & b_{i,k+1,x} &\geq b_{j,l+1,x} + q_{ri}\Phi - w, & (4) \\
b_{i,k,y} &\leq b_{j,l,y} + q_{bo}\Phi - w, & b_{i,k,y} &\leq b_{j,l+1,y} + q_{bo}\Phi - w, \\
b_{i,k+1,y} &\leq b_{j,l,y} + q_{bo}\Phi - w, & b_{i,k+1,y} &\leq b_{j,l+1,y} + q_{bo}\Phi - w, & (5) \\
b_{i,k,y} &\geq b_{j,l,y} + q_{up}\Phi - w, & b_{i,k,y} &\geq b_{j,l+1,y} + q_{up}\Phi - w, \\
b_{i,k+1,y} &\geq b_{j,l,y} + q_{up}\Phi - w, & b_{i,k+1,y} &\geq b_{j,l+1,y} + q_{up}\Phi - w, & (6) \\
q_{ri} + q_{le} + q_{up} + q_{bo} &= 3 + q_p & & & (7)
\end{aligned}$$

where  $w$  is the minimum spacing distance between two segments, which is larger than the value specified in the design check rules (see Section 3.3) and  $\Phi$  is an extremely large auxiliary integer variable.  $q_{le}$ ,  $q_{ri}$ ,  $q_{bo}$ ,  $q_{up}$ , and  $q_p$  are auxiliary binary variables, thereof  $q_p$  is the *Lagrange multiplier*, the minimization of which is included in the optimization targets. If  $q_p = 0$ , one of (3)(4)(5)(6) has to be non-trivial according to (7), and a feasible solution of this inequality group indicates locations of two segments without area overlap.

If the overlapping of two segments is inevitable with the current bending points status,  $q_p$  has to be set to 1. In this situation, phase 1 will be rerun and new bending points will be added to the channels with conflicts, thus enabling a detour around the conflict area. As shown in Figure 5(c), the initial states of channel  $c_1$  and  $c_2$  result in an inevitable crossing. Therefore, new bending points  $b_{1,a}$  and  $b_{2,b}$  are added to  $c_1$  and  $c_2$  respectively. The locations of the existing bending points can be adjusted within a limited floating range according to their previous locations, and the coordinates of  $b_{1,a}$  and  $b_{2,b}$  are constrained by the locations of these existing binding points as shown in Figure 5(c). With  $b_{1,a}$  and  $b_{2,b}$ , now the crossing can be avoided as shown in Figure 5(d).

### 3.1.4 Module Placement

The non-overlapping limitations of modules are realized in a similar manner as Section 3.1.3:

$$m_{i',x} - \frac{1}{2}m_{i',w} \geq m_{j',x} + \frac{1}{2}m_{j',w} - q_{ri}\Phi, \quad (8)$$

$$m_{i',x} + \frac{1}{2}m_{i',w} \leq m_{j',x} - \frac{1}{2}m_{j',w} + q_{le}\Phi, \quad (9)$$

$$m_{i',y} - \frac{1}{2}m_{i',h} \geq m_{j',y} + \frac{1}{2}m_{j',h} - q_{up}\Phi, \quad (10)$$

$$m_{i',y} + \frac{1}{2}m_{i',h} \leq m_{j',y} - \frac{1}{2}m_{j',h} + q_{bo}\Phi, \quad (11)$$

$$q_{ri} + q_{le} + q_{up} + q_{bo} = 3 + q_p. \quad (12)$$

If  $q_{ri}$  ( $q_{le}, q_{up}, q_{bo}$ ) is set to 0,  $m_{i'}$  will be placed to the *right* (*left, upper, bottom*) side of  $m_{j'}$ .

