Hamming-Distance-Based Valve-Switching Optimization for Control-Layer Multiplexing in Flow-Based Microfluidic Biochips

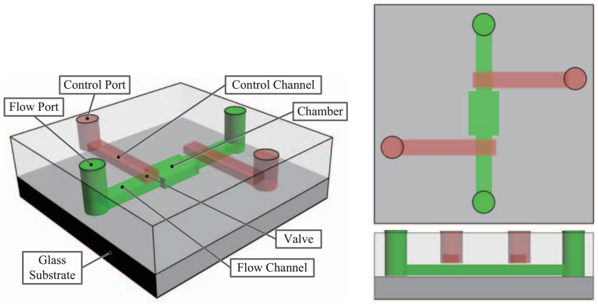
Qin Wang[1†], Shiliang Zuo[2†], Hailong Yao[1], Tsung-Yi Ho[3,4], Bing Li[5]
Ulf Schlichtmann[5], and Yici Cai[1]
1. Department of Computer Science & Technology, Tsinghua University
2. The Institute for Theoretical Computer Science (ITCS), Institute for Interdisciplinary Information Sciences, Tsinghua University
3. Institute for Advanced Study, Technical University of Munich (TUM) 4. National Tsing Hua University
5. Institute for Electronic Design Automation, Technical University of Munich (TUM)
woodythu@163.com, zsl14@mails.tsinghua.edu.cn, hailongyao@tsinghua.edu.cn
tyho@cs.nthu.edu.tw, b.li@tum.de, ulf.schlichtmann@tum.de, caiyc@tsinghua.edu.cn

ABSTRACT

Flow-based microfluidic biochips have progressed significantly in the past decade. Thanks to innovations in multilayer soft lithography (MSL) fabrication technology, the integration of thousands of microvalves along with large-scale networks of microchannels on a chip has been enabled. This progress has even been compared to the evolution of VLSI circuits following Moore's Law. In flow-based microfluidic biochips, microvalves are critical components to control the fluidic transportation for complex operations. To activate the open/close states of a microvalve, off-chip control pins are required. Due to the tremendous increase of the number of microvalves, a software-programmable microfluidic platform has been proposed to reduce the number of off-chip control pins, which integrates a microfluidic multiplexer on a separate control layer to control the array of microvalves. The multiplexer needs to be switched when the states of microvalves are changed between every two adjacent time slots. High switching frequency will make the multiplexer vulnerable and decrease the chip's reliability. We observe that different switching orders of microvalves lead to different switching frequencies of a multiplexer. Based on this observation, this paper proposes the first Hamming-distance-based switching order optimization method for microvalves to enhance the reliability of the multiplexer. Experimental results show that our method can significantly reduce the switching frequency of multiplexer, and the solution is very close to the theoretical optimal lower bound.

1. INTRODUCTION

Nowadays, with the aid of flow-based microfluidic biochips, automated laboratory procedures have emerged to revolutionize the biology and biochemistry experimental process [1–3]. The major advantages of microfluidic biochips over traditional laboratory analysis platforms include (1) greatly saving the assay cost by reducing expensive samples/reagents to nano-liter or pico-liter volume, (2) seamlessly integrating the automatic control logic for reduced human intervention and labor cost, (3) significantly increasing the sensitivity and accuracy of the biochemical assay, (4) potentially facilitating the portability characteristics for point-of-care diagnostic devices, and (5) naturally enabling the ultra-small chamber for single-cell culture and analysis requirements. As an example, microfluidic biochips have been developed for DNA synthesizing [4], cancer detection and HIV/AIDS diagnosis [5, 6].



(a) 3D view. (b) Top and side views.

Figure 1: Schematic of flow-based biochip [7].

