PAPER

Mixer-Based Washing Methods for Programmable Microfluidic Devices

Yifang BAO[†a)], ***Nonmember***, **Shigeru YAMASHITA**[††b)], ***Senior Member***, **Bing LI**[†††c)], ***and*** **Tsung-Yi HO**[††††d)], ***Nonmembers***

SUMMARY When we use a Programmable Microfluidic Device (PMD), we need to wash some contaminated area to use the chip for further experiments. Recently, a novel washing technique called Block-Flushing has been proposed. Block-Flushing washes contaminated area in PMDs by using buffer flows. In Block-Flushing, we need to keep a buffer flow from an input port to an output port of a PMD for a long period to dissolve residual contaminants. Thus, we may need a lot of buffer fluids and washing time even if the contaminated area is small. Another disadvantage of the washing method by Block-Flushing is such that we may not able to clean residual contaminants at valves completely by only buffer flows. To address the above-mentioned issues, this paper proposes a totally new idea to wash PMDs; our method does not use buffer flows, but washes contaminated area by using mixers. By using a mixer, we can dissolve residual contaminants at valves in the area of the mixer very efficiently. In this paper, we propose two methods to wash PMDs by using mixers. The first method can wash the whole chip area by using only four times of a single 2x2-mixer time. We also propose the second method which is a heuristic to reduce the number of moving valves because valves may wear down if they are used many times. We also show some experimental results to confirm that the second method can indeed decrease the number of used valves.
key words: *programmable microfluidic devices, washing, mixer*

1. Introduction

Programmable Microfluidic Devices (PMDs) [1], [2] have emerged as a new architecture for next-generation flow-based biochips. A PMD can be dynamically reconfigured to execute different bioassays flexibly and efficiently by appropriately opening or closing two-dimensional regularly-arranged valves. During execution of a bioassay or between the execution of multiple bioassays, some areas on the PMD become contaminated and must be cleaned similar to the conventional biochemical experiments [3].

Because a PMD can be dynamically reconfigured to perform various types of experiments, we need to wash a PMD after some experiments to use contaminated area again. It has been proposed to flush contaminated area by *buffer flows* to wash a PMD. Specifically, a washing method has been proposed [4] for traditional flow-based biochips. Recently, a washing algorithm dedicated to PMDs called Block-Flushing [5] has been proposed. Although it is natural to wash flow-based biochips by using buffer flows, there are potential problems in the previous washing methods.

First, we need to keep a buffer flow for a certain time to dissolve residual contaminants along with the flow. This may be efficient if the contaminated area is almost the whole chip. However, suppose the contaminated area is very small and at the center of the chip; it is obviously wasteful of time and buffer liquids to keep a long buffer flow crossing the entire chip for a certain time. Second, when we keep a buffer flow for a certain time to wash a chip, we keep every valve open/close; we do not move valves. Thus, we may not able to wash residual contaminants at valves by only buffer flows although we can surely wash cells by buffer flows.

Our contribution. Considering the above-mentioned issues, we propose a totally new approach to wash PMDs. Our approach has the following two features.

- We consider to wash valves while the previous methods consider to wash cells.
- We utilize *mixers* to wash valves while the previous methods use buffer flows.

Our approach can clean both cells and valves, and thus we can naturally solve the problem such that there may be residual contaminants at valves after cleaning a PMD by only buffer flows. Also, we can expect that our methods need less amount of buffer fluids than the previous methods to clean a PMD chip because our methods do not need to keep a buffer flow for a certain time unlike the previous methods.

We present two concrete methods to use mixers for washing PMDs. The first method can finish washing a PMD for at most 4 times of 2x2-mixing time with additional time to load buffer fluids, and thus the method should be very efficient in terms of processing time compared to the previous methods. However, the first method may open/close valves many times; it may not be good if we consider the current devices such that some valves may wear down after using many times. So, we propose another method which opens/closes valves less frequently than the first method. We compare the two methods by using randomly generated benchmarks, and confirm that the second method indeed can decrease the number of opening/closing valves in total.