In our module models, the flow channels are connected to a mixer module or a reaction chamber module from opposite directions. For the convenience of channel routing, we rule that if  $m_b$  and  $m_c$  are two modules connected to module  $m_a$ , then  $m_b$  and  $m_c$  must be placed at opposite directions relative to  $m_a$ . For example, if  $m_b$  is placed to the right side of  $m_a$ , then  $m_c$  has to be placed to the left side of  $m_a$ . We specify this rule by modifying the auxiliary binary variables in (8)-(12) (the detailed constraints are omitted here due to space limitations of the paper).

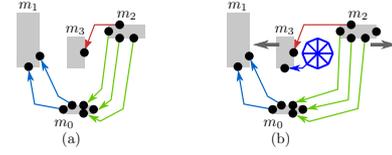


Figure 6: (a) An example of accurate layout. (b) An example of port-size recovery.

### 3.1.5 Objective

We represent the side lengths of a chip as two continuous variables  $s_x$  and  $s_y$ , and the set of all bending points as  $B$ . Then we introduce the following constraints to all  $b_{i,k} \in B$ :

$$b_{i,k,x} \geq d \wedge b_{i,k,x} \leq s_x - d \wedge b_{i,k,y} \geq d \wedge b_{i,k,y} \leq s_y - d \quad (13)$$

thereof  $d$  is the minimum spacing distance from the chip boundaries to a bending point. If  $b_{i,k}$  is the center point of a module  $m_{i'}$ , we increase  $d$  by  $\frac{1}{2}\max\{m_{i',w}, m_{i',h}\}$ . Since the chip area is a quadratic variable, we introduce a new continuous variable  $s_d = \max\{s_x, s_y\}$  to represent the dimension of the chip.

Therefore, we formulate our minimization objective as:

$$\alpha s_d + \alpha_1 s_x + \alpha_2 s_y + \beta \sum_{\forall cb_i \in C} cb_{i,l} + \gamma \sum_{\forall cb_i \in C_p} ncb_i + \sigma \sum_{\forall q_p \in E} q_p \quad (14)$$

thereof  $C$  is the set of all channel bundles,  $C_p$  is the set of all channel bundles that are connected with port groups,  $E$  is the set of Lagrange multipliers,  $cb_{i,l}$  is the length of  $cb_i$ , and  $\alpha$  (together with  $\alpha_1$  and  $\alpha_2$ ),  $\beta$ ,  $\gamma$ ,  $\sigma$  are thus the adjustable weight coefficients of chip area, total channel bundle lengths, the number of ports and Lagrange multipliers respectively, thereof  $\alpha_1, \alpha_2 \ll \alpha$  are used to prevent the abuse of chip area, and  $\alpha$  is increased progressively in the following phases.

## 3.2 Handling Pin Constraints

With the modelling results from phase 1 indicating the module locations and channel connections, we can specify the locations of valves and pins inside a module, which enables us to split a channel bundle into real channels connected with the pins for an accurate layout as shown in Figure 6(a), instead of with the center point of a module.

For a given module  $m_{i'}$ , we represent the set of all incoming channels connected to  $m_{i'}$  as  $C_{in}$ , the set of all outgoing channels connected to  $m_{i'}$  as  $C_{out}$ , and the set of pins in  $m_{i'}$  as  $P$ , thereof all pins are indexed according to the module models shown in Figure 2, and the location of a pin  $p_j \in P$  is represented as  $(m_{i',x} + \kappa_{j,x}, m_{i',y} + \kappa_{j,y})$ , where  $\kappa_{j,x}$  and  $\kappa_{j,y}$  are constants indicating the distance between  $p_j$  and the center point of  $m_{i'}$ .

Therefore, we can introduce the following constraints to locate the terminals  $b_{i,k}$  of control channels  $c_i \in C_{in} \cup C_{out}$  connected to  $m_{i'}$ :

$$b_{i,k,x} = m_{i',x} + \sum_{0 \leq j < |P|} q_{i,j} \kappa_{j,x}, \quad (15)$$

$$b_{i,k,y} = m_{i',y} + \sum_{0 \leq j < |P|} q_{i,j} \kappa_{j,y}, \quad (16)$$

thereof  $q_{i,j}$  is an auxiliary binary variable representing the selection of pins: if  $q_{i,j}$  is set to 1, channel  $c_i$  will be connected to pin  $p_j$ .