Figure 1 from [7] shows the schematic of a flow-based microfluidic biochip based on the *multilayer soft lithography* technology, where the microfluidic functional units are fabricated by elastomer material (polydimethylsiloxane, PDMS) [8]. Figure 1(a) presents the 3D view of a demonstrative chamber for cell culture, and Figure 1(b) shows the corresponding top and side views of the same chamber. A typical biochip includes two PDMS layers on top of the glass substrate: (1) the *flow layer* for fluid transportation, where the flow channels are fabricated, and (2) the *control layer* with control channels for implementing the controlling logic. Both control and flow channels are connected to input/output ports, which are punched holes through the PDMS material. A *microvalve* (referred to as *valve* hereafter) is located between control and flow channels at their intersection region. To switch the valve, external *pressure source* is connected through the *control pin* (also called a *control port*). Then hydraulic or pneumatic pressure (∼10 psi) is conducted through the control channel to the valve, squeezing the elastomer down into the flow channel to stop the flow. When the external pressure is released, the elastomer of the valve restores back, and the fluid transportation resumes due to the external pressure from the *flow port*. By controlling the actuation sequences on valves in a programmed way, fluidic operations are executed automatically [9].

Valve is a critical functional unit in flow-based biochips, which is a basic building block to form complex fluid-handling functional units (also called *components*), such as mixers, switches, and multiplexers [9, 10]. Figure 2 shows a biochip example with a

[†] Indicates equal student contribution. This work was supported in part by the National Basic Research Program of China Grant 2011CBA00300, 2011CBA00301, the National Natural Science Foundation of China Grant 61361136003

978-1-5090-1558-0/17/$31.00 ©2017 IEEE

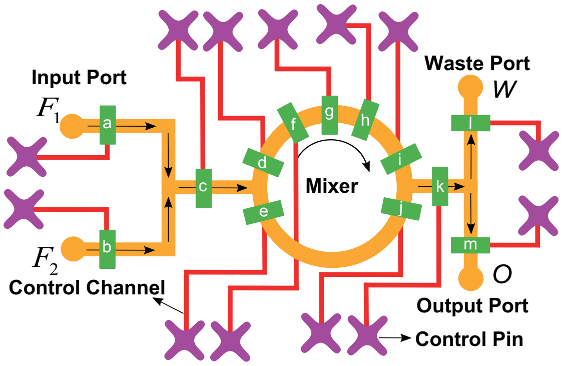


Figure 2: Microvalves for different microfluidic functional operations.

mixer, where valves are used for different functional operations [3, 10, 11]. In this figure, samples/reagents are dispensed from the two input ports F_1 and F_2. For mixing two fluids, the following major steps are needed: (1) Fluidic input from F_1 to the lower half mixer; (2) Fluidic input from F_2 to the upper half mixer; (3) Seal the mixer and start the mixing operation. (4) Transport mixed fluids in lower half mixer into output port; (5) Transport mixed fluids in upper half mixer into waste port.

1. Fluidic input from F_1 to the lower half mixer.
   - Open valves: a, c, e, j.
   - Close valves: b, d, i, k.
   - Don't-care valves: f, g, h, n, l, m.
2. Fluidic input from F_2 to the upper half mixer.
   - Open valves: b, c, d, f, g, h, i.
   - Close valves: a, e, j, k.
   - Don't-care valves: l, m.
3. Seal the mixer and start the mixing operation.
   - Open valves: d, e, i, j.
   - Close valves: c, k.
   - Don't-care valves: a, b, l, m.
   - Switching valves: f, g, h.
4. Transport mixed fluids in lower half mixer into output port.
   - Open valves: b, c, e, j, k, m.
   - Close valves: a, d, i, l.
   - Don't-care valves: f, g, h.
5. Transport mixed fluids in upper half mixer into waste port.
   - Open valves: a, c, d, f, g, h, i, k, l.
   - Close valves: b, e, j, m.