Manuscript received September 10, 2021.
Manuscript revised February 17, 2022.
Manuscript publicized April 18, 2022.
[†]The author is with KuanDeng (Beijing) Science And Technology Ltd., China.
[††]The author is with College of Information Science and Engineering, Ritsumeikan University, Kusatsu-shi, 525-8577 Japan.
[†††]The author is with Technical University of Munich, Germany.
[††††]The author is with the Department of Computer Science of National Tsing Hua University, Beijing, China.
a) E-mail: mori@ngc.is.ritsumei.ac.jp
b) E-mail: ger@cs.ritsumei.ac.jp
c) E-mail: b.li@tum.de
d) E-mail: ho.tsungyi@gmail.com
DOI: 10.1587/transfun.2021EAP1114

This paper is organized as follows. After providing necessary information for PMDs in Sect. 2, we propose our idea to use mixers to wash PMDs in Sect. 3. We also propose two methods based on our idea in the same section. Then, Sect. 4 shows the experimental result to compare the two methods in terms of the total number of used valves. Finally, Sect. 5 concludes the paper.

2. Preliminary

In this section, we provide necessary information for PMDs.

2.1 Programmable Microfluidic Devices (PMDs)

Figure 1(a) shows the model of a PMD; it consists of two-dimensionally arranged cells. There is a valve between two adjacent cells. Thus, as Fig. 1(b) shows, each cell is surrounded by four valves. We can place input ports and output ports at any cell around the surroundings of a PMD. Then, we can make a flow path dynamically from an input port to an output port by opening or closing each valve appropriately.

For example, in Fig. 1(c), yellow valves are opened, and green valves are closed. Then, we can have a path from the input port “In” to the output port “Out”; by adding air pressure, we can make a liquid flow as the blue line in Fig. 1(c). Thus, we can transport fluids into a PMD, and also we can get fluids out from a PMD.

We can also mix fluids in a PMD. For example, Fig. 2 shows so-called a “2x2-mixer” consisting of 2x2 cells. When we open and close horizontal and vertical valves in the mixer alternatively as Fig. 2, a circulating flow occurs; we can mix the fluid in the mixer. Thus, we can make a mixer dynamically at any place in a PMD. Indeed, by opening/closing some valves appropriately, both the transportation and mixing functions can be implemented dynamically, as shown in the videos in [6], [7].

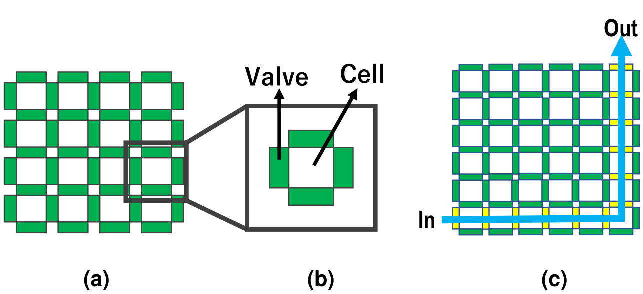


Fig. 1 (a) a PMD (b) a cell and valves (c) making a flow path.

2.2 Washing PMDs by Flushing Operations

After some operations are executed on a PMD, some areas on the chip become contaminated. In order to wash the contaminated area, a buffer flow is applied to flush the contaminants. We explain how we can wash PMDs in the following by using an example as shown in Fig. 3.

Let us suppose that the red cells as shown in Fig. 3(a) are contaminated after performing some experiments. To wash the cells in the red region, previous works consider to make a buffer flow from “In” to “Out” (the light blue line in Fig. 3(b)) to flush the contaminants in the red cells. To do so, we need to keep the buffer flow for a certain time until we dissolve residual contaminants in the red cells. We do not know the exact necessary time to dissolve residual contaminants for a specific case. However, we can consider that it may be wasteful to keep a long buffer flow across an entire chip for some time if we want to clean only few cells.

3. Proposed Methods

In this section, we first discuss some potential problems of washing PMDs by flushing operations. Then, we explain our

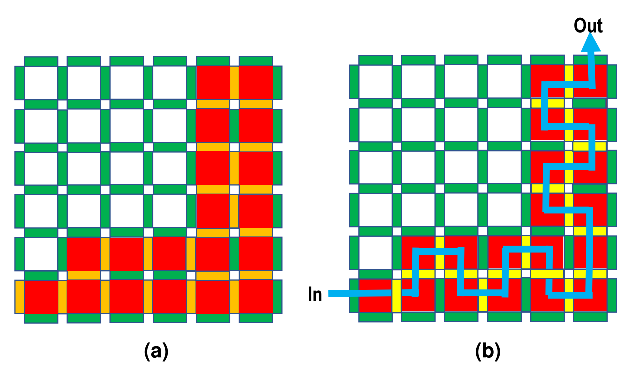


Fig. 3 Washing a PMD by a buffer flow.