Then we introduce the following constraints to ensure that  $c_i$  will be connected to exactly one pin of  $m_{i'}$  (17), and a

Table 1: Generated Design Features.

	$D(mm^2)$	$L(mm)$	$\#(m, r, s, fp, cp)$	$T$
kinase act.[19]	$15.05 \times 15.05$	163.54	2, 2, 3, 7, 24	5m22s
acid proc.[3]	$18.35 \times 18.15$	252.83	3, 3, 3, 11, 40	9m5s
ChIP (4IP)[20]	$27.95 \times 26.65$	298.25	5, 4, 2, 17, 44	9m56s
mRNA iso.[21]	$22.77 \times 24.30$	564.01	4, 4, 3, 18, 54	34m42s
ChIP (10IP)	$38.15 \times 38.11$	556.42	11, 10, 2, 23, 100	46m10s

$D$ : chip dimension.  $L$ : total length of channels.  $\#m$ ,  $\#r$ ,  $\#s$ ,  $\#fp$ ,  $\#cp$ : number of mixers, reaction chambers, switches, fluid ports, and control ports.  $T$ : program runtime.

pin  $p_j$  can be connected with at most one channel (18):

$$\forall c_i \in C_{in} \cup C_{out}, \sum_{0 \leq j < |P|} q_{i,j} = 1, \quad (17)$$

$$0 \leq j \leq |P|, \sum_{0 \leq i < |C_{in} \cup C_{out}|} q_{i,j} \leq 1. \quad (18)$$

As mentioned in Figure 2 and Section 3.1.2, each valve in our modules possesses two pins, which can be indexed as  $p_{j'}$  and  $p_{j'+1}$ ,  $j' \in \{2k | k \in \mathbb{N}_0\}$ .

For a valve controlling fluid flow, one of its pins must be connected with an incoming control channel, while the other pin can be connected with an outgoing control channel or be left unused. Therefore, the number of pins connected with outgoing control channels must be no more than the number of pins connected with incoming control channels. This is described by the following constraints:

$$\sum_{i \in \{i | c_i \in C_{in}\}} q_{i,j'} \geq \sum_{i \in \{i | c_i \in C_{out}\}} q_{i,j'+1}, \quad (19)$$

$$\sum_{i \in \{i | c_i \in C_{in}\}} q_{i,j'+1} \geq \sum_{i \in \{i | c_i \in C_{out}\}} q_{i,j'}. \quad (20)$$

Since we have two possible locations for a pump in our mixer module, not all valves forming peristalsis pumps need to be connected with control channels. As shown in Figure 2(a), if we regard  $pump_1$  and  $pump_2$  as valves possessing two pins, only one of them will be connected with control channels and the other will be left unused. Therefore, we introduce the following constraints to describe channel connections for mixer modules:

$$\sum_{i \in \{i | c_i \in C_{in}\}} (q_{i,j'} + q_{i,j'+1}) \leq 1, \quad (21)$$

$$\sum_{i \in \{i | c_i \in C_{in}\}, j \in \{12, 13, 14, 15\}} q_{i,j} = 1. \quad (22)$$

We also put the non-crossing constraints mentioned in Section 3.1.3 on the new control channels generated in this phase, and rerun the optimization until no channel crossing exists.

### 3.3 Port Module Restoration

Since a port module contains only one pin, which can be located anywhere on the module boundary, the connection between ports and other modules is not so complicated as the connection between two modules containing valves. Therefore, for the sake of model reduction, the area costs of ports are ignored in phase 1 and phase 2.

