Sieve-valve-aware Synthesis of Flow-based Microfluidic Biochips Considering Specific Biological Execution Limitations

Mengchu Li[‡], Tsun-Ming Tseng[‡], Bing Li[‡], Tsung-Yi Ho[⋆◇], and Ulf Schlichtmann[‡]
[‡]Institute for Electronic Design Automation, Technische Universität München, Arcisstrasse 21, D-80333 Munich, Germany
[⋆]Department of Computer Science, National Tsing Hua University, No. 101, Section 2, Kuang-Fu Road, 30013 Hsinchu, Taiwan
[◇]Institute for Advanced Study, Technische Universität München, Lichtenbergstrasse 2 a, D-85748 Garching, Germany
mengchu.li@campus.lmu.de, {tsun-ming.tseng, b.li, ulf.schlichtmann}@tum.de, tyho@cs.nthu.edu.tw

***Abstract*—Microfluidic biochips are being used to perform ever more complex and error-prone bioassays. This results in increasing demand for design automation for such biochips, as these sophisticated designs are beyond the scope of manual design. So far, much research in the field of design automation has been devoted to satisfy this demand from biology, but the gap between design automation and biology is still huge. To narrow this gap, we propose a synthesis method in which sieve valves, which are key components in flow-based microfluidic biochips, are considered for the first time. In addition, we integrate three more constraints into our synthesis that are commonly seen in bioassays but have so far been neglected by design automation: immediate execution, mutual exclusion, and parallel execution. Experiments show that compared with traditional synthesis, this new method shows significant improvements, and the gap between design automation and biology is getting bridged.**

I. Introduction

Microfluidic biochips have become an important platform in performing complex and error-prone bioassays. They can increase throughput by performing assays in parallel [1] and generate products with higher yields compared with traditional microarrays [2]. Since the invention of the first PDMS (Polydimethylsiloxane) pneumatic micro-valve in 2000 [3], flow-based microfluidic biochips have evolved rapidly. It is even possible now to fabricate one million valves in a single chip [4], thus attracting EDA (Electronic Design Automation) researchers' attention to this field. However, the rapid development of microfludic biochips and technological innovation of bioassay protocols always lead to new requirements for the design, which may not easily be fulfilled by existing design autmation methods. This results in a gap between biology and design automation.

Up to now EDA researchers have integrated several types of bioassay operations into the automated synthesis: mixing, detecting, storing, heating, as well as washing and filtering [5] [6] [7] [8]. Thereof washing operations were discussed in EDA research for digital biochips in [9] [10] [11] and first mentioned for flow-based biochips in [12]. But the washing operations mentioned in these works are more like the rinsing or cleaning operations in bioassays, which are performed to prevent contamination of devices. In fact, the washing

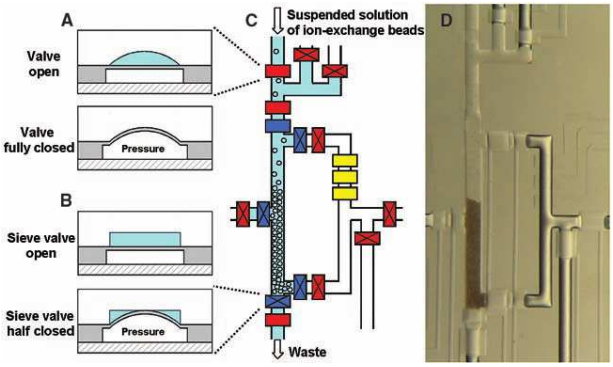

Figure 1: Sieve valve function [15]: (a) Regular valve. (b) Sieve valve. (c) Stacked beads. (d) Snapshot of (c).

operations in bioassay protocols are usually defined as a particular type of operations to extract target particles from the suspension, and thus to increase the target concentrations, or to collect products from earlier operations [13] [14]. The latter use sounds more like the filtering operations mentioned in [5] [6], but this research has not given a detailed description of filtering operations. In the remainder of this paper, when we refer to "washing operation", it indicates the washing operation in bioassay protocols.