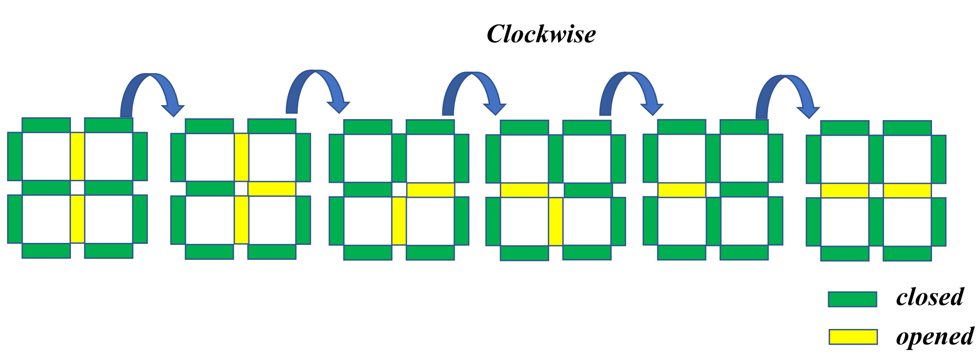


Fig. 2 A 2x2-mixer.

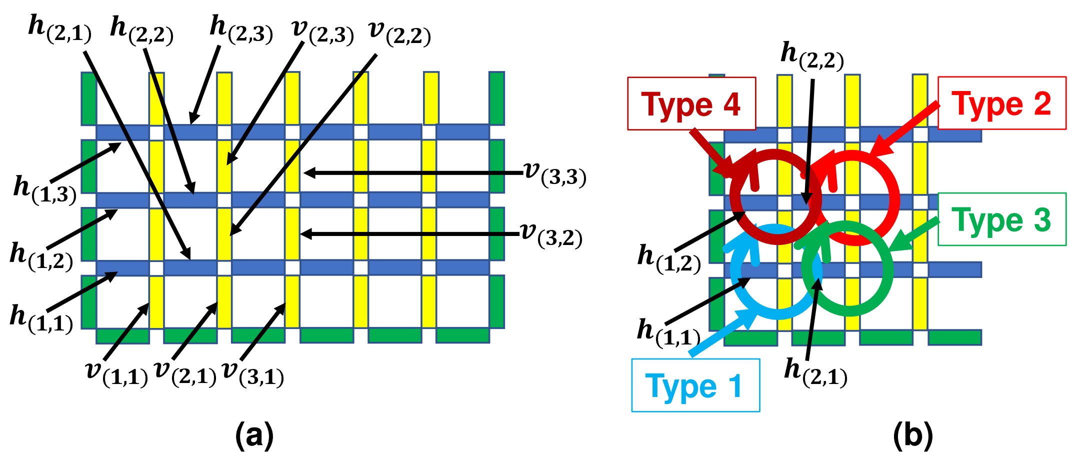


Fig. 4 Notations used in the proposed methods.

idea to use many mixers simultaneously to wash PMDs. To utilize our idea, we propose two concrete methods, but we would like to note that we can consider other methods based on our idea.

3.1 Potential Problems of Washing by Flushing Operations

As we mentioned in the previous section, we usually wash contaminated cells by a buffer flow. This seems to be a very natural way, but we consider that there are two potential problems as follows.

- We need to make a buffer flow from an input port to an output port of a PMD for a long period to dissolve all residual contaminants in the chip. Thus, we may need a lot of buffer fluids and washing time even if the contaminated area is small.
- If we only consider to wash cells, we may fail to clean some contaminated valves. For example, suppose the red cells and the orange valves in Fig. 3(a) are contaminated after performing some experiments. Then the previous works consider to wash contaminated area by buffer flows. For example, we may use a buffer flow as shown as the light blue line in Fig. 3(b); we close the green valves in the red region in Fig. 3(b), and in such a case we cannot clean all the green contaminated valves in the red region at the same time because they are closed.