In Figure 2, the mixer consists of a *micropump*, which is a combination of three valves f, g, and h. By actuating the three valves in a peristaltic sequence at a high frequency (∼100 Hz), the fluids are forced to circulate around and mix together [11]. In flow-based microfluidic biochips, air in flow channels can be pushed through a closed valve, but fluids cannot bypass a closed valve. For example, when loading the fluidic input from F_1 to the lower half part of the mixer, valve k can be closed with correct functionality. A typical flow-based biochip may integrate hundreds or even thousands of valves, which requires automated design methods to achieve correct functionality and reduced turnaround time.

Valves are controlled by control pins and they are connected to each other through control channels. The mapping method from valves to control pins is called addressing. The mapping method is typically direct addressing, where one valve maps to one control pin. However, this method will lead to dramatically increased off-chip control pins. As shown in Figure 2, a mixer needs 13 control pins to finish the mix operation. Thus, hundreds of mixers on a chip can easily make the number of off-chip control pins unaffordable and cause design failure due to the increased control layer complexity. To solve this problem, an additional layer on the top of control layer is introduced to control valves in a software-programmable way [12]. In [13], some extra components are added intending for designing very large scale integration (VLSI) microfluidics in a user-reconfigurable way. [14] has proposed a software-programmable microfluidic platform using time division multiplexer to reduce the number of control pins as shown in Figure 3. The main idea is to use multiplexer to control the valves, which significantly reduces the number of necessary off-chip control pins. As shown in Figure 3, (a) presents the photograph of the software-programmable microfluidic device (PMD) outlining its constitutive modules. A control-channel network (blue) sits at the center of a core valve array (red) controlled by a 7-bit microfludic multiplexer (green) and a master core input. The micrograph of the central control-channel network is presented in (b). Individually addressable valves surround each network node providing full control over their connectivity. Routing channels access valves through microfluidic vias from a separate layer above. The micrograph of a network node is given in (c). Each node constitutes a 300 pl vessel and consists of several core valves. In this architecture, multiplexer is a major component, which determines the chip's functionality and reliability.

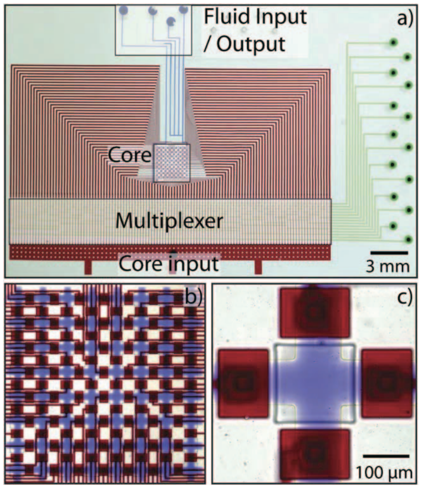
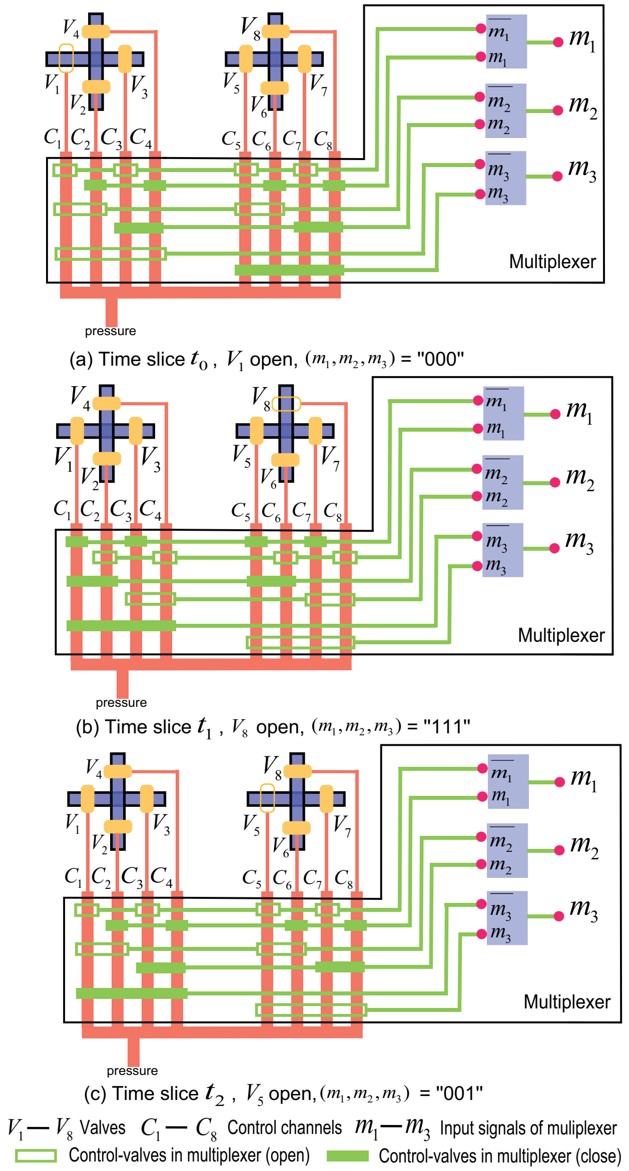