Washing operations are essential in bioassays involving cells. After several mixing operations, cells or the reaction products of cells can be diluted to an extremely low concentration, which could be unsuitable for further reaction or observation. In order to extract the target particles from the suspension, biologists perform washing operations using sieve valves.

A Sieve valve is a special PDMS component implemented on the chip. It is a variant of partially closed valves. An example of a sieve valve is shown in Figure 1(b). Unlike the regular valve shown in Figure 1(a), when a sieve valve is closed, there remains a gap that is small enough to stop larger particles, but big enough for tiny particles and fluids to flow through. As shown in Figure 1(c)(d), anion exchange beads are stacked against the closed sieve valve and thus form a bead column. In this manner, the target can be concentrated from a 40 mL volume into less than 400 nL [16] with the same

DATE 2016, DOI: 10.3850/9783981537079_0261
http://ieeexplore.ieee.org/document/7459386/

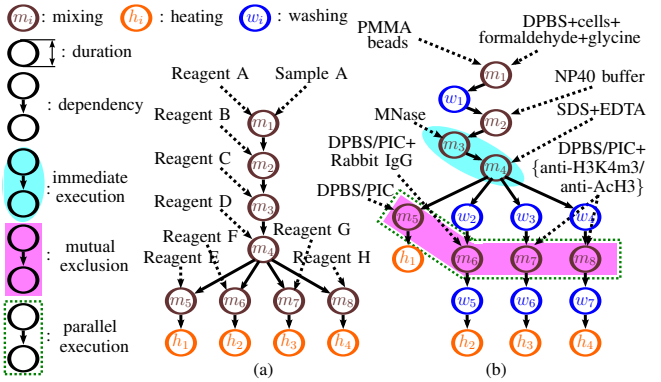

Figure 2: Sequencing graphs of ChIP: (a) Simplified graph. (b) Realistic graph.

amount of cells contained. Thereafter, the highly-concentrated target particles can be easily transported to the next device or platform by opening the sieve valve and pushing the stacked column with a new fluid.

Besides the washing operation, there are some further execution limitations of bioassays that used to be neglected in previous EDA research, which should however be strictly obeyed. We take a ChIP (Chromatin Immunoprecipitation) assay [17] as an example to introduce these execution limitations. ChIP is an assay used to study interactions between protein and DNA. The protocol of this assay is shown as a sequencing graph in Figure 2(b), which may be mistakenly illustrated as Figure 2(a) if washing operations and execution limitations are not considered.

The first execution limitation is *immediate execution*. As shown in Figure 2(b), MNase (Micrococcal Nuclease) is loaded as the input of operation m_3 to digest DNA from cells into fragments. After that, SDS/EDTA (Sodium Dodecyl Sulfate/Ethylenediaminetetraacetic acid) buffer is loaded as the input of operation m_4 to stop DNA digestion and to lyse cells completely. To prevent over-digestion, m_4 is required to be performed right after the completion of m_3; otherwise DNA ends would be harmed and the required dinucleosome signal would disappear [18].

The second execution limitation is *mutual exclusion*. As shown in Figure 2(b), m_4 is the final operation of the chromatin (Ch) process. Its product, the immunoprecipitation (IP)-ready material, is sent to four mixing operations m_5, m_6, m_7, and m_8 to perform the IP process. Thereof only m_7 and m_8 take DPBS/PIC (Dulbecco's phosphate-buffered saline/Protease Inhibitor Cocktail) buffer with target antibodies as their other inputs, which means that target proteins should not be specified in the products of m_5 and m_6. Therefore, these IP operations need to be performed in completely different mixers in ChIP designs [17] [19].

Another execution limitation is *parallel execution*. We also take the IP operations as shown in Figure 2(a) as an example, thereof m_5 is a reference operation and takes no antibody as its input; m_6 takes antibodies that specify no target proteins as negative control; m_7 and m_8 are replicates taking the same target antibodies as their inputs. To achieve a fair comparison, these four IP operations need to be executed in parallel [17] [19].