3.2 Problem Formulation: Cleaning Valves with Mixers

To address the above-mentioned problems, we propose a totally new idea to wash PMDs; our method does not use buffer flows unlike previous methods, but washes contaminated area by using mixers. By using a mixer, we can dissolve residual contaminants at valves and cells in the mixer very efficiently in terms of the necessary amount of buffer liquids.

Our problem formulation is as follows.

Input. We are given a set of contaminated valves in a PMD.

Task. Our task is to determine which mixers in the PMD should be used to wash all the contaminated valves.

For example, after performing the experiment as shown in Fig. 3(a), the red valves in Fig. 3(a) may be contaminated, and our problem is to wash these red valves. Note that the red cells in Fig. 3(b) are also contaminated, but we do not need to consider the contaminated cells in our method; we consider only valves in our method because the red cells in Fig. 3(b) can be washed automatically when we wash the red valves in Fig. 3(a) by mixers.

3.3 Method 1: Cleaning with 2x2-Mixers

We explain our first method which washes a PMD by using 2x2-mixers. To explain the method, let us introduce some notations to denote valves by using Fig. 4(a). We classify valves into two types: a valve is called *vertical* or *horizontal* if it is placed in a vertical or horizontal direction, respectively. For example, the yellow and the dark blue valves in Fig. 4(a) are vertical and horizontal valves, respectively. Then, as the figure shows, the lower left vertical and horizontal valves are denoted by $v_{(1,1)}$ and $h_{(1,1)}$, respectively. Then, a vertical (horizontal) valve is denoted as $v_{(i,j)}$ ($h_{(i,j)}$), if it is placed at the (i, j)-positions from $v_{(1,1)}$ ($h_{(1,1)}$) as shown in Fig. 4(a).

In our first method, we classify 2x2-mixers into the following four types by the index of the left horizontal valve $h_{(i,i)}$ in mixers.

Type 1 if i and j are both odd numbers.
Type 2 if i and j are both even numbers.
Type 3 if i and j are even and odd numbers, respectively.
Type 4 if i and j are odd and even numbers, respectively.

For example, the blue 2x2-mixer in Fig. 4(b) is Type 1 because its left horizontal valve is $h_{(1,1)}$. In the same figure, the green 2x2-mixer is Type 3 because $h_{(2,1)}$ is the left horizontal valve in the mixer.

Now we are ready to explain how our first method determines the necessary 2x2-mixers to wash the given set of valves.

Step 1: Select Type 1 mixers which contain at least one of

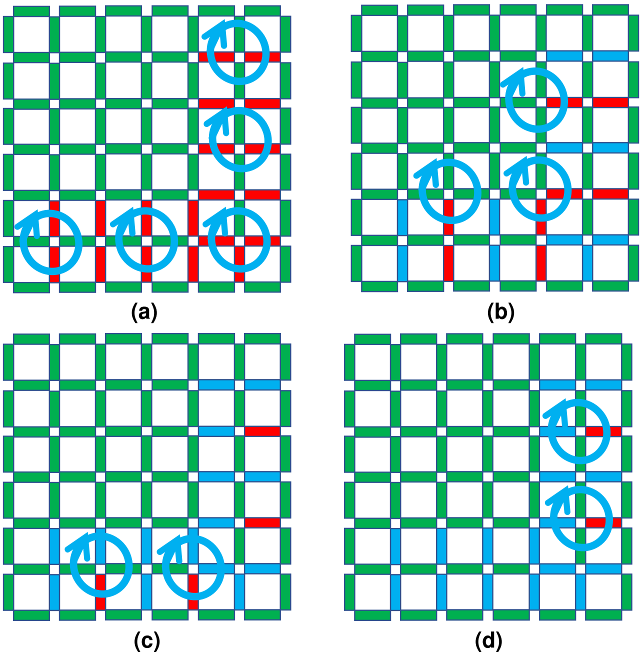


Fig. 5 Method 1: cleaning with 2x2-mixers.