Figure 3: Software-programmable microfluidic platform [14].

The concept of actuation sequence in flow-based microfluidic biochips is very similar with the definition of actuation sequence in DMFB [16]. A little different is that the actuation sequences control the open/close of valves in flow-based microfluidic biochips, but they control the high/low voltage of electrodes in DMFB. Given the actuation sequences of valves at current time step and next time step, we need a switching order to make the different states of valves switched in a given order. Furthermore, different switching orders result in different switching frequencies of multiplexer. A high switching frequency makes the multiplexer vulnerable and reduces the chip's reliability. Thus, this paper proposes a Hamming-distance-based switching order optimization method to decrease the switching frequency of multiplexer in flow-based microfluidic biochips. Major contributions of this paper are summarized as follows.

- We observe for the first time the switching order optimization problem, which is critical for enhancing the reliability of multiplexer in software-programmable microfluidic platform.
- The first switching order optimization method is proposed for multiplexer in flow-based microfluidic biochips.
- The total switching frequency of multiplexer is greatly reduced, which significantly improves the reliability of multiplexer and enhances the lifetime of chips.



(a) Time slice t_0, V_1 open, (m_1,m_2,m_3) = "000"

(b) Time slice t_1, V_8 open, (m_1,m_2,m_3) = "111"

(c) Time slice t_2, V_5 open, (m_1,m_2,m_3) = "001"

Figure 4: Switching of multiplexer.

- The proposed Hamming-distance-based method obtains the solution very close to the optimal lower bound.

The remainder of the paper is organized as follows. Section 2 presents a background on time division multiplexer and the problem formulation. An overview of the proposed method is given in Section 3. The proposed Hamming-distance-based algorithm is described in Section 4. Section 5 presents and discusses the experimental results. Finally, conclusion is drawn in Section 6.

2. PROBLEM FORMULATION

As shown in Figure 4, valves are used to control fluidic transportation, and control-valves are used to open/close a valve. We use "0" and "1" to denote the close and open states of a valve. Time division is the major idea in [14], which indicates that time units for valves and control valves are different. Here we give the definitions of the two time units as follows:

- *Time slice:* The time unit for control-valve switching in multiplexer.
- *Time slot:* The time unit for valve switching in control layer. A time slot includes many time slices.