In this phase, we restore the ports as modules with given side length. Since we have kept a relatively large minimum spacing distance between channels and modules in previous phases as mentioned in Section 3.1.3 and Section 3.1.4, it is not difficult to find suitable locations for these ports

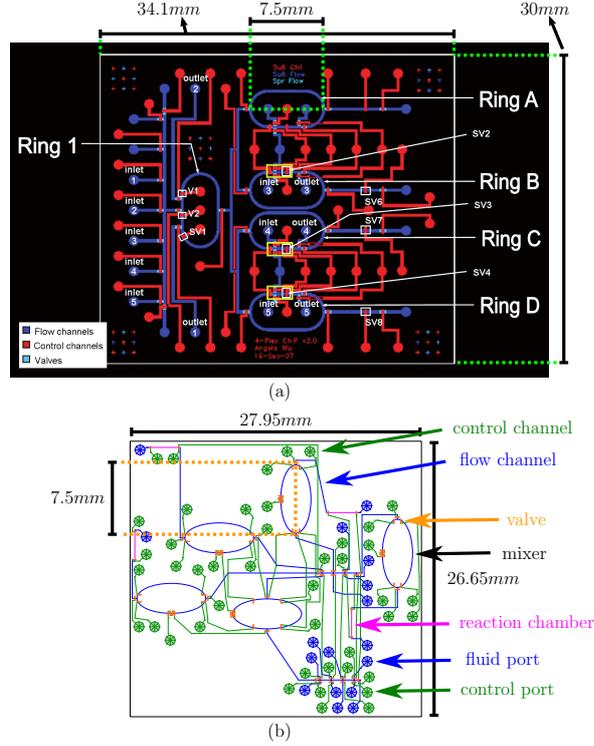


Figure 7: ChIP [20] design: (a) Manual (b) Columba.

with little draw-and-push efforts as shown in Figure 6(b). Then we put the non-crossing constraints on the channels connected with ports and other modules, and rerun the optimization until no channel crossing exists.

### 3.4 Refinement

In the last phase of our modelling process, we replace the above mentioned large minimum spacing distance with the actual requirement according to [6], and re-run the optimization for better utilization of the chip area.

## 4 Experimental Results

We implemented our method in C++ on a computer with 2.40 GHz CPU. The proposed model is linear and thus the optimization problem can be solved by the Mixed Integer Linear Programming (MILP) solver Gurobi [22].

Table 1 shows the feature values of the automatically-generated designs by Columba. The first four test cases are from [3] [19] [20] [21] and are listed with the order as their sizes, To show the scalability of Columba, we design a 10 Immunoprecipitation (IP) chip (ChIP (10IP)) by duplicating 6 more parallel IP process as the fifth test case. The number of the synthesized control ports  $\#cp$  as well as the program runtime  $T$  increase smoothly proportional to the design complexity, which is denoted by  $\#m$  (mixers),  $\#r$  (reaction chambers),  $\#s$  (switches),  $\#fp$  (fluid ports) as the given inputs from scheduling results.

Figure 7 shows the comparison between the manual design and the automatic design by Columba for ChIP (4IP). Blue and green lines in the figure indicate the flow and control channels, small blue and green circles represent the fluid and control ports, big blue ellipses represent the mixers, pink rectangles represent the reaction chambers, and orange segments represent the valves. Since the dimension of each mixer is clearly specified as  $7.5mm \times 3.8mm$  in [20], we can

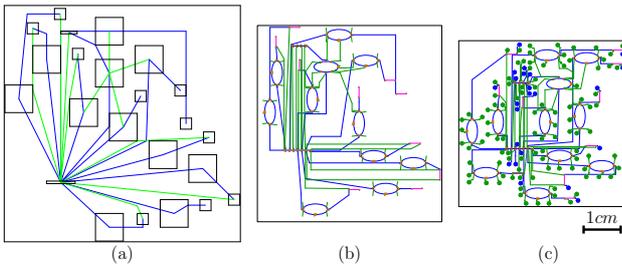


Figure 8: Temporary resultant graphs of test case ChIP(10IP): (a) Phase 1. (b) Phase 2. (c) Phase 3.