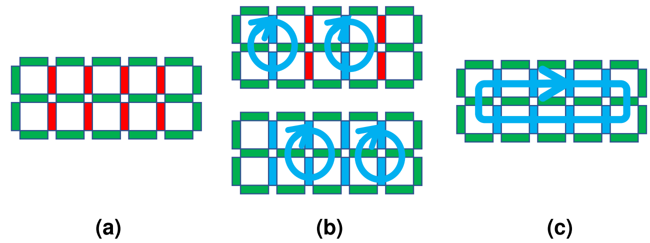


Fig. 6 Comparison between 2×2-mixers and a 2×5-mixer.

Algorithm 1 Method 2: Cleaning with Mixers of Various Sizes

Require: *ContaminatedValves*: the set of contaminated valves.
CleaningMixerList = empty
while *ContaminatedValves* is not empty **do**
  Select the mixer with the largest CE value. ▹ If there are more than one mixer with the largest CE value, select the largest mixer.
  Add the selected mixer to CleaningMixerList.
  Remove valves washed by the selected mixer from *ContaminatedValves*.
end while

the contaminated valves. Remove valves covered by the selected mixers from the set of contaminated valves. If the set becomes empty, we finish the procedure.

Step 2: Select Type 2 mixers which contain at least one of the contaminated valves. Remove valves covered by the selected mixers from the set of contaminated valves. If the set becomes empty, we are done.

Steps 3 and 4: We select Type 3 mixers at Step 3, and Type 4 mixers at Step 4 in the same way as Steps 1 and 2. Note that the set of contaminated valves should become empty after Step 4 even in the worst case.

Note that all the mixers of the same type can be performed at the same time. Therefore, our method can finish washing a PMD for at most 4 times of 2x2-mixing time with additional time to load buffer fluids. In our problem formulation, we do not consider the loading time because the loading time would be much smaller than the necessary time to perform a mixer to dissolve residual chemicals in valves and cells [1].

For example, when we want to wash the red valves in Fig. 5(a), at Step 1 we select five Type 1 2x2-mixers as shown in Fig. 5(a). Then, the remaining contaminated valves are as shown in red in Fig. 5(b). Thus, at Step 2, we select the three Type 2 mixers as shown in Fig. 5(b). After that we select Type 3 and Type 4 mixers as shown in Fig. 5(c) and (d), respectively.

Note that if we change the order of types of mixers to select, the total number of mixers may become different. To consider such orders is out of the topic of this paper; we consider that would be one of our future work.

3.4 Method 2: Cleaning with Mixers of Various Sizes

If we close and open a valve many times, the valve may wear down [8]. Thus, we want to minimize the switching activities of opening/closing valves. To tackle this problem, we introduce a notion called "Cleaning Efficiency (CE)" in the following paragraph.

Suppose we want to wash the red valves in Fig. 6(a). If we wash them by using 2x2-mixers, we need to use four 2x2-mixers as shown in Fig. 6(b). Each 2x2-mixer opens/closes four valves to clean two valves. In such a case, we say the Cleaning Efficiency (CE) value of the mixer is 2/4 (= 1/2) because it washes two valves by moving four valves.

Since we want to decrease the number of moving valves, we want to use a mixer whose CE value is large. To wash the red valves in Fig. 6(a), we can use a 2x5-mixer as shown in Fig. 6(c); the CE value of the mixer is 8/10. Indeed the total numbers of used valves are 16 and 10 in Fig. 6(b) and Fig. 6(c), respectively; we consider it is better to use a 2x5-mixer for this example.

The method proposed in Sect. 3.3 uses only 2x2-mixers, so it may use a mixer whose CE value is only 1/4 in the worst case. Thus, in order to use a mixer with a large CE value, we propose our second method to use mixers of various sizes in the following. Our second method is a simple greedy heuristic as described in Algorithm 1.

For example, when we want to wash the red valves in Fig. 7(a), we select mixer B in Fig. 7(b) as the first mixer because the CE value of mixer B is 100%. Note that we show only some of rectangular mixers whose CE values are large in Fig. 7(b). After selecting mixer B, we update *ContaminatedValves*, and then we update CE values of mixers based on the updated *ContaminatedValves*. Next, we select mixer A as the second mixer based on the CE values.