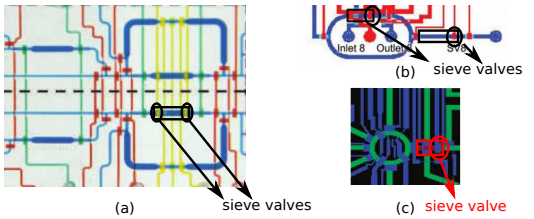

Figure 3: (a) Design for kinase activity assay [22]. (b) Design for ChIP assay [19]. (c) Design for single-cell mRNA isolation assay [16].

II. Background and Problem Formulation

In this section, we first introduce the working principle of sieve valves in more detail, then we discuss the application of washing operations further, and define the problem formulation of this work.

A. Sieve Valve

Sieve valves evolved from partially closed valves [20] and are sometimes called column valves [21]. Since the first appearance of PDMS sieve valves in [15], biologists have developed a mature procedure to connect tiny target cells or RNA/DNAs with large particles, which can be stacked against sieve valves and thus concentrated or collected. This invention has drastically increased the efficiency and accuracy of bioassays.

We take the above mentioned ChIP assay as an example. In the Ch process, target DNA is cross-linked by proteins, and in the IP process, proteins can be specified by antibodies coated on large beads, thus forming large particles that can be stacked against sieve valves. Since the beads are usually magnetic, they can be separated from DPBS buffer by using a magnet [19]. Therefore, target DNA can be easily purified from these beads by either a chemical [19] or a heating [17] operation.

B. From Washing Operation to Washing Behavior

With the help of sieve valves, biochemists can perform washing operations for two typical uses. The first one is to increase the concentration of inputs before a mixing operation, with a sieve valve commonly integrated inside a mixer or close to the mixer entrance. We call this kind of washing operations pre-washing operations. The second one is to collect products after the completion of a mixing operation, with a sieve valve located inside or at the exit of a mixer. We call this kind of washing operations post-washing operations.

Figure 3 shows three sample designs demonstrating the locations of sieve valves and corresponding bead columns, which are marked by circles and rectangles, respectively. The sieve valves shown in Figure 3(a) are integrated inside a mixer for pre-washing operations. Figure 3(b) shows two locations of sieve valves: one is inside a mixer for pre-washing operations, and the other is near the exit of the mixer for post-washing operations. Figure 3(c) shows a sieve valve located near the

exit of a mixer for post-washing operations.

Since a washing operation is usually accompanied by a mixing operation, we transform the washing operation into washing behavior, which does not necessarily occupy exclusive chip area for execution, and can be considered as a specific requirement of a mixing operation. In this way, we can easily control the number of sieve valves that need to be integrated by checking the requirements of the related mixing operations. For example, if several mixing operations requiring pre-washing behaviors are bound to the same mixer, we only need to integrate one sieve valve before this mixer. Therefore, we can reduce the area cost by mapping new mixing operations requiring washing behaviors to existing free mixers with sieve valves, and reduce the time cost by considering the execution time of washing behaviors in our synthesis process.

C. Problem Formulation

Input:

A bioassay sequencing graph, which specifies operation durations, dependency, types, execution limitations including immediate execution, mutual exclusion, and parallel execution. Every washing operation is bound to a mixing operation as either its pre-washing or post-washing behavior.

Objective:

Reduce assay execution time and chip area cost with adjustable coefficients.

Output:

The bioassay synthesis result, which specifies scheduling and binding results of assay operations.

III. Sieve-Valve- and Execution-Limitations-aware Synthesis

In order to obtain an optimal biochip design for a bioassay, we introduce an integer-linear-programming (ILP) model to simulate the assay process. According to the number of operations in an assay, we set up a number of devices in our model, some of which will be removed if the synthesis result shows that no operations are executed in them. The number of such devices is adjustable and represents the maximum number of devices that are allowed to be integrated in the final design. For convenience, we index all the operations and devices.

A. General characteristics of biochemical assays

Before dealing with the new constraints mentioned above, we first introduce some general characteristics of biochemical assays, thereby introducing our modeling methodology.

1) Type consistence

We define a biochemical assay as a series of operations, which can be divided into different types and thus must be executed in corresponding devices. In this work, we deal with three main operation-types: mixing, detecting, and heating, and respectively three device-types: mixer, detector, and heater. In order to make sure that an operation is bound to a device of the proper type, we introduce an operation-device-mapping variable $d\_o_{i,j}$ to our model, which represents whether operation j is bound to device i. If the answer is yes, $d\_o_{i,j}$ is set to 1, otherwise it is set to 0.