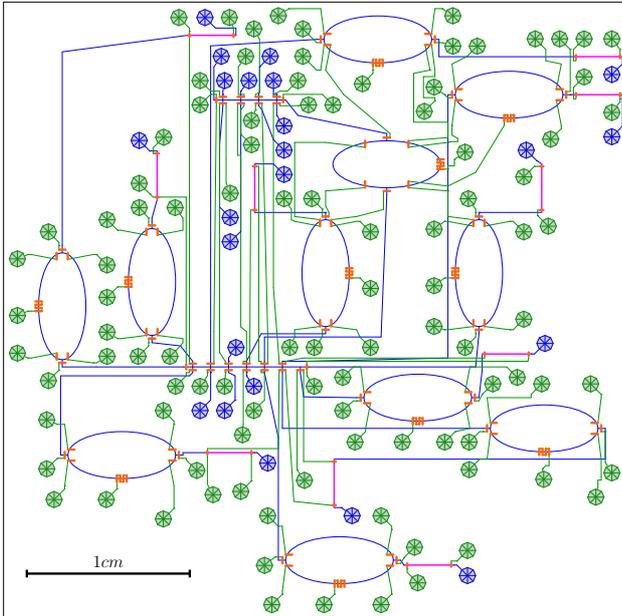


Figure 9: The final design of test case ChIP(10IP).

estimate the chip size of the manual design, which is larger than the automatic design generated by Columba.

Figure 8 shows the temporary resultant graphs of test case ChIP(10IP). The global layout is determined in phase 1 as shown in Figure 8(a); in phase 2, channel bundles are split into real channels and connected to pins of modules instead of the central points of modules as shown in Figure 8(b); and in phase 3, chip ports are restored as modules with area cost as shown in Figure 8(c). Since the weight coefficient  $\alpha$  of area cost is increased progressively, the chip area is gradually reduced. Figure 9 shows the snapshot of the final design generated by Columba for ChIP(10IP), which, though large and complicated, requires less than 1 hour for the entire synthesis.

## 5 Conclusion

In this paper we propose the co-layout synthesis tool Columba that performs simultaneous place-and-route for both control and flow layers and supports the major types of modules which are used in microfluidics. The library of the module models could be extended in the future and could be supported directly by Columba. Experimental results showed that Columba could deal with large designs within complex interaction between different layers very well.

## 6 References

[1] Richard A White III, Paul C Blainey, H Christina Fan, and Stephen R Quake. Digital PCR provides sensitive and

absolute calibration for high throughput sequencing. *BMC Genomics*, 2009.