The multiplexer can use three bits m_1, m_2, and m_3 to activate control-valves in Figure 4. As shown in Figure 4(a), by assigning m_1 to m_3 "000", valve v_1 is opened and other valves are closed. Assume that the actuation sequence of valves from v_1 to v_8 at current time step is "10000000", and the actuation sequence at next time step is "00001001". v_5 and v_8 need to change state from close to open, and v_1 needs to change state from open to close (see Figure 4(b) and 4(c)). Please note that each actuation sequence of multiplexer (m_1, m_2, m_3) can only open one valve at a time, and the other valves are all closed. Thus, we only need to switch valves from close to open. If the switching order of valves is <v_8, v_5>, the actuation sequence of multiplexer from m_1 to m_3 need to be changed as "000", "111", "001". The switching times of control-valves in the multiplexer is 3 from "000" to "111", and 2 from "111" to "001". Thus the total switching times of the control-valves in the multiplexer is 5. However, if the switching order of valves is <v_5, v_8>, the actuation sequence of multiplexer needs to be changed as "000", "001", "111". The switching times of control-valves in the multiplexer is 1 from "000" to "001" and 2 from "001" to "111". In this way, the switching times of the control-valves in the multiplexer can be decreased from 5 to 3.

Figure 5 illustrates the two different solutions mentioned above for switching the multiplexer. During the process we only need to focus on the valves from close state to open state, because each actuation sequence of multiplexer can open only one valve in a time slice. Although there are two valves need to be changed in a time slot, time division strategy can be applied to switch the states of valves in an specific order. According to the definitions above, the control valves of the multiplexer can switch one time in a time slice. Please note that a time slot can include many time slices. Thus, the switch of valves could be finished in two time slices in Figure 5.

In summary, this paper presents a switching order optimization method addressing the following problem:

Given: The number of valves n, and the actuation sequences of valves $C^t = \{C_1^t, C_2^t, C_3^t, ..., C_n^t,\}$ from the beginning time step T_{begin} to the end time step $T_{end} (t \in [T_{begin}, T_{end}])$.

Find: Switching order $S = \{M_{T_{begin}}, ..., M_{T_i}, M_{T_{i+1}}, ...M_{T_{end}}\}$ of multiplexer from T_{begin} to T_{end}, where each time slot (T_i, T_{i+1}) includes many time slices.

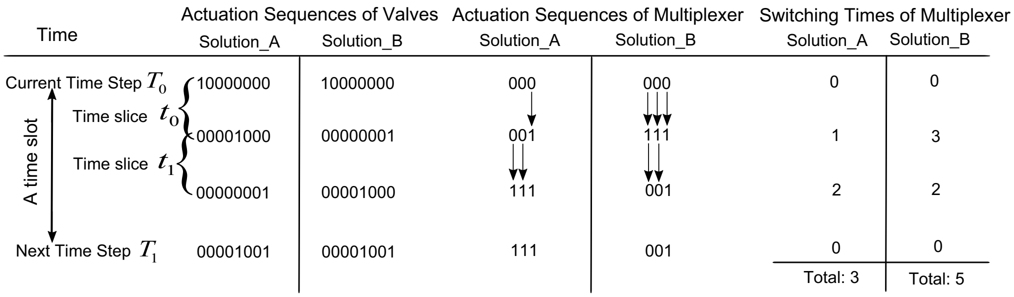
Objective: Minimize the cost of total switching times of the control-valves in the multiplexer.

Subject to: All of the different control signals in C^t from current time step t to next time step $t+1$ must be switched.

3. OVERALL DESIGN FLOW

Figure 6 presents the design flow of our method. The input of the flow includes actuation sequences of valves from the beginning of an assay to the end of an assay. Then, we compare the two adjacent actuation sequences from current time step T_i to next time step T_{i+1}, and find the different control signals to be switched. Next, our method determines the switching order of valves from T_i to T_{i+1} using Hamming-distance-based strategy. The next step is to calculate and record the switching frequency of multiplexer in current time slot (T_i, T_i). Then, the method moves to next time step and deals with the following two adjacent actuation sequences in the same way. The flow completes when the current time step reaches to the end of the assay (T_{end}). Finally, the switching order and switching frequency of control valves in the multiplexer are reported.