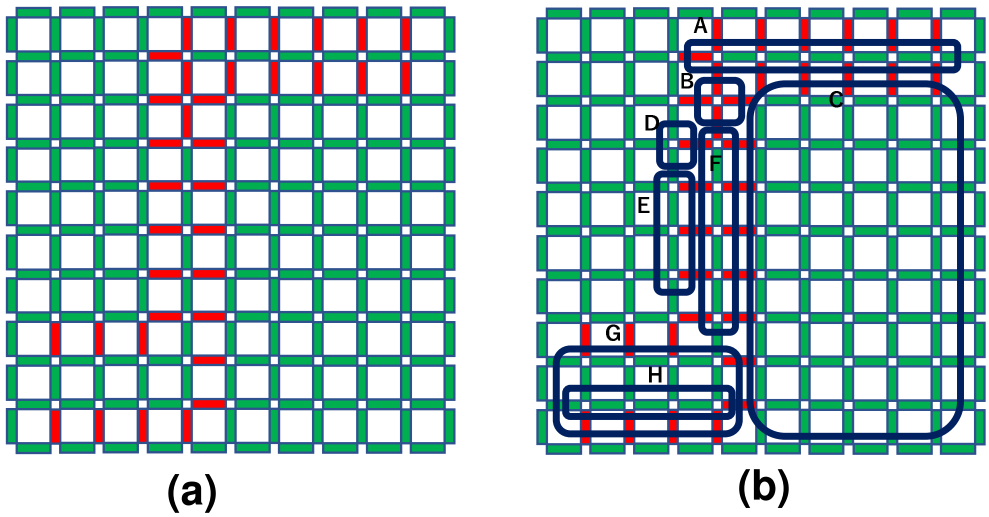


Fig. 7 An example of cleaning with mixers of various sizes.

Table 1 Comparison of the cleaning efficiency between the two methods.

Test	♯ Contaminated Valves	♯ Used Valves		Cleaning Efficiency (%)		CPU time (ms)	
		Method 1	*Method 2*	*Method 1*	*Method 2*	*Method 1*	*Method 2*
Ex. 1	20	52	24	0.38	0.83	29	29
Ex. 2	25	72	36	0.35	0.69	27	32
Ex. 3	22	64	40	0.34	0.55	26	36
Ex. 4	31	88	42	0.35	0.74	27	32
Ex. 5	20	48	28	0.41	0.71	28	35
Ex. 6	27	72	46	0.38	0.59	27	38
Ex. 7	30	88	42	0.34	0.71	27	35
Ex. 8	31	84	54	0.37	0.57	39	45
Ex. 9	30	80	50	0.38	0.60	31	35
Ex. 10	33	96	54	0.34	0.61	29	47
Ex. 11	40	104	58	0.38	0.69	28	40
Ex. 12	51	128	80	0.40	0.64	30	50
Ex. 13	45	128	70	0.35	0.64	30	43
Ex. 14	46	120	62	0.38	0.74	30	43
Ex. 15	42	112	64	0.38	0.83	29	44
Average	33	89	50	0.37	0.67	26	34.2

We continue this procedure until *ContaminatedValves* becomes empty. Note that the CE value of a mixer may change after we select one mixer. For example, the CE value of mixer A is 13/14 and 12/14 before and after selecting mixer B, respectively.

In our method, we only consider rectangular mixers. However, any closed loop may be used essentially as a mixer in a PMD. Therefore, it would be interesting to consider complex shapes for mixers to clean PMDs with fewer number of moving valves.

4. Simulation Results

As mentioned above, our approach is totally different from the previous methods, i.e., Block-Flushing [5]. Because we do not use buffer flows, we can expect that our method can perform washing by less amount of buffer fluids than Block-Flushing. Also, our method can explicitly wash contaminated valves which are not considered in the previous works. Thus, our approach would be a new and efficient strategy to wash PMDs.

Therefore, we do not need any experiment to compare our approach with the previous methods. However, it is not obvious how our proposed two methods differ. Thus, we show some simulation results to check the CE values by the proposed two methods, namely, Method 1 (Cleaning with 2x2-Mixers) described in Sect. 3.3 and Method 2 (Cleaning with Mixers of Various Sizes) described in Sect. 3.4.