- [2] Xiran Jiang, Ning Shao, Wenwen Jing, Shengce Tao, Sixiu Liu, and Guodong Sui. Microfluidic chip integrating high throughput continuous-flow PCR and DNA hybridization for bacteria analysis. *Talanta*, pages 246–250, 2014.
- [3] Jong Wook Hong, Vincent Studer, Giao Hang, W French Anderson, and Stephen R Quake. A nanoliter-scale nucleic acid processor with parallel architecture. *Nature Biotechnology*, 22(4):435–439, 2004.
- [4] Frederick K. Balagadde, Lingchong You, Carl L. Hansen, Frances H. Arnold, and Stephen R. Quake. Long-term monitoring of bacteria undergoing programmed population control in a microchemostat. *Science*, 309(5731):137–140, 2005.
- [5] Ismail Emre Araci and Stephen R. Quake. Microfluidic very large scale integration (mvlsi) with integrated micromechanical valves. *Lab on a Chip*, 12:2803–2806, 2012.
- [6] Stanford Foundry. *Basic Design Rules*. <http://web.stanford.edu/group/foundry>.
- [7] Liang Li and Rustem F. Ismagilov. Protein crystallization using microfluidic technologies based on valves, droplets, and slipchip. *Annu. Rev. Biophys.*, 39:139–158, 2010.
- [8] Todd Thorsen, Sebastian J. Maerkl, and Stephen R. Quake. Microfluidic large-scale integration. *Science*, 298(5593):580–584, 2002.
- [9] Zhanhui Wang, Min-Cheol Kim, Manuel Marquez, and Todd Thorsen. High-density microfluidic arrays for cell cytotoxicity analysis. *Lab on a Chip*, 7:740–745, 2007.
- [10] Wajid Hassan Minhass, Paul Pop, Jan Madsen, and Felician Stefan Blaga. Architectural synthesis of flow-based microfluidic large-scale integration biochips. In *Proc. Int. Conf. Compil., Arch. and Syn. Embed. Sys.*, pages 181–190, 2012.
- [11] Wajid Hassan Minhass, Paul Pop, Jan Madsen, and Tsung-Yi Ho. Control synthesis for the flow-based microfluidic large-scale integration biochips. In *Proc. Asia and South Pacific Des. Autom. Conf.*, pages 205–212, 2013.
- [12] Hailong Yao, Qin Wang, Yizhong Ru, Tsung-Yi Ho, and Yici Cai. Integrated flow-control co-design methodology for flow-based microfluidic biochips. *IEEE Des. Test. Comput.*, 2015.
- [13] Rafael Gomez-Sjoeberg, Anne A. Leyrat, Dana M. Pirone, Christopher S. Chen, and Stephen R. Quake. Versatile, fully automated, microfluidic cell culture system. *Anal. Chem.*, 79:8557–8563, 2007.
- [14] Jiang F. Zhong, Yan Chen, Joshua S. Marcus, Axel Scherer, Stephen R. Quake, Clive R. Taylor, and Leslie P. Weiner. A microfluidic processor for gene expression profiling of single human embryonic stem cells. *Lab on a Chip*, 8(1):68–74, 2008.
- [15] Adam K. White, Michael VanInsberghe, Oleh I. Petriv, Mani Hamidi, Darek Sikorski, Marco A. Marra, James Piret, Samuel Aparicio, and Carl L. Hansen. High-throughput microfluidic single-cell RT-qPCR. *Proc. Natl. Acad. Sci.*, 108(34):13999–14004, 2011.
- [16] William Thomas Tutte. How to draw a graph. *Proceedings of the London Mathematical Society*, Third Series(13):743–767, 1963.
- [17] Muhammet Mustafa Ozdal and Martin D. F. Wong. Algorithmic study of single-layer bus routing for high-speed boards. *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, 25(3):490–503, 2006.
- [18] Tsun-Ming Tseng, Bing Li, Tsung-Yi Ho, and Ulf Schlichtmann. Ilp-based alleviation of dense meander segments with prioritized shifting and progressive fixing in pcb routing. *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, 34(6):1000–1013, 2015.
- [19] Cong Fang, Yanju Wang, Nam T. Vu, Wei-Yu Lin, Yao-Te Hsieh, Liudmilla Rubbi, Michael E. Phelps, Markus Müschen, Yong-Mi Kim, Arion F. Chatzizoannou, Hsian-Rong Tseng, and Thomas G. Graeber. Integrated microfluidic and imaging platform for a kinase activity radioassay to analyze minute patient cancer samples. *Cancer Res.*, 70(21):8299–8308, 2010.
- [20] Angela R. Wu, Joseph B. Hiatt, Rong Lu, Joanne L. Attema, Neethan A. Lobo, Irving L. Weissman, Michael F. Clarke, and Stephen R. Quake. Automated microfluidic chromatin immunoprecipitation from 2,000 cells. *Lab on a Chip*, 9:1365–1370, 2009.
- [21] Joshua S. Marcus, W. French Anderson, and Stephen R. Quake. Microfluidic single-cell mRNA isolation and analysis. *Anal. Chem.*, 78:3084–3089, 2006.
- [22] Gurobi Optimization, Inc. *Gurobi Optimizer Reference Manual*. <http://www.gurobi.com>.