We first introduce the following constraint on every operation to ensure that each operation is bound to exactly one device:

$$\sum_{i\in I} d\_o_{i,j} = 1, \tag{1}$$

where I is the index set of all devices.

Then we introduce the following constraints on every device, thereby guaranteeing that operations of two different types will not be bound to the same device:

$$\sum_{m\in M} d\_o_{i,m} \le q_2 \cdot C, \tag{2}$$

$$\sum_{d\in D} d\_o_{i,d} \le q_3 \cdot C, \tag{3}$$

$$\sum_{h\in H} d\_o_{i,h} \le q_4 \cdot C, \tag{4}$$

$$q_2 + q_3 + q_4 \le 1, \tag{5}$$

where M is the index set of all mixing operations, D is the index set of all detecting operations, H is the index set of all heating operations, q_2, q_3, q_4 are auxiliary binary variables, and C is a very large constant. If q_k, $k \in \{2,3,4\}$ is set to 1, constraint (k) becomes trivial. Conversely, if q_k, $k \in \{2,3,4\}$ is set to 0, all the operation-device-mapping variables involved in constraint (k) become 0 automatically. Since at least two of the auxiliary variables will be set to 0, a device will be shut off from at least two types of operations.

2) Operation dependency

In a biochemical assay, the product of a preceding operation o_a can be the input of a later operation o_b. In this situation, we define o_a as the parent operation of o_b and o_b as the child operation of o_a correspondingly. Since an operation can only start with all its inputs ready, a child operation will not start until the completion of all its parent operations.

We model this dependency relationship among operations by introducing a time variable t_i, which represents the start time of operation i. And we introduce the following constraints on each pair of operations connected by this dependency relationship:

$$t_c \ge t_p + dur_p + t_{trans}, \tag{6}$$

where c represents a child operation, whose parent operation is represented by p, dur_p represents the duration of operation p and t_{trans} is a constant representing the transportation time between devices or the preparation time between sequential operations.

3) Non-interfering proceeding

A new operation should be executed in an unused device. If a device is occupied by an operation in progress, other operations should not be bound to this device until the completion of the current operation. Therefore, we introduce following constraints on every two operations:

$$t_a+q_7\cdot C\geq t_b+dur_b+t_{trans}, \tag{7}$$
$$t_a+dur_a+t_{trans}-q_8\cdot C\leq t_b, \tag{8}$$
$$d\_o_{i,a}+d\_o_{i,b}-q_9\cdot C\leq 1,\quad \forall i\in I, \tag{9}$$
$$q_7+q_8+q_9\leq 2, \tag{10}$$

where a, b are the index of operations, t_a and t_b represent the start time of a and b, and therefore $t_a+dur_a+t_{trans}$ and $t_b+dur_b+t_{trans}$ represent the end time of a and b. q_7, q_8, q_9 are auxiliary binary variables. If q_k, $k\in\{7,8,9\}$ is set to 1, constraint (k) becomes trivial. Therefore, above constraints ensure that two operations a and b can only be bound to the same device, when their execution times do not overlap each other (a starts after the completion of b, or a ends before the execution of b).

4) General optimization objective

In order to reduce the time and area cost of our design, we introduce two variables t_e and sum_d, which represent the duration of the whole assay and the number of devices that need to be integrated in the chip, respectively.

Suppose that the first operation in an assay starts at 0s, then the duration of this assay can be regarded as the time the last operation terminates. Therefore, we introduce the following constraints to make sure that every operation will be finished within t_e:

$$t_e\geq t_i+dur_i,\quad \forall i\in O \tag{11}$$

where O is the index set of all the operations.