We manually generated 15 experiments on a 10x10 PMD. Then, we applied two proposed methods to select mixers to clean the contaminated valves for the 15 test cases. The experimental results are shown in Table 1. The second column shows the number of contaminated valves to be washed. The third and the forth columns report the total number of valves in the selected mixers by Method 1 and Method 2, respectively. Thus, the fifth and the sixth columns show the average CE value among all the mixers to wash by Method 1 and Method 2, respectively. The program was implemented in C++, and run on Windows 10 (Intel Core i7-6700HQ @ 2.60 GHz). The CPU time (ms) to determine all the mixers by Method 1 and Method 2 are reported in the seventh and the eighth columns, respectively.

In Method 2, we need to enumerate all the possible rectangles. To do so, our current implementation is very naive; our method simply generates lines for connected contaminated valves, and then enumerate rectangles which contains at least one of those lines, and we try from the rectangle with the highest CE value as described in Algorithm 1. However, since the PMD chip size is not so large, we expect that this naive implementation may not need much CPU time, which was indeed confirmed by the experiment. As for the comparison between Method 1 and Method 2 in terms of the CE values, we confirm that Method 2 is much better than Method 1.

As for the washing time, we consider that Method 2 is comparable to Method 1 by the following reason. If there is no overlap between two mixers, we can perform the two mixers at the same time. Thus, for most cases, many mixers obtained by Method 2 can be performed simultaneously. For example, the total necessary steps to use mixers simultaneously is only two for the example in Fig. 7. Indeed the total necessary steps to use mixers by Method 2 is two or three in our experiment. Thus we consider that Method 2 is comparable to Method 1 in terms of the necessary time although a larger mixer should need more time than a 2x2-mixer to wash cells and valves.

5. Conclusion

In this paper, we have proposed a unique idea to wash a PMD by using mixers whereas previous methods wash PMDs by buffer flows. By utilizing our idea, we expect that we can wash a PMD faster and with much less amount of buffer fluids. Also our approach can clean residual contaminants at valves completely unlike the previous methods.

Based on the idea, we propose two methods to wash PMDs by using mixers. The first method is based on only 2x2-mixers, and it can wash even the whole chip area for only four times of a single 2x2-mixer time. We also propose the second method which is a heuristic to reduce the number of moving valves.

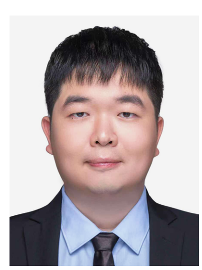
We consider that there are some rooms to improve both methods. For example, as for the first method, the efficiency of washing becomes obviously different by the order of types of mixers to select. Thus, we should consider the order in our future work. As for the second method, we only consider rectangular shapes for mixers. In PMDs, we can make mixers of different shapes. To consider various shapes for mixers could improve the second method. Because we do not know exactly how much time and how much buffer fluids are needed to clean a PMD chip for a specific case, we cannot provide a quantitative comparison between our method and Block-Flushing in this paper. Thus, to analyze the performance of our mixer-based washing strategy quantitatively by measuring experimental data on a real PMD chip is left to a future work.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number 20H04160.