With our device-mapping variables, we can easily judge whether any operations have been mapped to a device, and thus count the number of devices that need to be integrated. We first introduce the following constraints on each device:

$$d\_o_{i,j}-act_i\leq 0,\quad \forall j\in O, \tag{12}$$

where act_i is an auxiliary binary variable, which will be set to 1, if at least one operation has been mapped to device i. Then we can obtain sum_d by counting the sum of act_is:

$$sum_d=\sum_{i\in I} act_i. \tag{13}$$

Now we can derive an ILP model considering the general characteristics of bioassays and describe it as:

$$\text{Minimize:}\quad t_e\cdot C_{t_e}+sum_d\cdot C_{sum_d}, \tag{14}$$
$$\text{Subject to:}\quad \text{constraints}(1)-(13), \tag{15}$$

where the values of C_{t_e} and C_{sum_d} are adjustable, which helps to control the weight of time and area cost.

B. Sieve-valve-aware synthesis

As mentioned above, besides the general characteristics of bioassays, there are still some practical issues that used to be neglected. In order to stack target particles, experimenters usually perform washing behaviors. Therefore, sieve valves need to be integrated in the chip, which will affect the scheduling results and lead to extra area cost and control effort.

Washing behaviors are performed for two typical uses: either before a mixing operation, for increasing the concentration of inputs; or after a mixing operation, for collecting the products. Therefore, we can refine the execution time of each mixing operation requiring washing behavior by adding it with the execution time of the washing behavior. Then we introduce two binary variables $pre\_w_{i,j}$ and $post\_w_{i,j}$, to represent whether a sieve valve needs to be integrated before or after a device i to which operation j has been mapped. And we introduce the following constraints on each operation that requires a pre- or post-washing behavior, to make sure that such operations can only be bound to devices connected with corresponding sieve valves:

$$if\ j\ requires\ a\ pre-washing\ behavior,$$
$$d\_o_{i,j}=pre\_w_{i,j},\quad \forall i\in I, \tag{16}$$
$$if\ j\ requires\ a\ post-washing\ behavior,$$
$$d\_o_{i,j}=post\_w_{i,j},\quad \forall i\in I. \tag{17}$$

The above constraints mean that for an operation j requiring pre-/post-washing behavior, the value of $d\_o_{i,j}$ and $pre/post\_w_{i,j}$ must stay the same.

Now we can obtain the number of sieve valves that need to be integrated in the same manner as we obtain the number of devices: we first set up a variable sum_s to represent the number of sieve valves, then introduce the following constraints with auxiliary variables $act\_svpre/post_i$ on each device:

$$pre\_w_{i,j}-act\_svpre_i\leq 0,\quad \forall j\in preO, \tag{18}$$
$$post\_w_{i,j}-act\_svpost_i\leq 0,\quad \forall j\in postO, \tag{19}$$

where $pre/postO$ are the index sets of operations requiring pre-/post- washing behaviors. And we obtain sum_s as the sum of auxiliary variables:

$$sum_s=\sum_{i\in I}(act\_svpre_i+act\_svpost_i). \tag{20}$$

Considering the cost of fabricating sieve valves, we modify our optimization objective into the following new one:

$$\text{Minimize:}\quad t_e\cdot C_{t_e}+sum_d\cdot C_{sum_d}+sum_s\cdot C_{sum_s}, \tag{21}$$

where C_{sum_s} is the adjustable weight coefficient of sum_s.

C. Specific execution limitations

The execution of biochemical operations can be limited by different reagent properties and assay objectives, which requires synthesis adaption. Therefore, we modify our model further and provide a synthesis method considering three common practical limitations.

1) Immediate execution

Sometimes, the transition time between sequential operations needs to be strictly controlled to prevent the reagents from overreaction. In these cases, a child operation is required to be performed immediately after the completion of its parent operation. Therefore, we introduce the following constraints on sequential operations requiring immediate execution:

$$t_b=t_a+dur_a+t_{trans}, \tag{22}$$

where b represents the child operation of a. Thus this constraint means that when operation a is finished, operation b will start within an experimenter-definable transition time.

2) Mutual exclusion

Some biochemical operations are mutually exclusive, since the contamination of their reagents may cause serious errors to the assay. Therefore, these operations are supposed to be bound to two different devices, which can be modeled by introducing the following constraint on each device:

$$d\_o_{i,a} + d\_o_{i,b} \leq 1, \tag{23}$$

where a, b are operations with mutual exclusion, which are prevented by this constraint from being bound to the same device i.

3) Parallel execution

Another commonly seen requirement in bioassays is the execution of replicate operations, some of which are performed as control group for reference. In order to provide a fair environment for these operations, they are usually performed in parallel in different devices. Since overlapping operations have been prevented from being bound to the same device, we only need to ensure that these operations start at the same time:

$$t_a = t_b, \tag{24}$$

where a, b are operations requiring parallel execution.

IV. Experimental Results

We implemented the proposed synthesis algorithm in C++ on a computer with a 2.67 GHz CPU. The ILP model was solved by the ILP solver Gurobi [24]. To demonstrate the effectiveness of our method, we applied our method to synthesize four bioassays [17] [19] [22] [23] .

The first test case is a ChIP assay with the protocols from [17] including 12 operations with 7 washing behaviors. As shown in Figure 2(b), operation 4 needs to be executed directely after operation 3 and operations $5,6,7,8$ are supposed to be executed in parallel. The second test case is a kinase activity assay with the protocols from [22]. This case includes 22 operations with 22 washing behaviors, two operations thereof are mutually exclusive and should not be executed in the same device. The third test case is another ChIP assay with more IPs with the protocols from [19]. 28 operations with 23 washing behaviors are included in this assay, thereof 12 operations are supposed to be executed in different devices in parallel. The fourth test case is a 20-single-cell mRNA-to-cDNA synthesis assay with the protocols from [23]. This is a big test case including 80 operations with 60 washing behaviors and correspondingly more execution limitations. If the durations of some assay operations are not specified in the protocols, we arbitrarily assign their values with the same value of the other operations in the assay.

Since the traditional synthesis does not consider the integration of sieve valves and the above mentioned execution limitations, its synthesis result has to be refined to fit the bioassay requirements:

1. We integrate sieve valves before or after devices, which are bound by mixing operations requiring pre- or post-washing behaviors according to the binding result.

2. We refine the scheduling result by taking the execution time of washing behaviors into account.

3. We analyze the synthesis result further, to treat the violations of execution limitations.

• For violations of immediate execution, in order not to mess up the scheduling result, we remap the parent operation to an additional device and reschedule it to make it completely right before the start of its "immediately to be executed" child operation.

• For violations of mutual exclusion, we remap some operations to additional devices, to ensure that all operations with mutual exclusion are mapped to different devices.

• For violations of parallel execution, suppose that the set of operations which need to be executed in parallel is P, and op_l is the operation with the latest start time in P. We let op_l keep its original status and remap the other operations in P to additional devices and reschedule them so that they start simultaneously with op_l.

In the remapping process, we first check whether there is an additional device which is not occupied during the execution time of the to be remapped operation. If there is, we map the operation to this free device to save the area cost. If there isn't any free device available, we add one more additional device to the design and map the operation to this new device.

In the rescheduling process, if operation a is rescheduled, the schedule of its succeeding operations and the schedule of the operations that share the same device with a and start later than a will also be influenced. Therefore, we reschedule these operations as well.

Table I shows the experimental results of our method and the traditional synthesis method. Two groups of data are provided with different emphases on model objective as: area-cost-sensitive and execution-time-sensitive as shown in each row. The meaning of the columns is as following:

$\#o(\#w)$ **:**	number of operations and washing behaviors.
$emp.$**:**	emphasis on model objective.
$\#vio$**:**	numbers of violations of execution limitations.
$\#(d_o+d_a)$**:**	number of devices in the original result applying traditional method and the number of additional devices by refinement.
$\#(s_o+s_a)$**:**	number of sieve valves integrated in the devices in the original result applying traditional method and in the additional devices by refinement.
T_e**:**	total assay execution time.
T_r**:**	program runtime.
$\#d$**:**	number of devices in the result applying our method.
$\#s$**:**	number of sieve valves in the result applying our method.
$\#(d+s)\%$**:**	comparison of the usage of devices and sieve valves.
$T_e\%$**:**	comparison of the execution time.

As shown in Table I, since the traditional method hasn't considered all the necessary constraints during the optimization process, its original synthesis results may bring about numerous violations of execution limitations as shown in column $\#vio$, which will prevent these results from being realized as practical designs. After we refine these results and make them meet the requirements of the test cases, the refined results are still outperformed by our method.