References

[1] L.M. Fidalgo and S.J. Maerkl, “A software-programmable microfluidic device for automated biology,” Lab. Chip., vol.11, no.9, pp.1612–1619, 2011.
[2] J. Melin and S.R. Quake, “Microfluidic large-scale integration: The evolution of design rules for biological automation,” Annu. Rev. Biophys. Biomol. Struct., vol.36, no.1, pp.213–231, 2007.
[3] G.M. Whitesides, E. Ostuni, S. Takayama, X. Jiang, and D.E. Ingber, “Soft lighography and biochemistry,” Annu. Rev. Biomed. Eng., vol.3, no.1, pp.335–373, 2001.
[4] K. Hu, T. Ho, and K. Chakrabarty, “Wash optimization and analysis for cross-contamination removal under physical constraints in flow-based microfluidic biochips,” IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol.35, no.4, pp.559–572, April 2016.
[5] Y.-H. Lin, T.-Y. Ho, B. Li, and U. Schlichtmann, “Block-flushing: A block-based washing algorithm for programmable micofluidic devices,” Design Automation Test in Europe Conference, pp.1531–1536, 2019.
[6] L.M. Fidalgo and S.J. Maerkl, “Fluid transportation in PMD,” http://www.rsc.org/suppdata/lc/c0/c0lc00537a/videos2.mov, 2011.
[7] L.M. Fidalgo and S.J. Maerkl, “Mixing operation in PMD,” http://www.rsc.org/suppdata/lc/c0/c0lc00537a/videos3.mov, 2011.
[8] T.-M. Tseng, B. Li, T.-Y. Ho, and U. Schlichtmann, “Reliability-aware synthesis for flow-based microfluidic biochips by dynamic-device mapping,” Design Automation Conference, pp.1–6, 2015.
[9] T.-M. Tseng, B. Li, T.-Y. Ho, and U. Schlichtmann, “Storage and caching: Synthesis of flow-based microfluidic biochips,” IEEE Des. Test., vol.32, no.6, pp.69–75, Dec. 2015.



Yifang Bao received the B.E. degrees in Information Science and Engineering from Ritsumeikan University in 2018. He also received the B.E. degrees in Information Science and Engineering from Dalian University of Technology in 2018. He received the M.E. degrees in Information Science and Engineering from Graduate School of Information Science and Engineering, Ritsumeikan University, Shiga, Japan, 2020. He is currently with KuanDeng (Beijing) Science And Technology Ltd.

Shigeru Yamashita is a professor at College of Information Science and Engineering, Ritsumeikan University. He received his B.E., M.E. and Ph.D. degrees in Information Science from Kyoto University, Kyoto, Japan, in 1993, 1995 and 2001, respectively. His research interests include new types of computation and logic synthesis for them. He received the 2000 IEEE Circuits and Systems Society Transactions on Computer-Aided Design of Integrated Circuits and Systems Best Paper Award, SASIMI 2010 Best Paper Award, 2010 IPSJ Yamashita SIG Research Award, and 2010 Marubun Academic Achievement Award of the Marubun Research Promotion Foundation. He is a senior member of IEEE, and a member of IPSJ.

Bing Li received the Dr.-Ing. degree from Technical University of Munich (TUM), Munich, Germany, in 2010 and finished the Habilitation there in 2018. 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 as well as neuromorphic computing. He has served on the Technical Program Committee of major conferences including DAC, ICCAD, DATE, ASP-DAC, etc.

Tsung-Yi Ho received his Ph.D. in Electrical Engineering from National Taiwan University in 2005. He is a Professor with the Department of Computer Science of National Tsing Hua University, Hsinchu, Taiwan. His research interests include several areas of computing and emerging technologies, especially in design automation of microfluidic biochips. He has been the recipient of the Invitational Fellowship of the Japan Society for the Promotion of Science (JSPS), the Humboldt Research Fellowship by the Alexander von Humboldt Foundation, the Hans Fischer Fellowship by the Institute of Advanced Study of the Technische Universität München, and the International Visiting Research Scholarship by the Peter Wall Institute of Advanced Study of the University of British Columbia. He was a recipient of the Best Paper Awards at the VLSI Test Symposium (VTS) in 2013 and IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems in 2015. He served as a Distinguished Visitor of the IEEE Computer Society for 2013–2015, a Distinguished Lecturer of the IEEE Circuits and Systems Society for 2016–2017, the Chair of the IEEE Computer Society Tainan Chapter for 2013–2015, and the Chair of the ACM SIGDA Taiwan Chapter for 2014–2015. Currently, he serves as the Program Director of both EDA and AI Research Programs of Ministry of Science and Technology in Taiwan, VP Technical Activities of IEEE CEDA, an ACM Distinguished Speaker, and Associate Editor of the ACM Journal on Emerging Technologies in Computing Systems, ACM Transactions on Design Automation of Electronic Systems, ACM Transactions on Embedded Computing Systems, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, and IEEE Transactions on Very Large Scale Integration Systems, Guest Editor of IEEE Design & Test of Computers, and the Technical Program Committees of major conferences, including DAC, ICCAD, DATE, ASP-DAC, ISPD, ICCD, etc. He is a Distinguished Member of ACM.