Table I: Result comparison between traditional synthesis and our synthesis under different emphases.

			Traditional Synthesis Method					Our Synthesis Method					
	$\#o(\#w)$	$emp.$	$\#vio$	$\#(d_o+d_a)$	$\#(s_o+s_a)$	T_e	T_r	$\#d$	$\#s$	T_e	T_r	$\#(d+s)\%$	$T_e\%$
ChIP [17]	12(7)	area	7	2+3	1+6	378	3.074	5	6	301	1.926	91.67%	79.63%
(4 parallel IPs)		time	8	4+3	1+6	321	5.059	7	6	235	4.057	92.86%	73.21%
kinase	22(22)	area	1	1+1	2+2	755	30.489	2	3	835	30.263	83.33%	110.60%
activity [22]		time	1	3+1	3+2	360	0.759	4	6	274	1.026	111.11%	76.11%
ChIP [19]	28(23)	area	11	2+11	1+22	1775	30.475	13	22	565	30.486	97.22%	31.83%
(16 parallel IPs)		time	24	6+11	1+22	387	5.541	16	22	288	5.434	95%	74.42%
mRNA-to-cDNA	80(60)	area	48	2+38	2+38	1285	103.040	21	22	1780	101.424	53.75%	138.52%
synthesis [23]		time	57	15+38	12+38	493	113.135	50	46	88	100.989	93.20%	17.85%

Under consideration of washing behaviors and execution limitations, our method maximizes the utilization of devices and sieve valves. In some cases, when area (time) cost is emphasized, the time (area) cost may be slightly increased as a trade-off, since the weight coefficient of area (time) cost is much larger. Within similar program runtime, our method provides results with less area cost (fewer devices or fewer sieve valves) under area-cost-sensitive setting, and shortens the assay execution time under time-cost-sensitive setting in all four test cases. When dealing with the biggest test case, the area cost is nearly halved and the time cost is cut to less than $1/5$ under corresponding settings.

V. Conclusion

In this paper we try to narrow the gap between biology and design automation for microfluidic biochips by proposing a synthesis method considering sieve valves along with three biological execution limitations (immediate execution, mutual exclusion, and parallel execution) for the first time in the design automation field. We build an ILP model for the synthesis and perform the optimization, by which scheduling and binding results can be specified. Experimental results show that our method provides good solutions considering time and area cost, without any violation of execution limitations.

References

[1] R. A. White III, P. C. Blainey, H. C. Fan, and S. R. Quake. Digital PCR provides sensitive and absolute calibration for high throughput sequencing. *BMC Genomics*, 2009.

[2] C.-C. Lee, T. M. Snyder, and S. R. Quake. A microfluidic oligonucleotide synthesizer. *Nucleic Acids Research*, 38(8):2514–2521, 2010.

[3] M. A. Unger, H.-P. Chou, T. Thorsen, A. Scherer, and S. R. Quake. Monolithic microfabricated valves and pumps by multilayer soft lithography. *Science*, 288(5463):113–116, 2000.

[4] I. E. Araci and S. R. Quake. Microfluidic very large scale integration (mvlsi) with integrated micromechanical valves. *Lab on a Chip*, 12:2803–2806, 2012.

[5] W. H. Minhass, P. Pop, and J. Madsen. System-level modeling and synthesis of flow-based microfluidic biochips. In *Proc. Int. Conf. Compil., Arch. and Syn. Embed. Sys.*, pages 225–234, 2011.

[6] W. H. Minhass, P. Pop, J. Madsen, and F. S. 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.

[7] 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. Phy. Des.*, pages 123–129, 2013.

[8] T.-M. Tseng, B. Li, T.-Y. Ho, and U. Schlichtmann. Reliability-aware synthesis for flow-based microfluidic biochips by dynamic-device mapping. In *Proc. Design Autom. Conf.*, pages 141:1–141:6, 2015.

[9] Y. Zhao and K. Chakrabarty. Synchronization of washing operations with droplet routing for cross-contamination avoidance in digital microfluidic biochips. In *Proc. Design Autom. Conf.*, pages 635–640, 2010.

[10] T.-W. Huang, C.-H. Lin, and T.-Y. Ho. A contamination aware droplet routing algorithm for the synthesis of digital microfluidic biochips. *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, 29(11):1682–1695, 2010.

[11] Q. Wang, Y. Shen, H. Yao, T.-Y. Ho, and Y. Cai. Practical functional and washing droplet routing for cross-contamination avoidance in digital microfluidic biochips. In *Proc. Design Autom. Conf.*, pages 143:1–143:6, 2014.

[12] K. Hu, T.-Y. Ho, and K. Chakrabarty. Wash optimization for cross-contamination removal in flow-based microfluidic biochips. In *Proc. Asia and South Pacific Des. Autom. Conf.*, pages 244–249, 2014.

[13] A. K. White, M. VanInsberghe, O. I. Petriv, M. Hamidi, D. Sikorski, M. A. Marra, J. Piret, S. Aparicio, and C. L. Hansen. High-throughput microfluidic single-cell RT-qPCR. *Proc. Natl. Acad. Sci.*, 108(34):13999–14004, 2011.

[14] N. Vergauwe, S. Vermeir, J. B. Wacker, F. Ceyssens, M. Cornaglia, R. Puers, M. A. M. Gijs, J. Lammertyn, and D. Witters. A highly efficient extraction protocol for magnetic particles on a digital microfluidic chip. *Sensors and Actuators B: Chemical*, pages 282–291, 2014.

[15] C.-C. Lee, G. Sui, A. Elizarov, C. J. Shu, Y.-S. Shin, A. N. Dooley, J. Huang, A. Daridon, P. Wyatt, D. Stout, H. C. Kolb, O. N. Witte, N. Satyamurthy, J. R. Heath, M. E. Phelps, S. R. Quake, and H.-R. Tseng. Multistep synthesis of a radiolabeled imaging probe using integrated microfluidics. *Science*, 310(5755):1793–1796, 2005.

[16] J. S. Marcus, W. F. Anderson, and S. R. Quake. Microfluidic single-cell mRNA isolation and analysis. *Anal. Chem.*, 78:3084–3089, 2006.

[17] A. R. Wu, J. B. Hiatt, R. Lu, J. L. Attema, N. A. Lobo, I. L. Weissman, M. F. Clarke, and S. R. Quake. Automated microfluidic chromatin immunoprecipitation from 2,000 cells. *Lab on a Chip*, 9:1365–1370, 2009.

[18] O. Flores, Ö. Deniz, M. Soler-López, and M. Orozco. Fuzziness and noise in nucleosomal architecture. pages 1–13, 2014.

[19] A. R. Wu, T. L. A. Kawahara, N. A. Rapicavoli, J. van Riggelen, E. H. Shroff, L. Xu, D. W. Felsher, H. Y. Chang, and S. R. Quake. High throughput automated chromatin immunoprecipitation as a platform for drug screening and antibody validation. *Lab on a Chip*, 12:2190–2198, 2012.

[20] J. W. Hong, V. Studer, G. Hang, W. F. Anderson, and S. R. Quake. A nanoliter-scale nucleic acid processor with parallel architecture. *Nature Biotechnology*, 22(4):435–439, 2004.

[21] Y. Huang, P. Castrataro, C.-C. Lee, and S. R. Quake. Solvent resistant microfluidic DNA synthesizer. *Lab on a Chip*, 7:24–26, 2007.

[22] C. Fang, Y. Wang, N. T. Vu, W.-Y. Lin, Y.-T. Hsieh, L. Rubbi, M. E. Phelps, M. Müschen, Y.-M. Kim, A. F. Chatziioannou, H.-R. Tseng, and T. 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.

[23] J. F. Zhong, Y. Chen, J. S. Marcus, A. Scherer, S. R. Quake, C. R. Taylor, and L. P. Weiner. A microfluidic processor for gene expression profiling of single human embryonic stem cells. *Lab on a Chip*, 8(1):68–74, 2008.

[24] Gurobi Optimization, Inc. Gurobi optimizer reference manual, 2013.