Although the number of valves on a chip may be hundreds, we do not need to activate all of them in a time slot. It is determined by the assay and the synthesis and scheduling results. Actually, only a part of valves need to be controlled in each time slot (T_i, T_{i+1}). Thus, the states of other valves can be open or close, and this kind of state is called don't-care state. For actuation sequences of valves, we use "1" to denote open state, "0" for close state and "X" for don't-care state.



Time	Actuation Sequences of Valves		Actuation Sequences of Multiplexer		Switching Times of Multiplexer	
	Solution_A	Solution_B	Solution_A	Solution_B	Solution_A	Solution_B
Current Time Step T_0	10000000	10000000	000	000	0	0
Time slice t_0	00001000	00000001	001	111	1	3
Time slice t_1	00000001	00001000	111	001	2	2
Next Time Step T_1	00001001	00001001	111	001	0	0
					Total: 3	Total: 5

Figure 5: Different solutions for switching of a multiplexer.

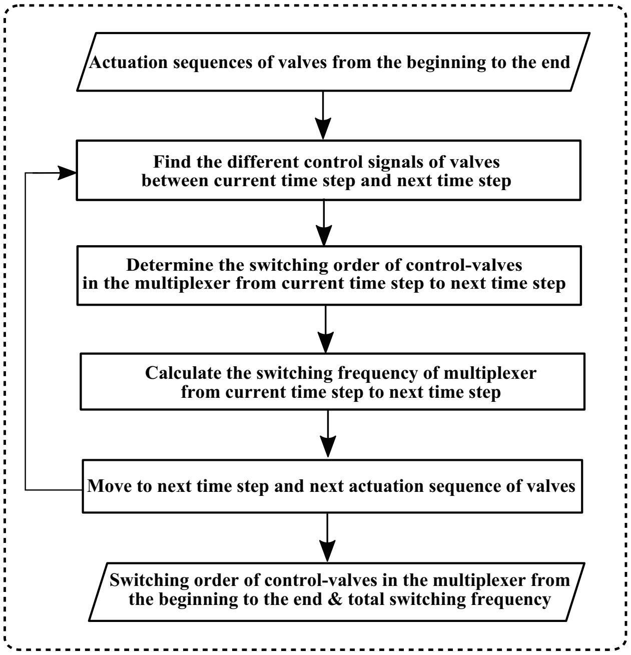

Actuation sequences of valves from the beginning to the end
Find the different control signals of valves between current time step and next time step
Determine the switching order of control-valves in the multiplexer from current time step to next time step
Calculate the switching frequency of multiplexer from current time step to next time step
Move to next time step and next actuation sequence of valves
Switching order of control-valves in the multiplexer from the beginning to the end & total switching frequency

Figure 6: Overall design flow of our method.

```
Input: The number of valves $n$, and actuation sequences of valves from the
       beginning to the end of an assay.
Output: Switching order of multiplexer from the beginning to the end, and total
        switching frequency.
Let $C^t = \{C_1^t, C_2^t, C_3^t, ..., C_n^t,\}$ be the actuation sequence of valves at time slot $t$;
Let $T_{begin}$ be the beginning time of an assay and $T_{end}$ be the end time of an assay,
  and $t \in [T_{begin}, T_{end}]$ ;
Let $S$=$\{M_{T_{begin}}, M_{t_i}, M_{t_{i+1}}, ..., M_{T_{end}}\}$ be the switching order of multiplexer from
  $T_{begin}$ to $T_{end}$;
Let $Res$ be the total switching times from $T_{begin}$ to $T_{end}$;
Let $Frequency$ be the total switching frequency from $T_{begin}$ to $T_{end}$;
for $t = T_{begin}$ to $T_{end}$ do
    for $i = 1$ to $n$ do
        $c = C_i^t$;
        if $c ==$ "X" then
            if $C_i^{t-1} == 0$ and $C_i^{t+1} == 0$ then
                $c = 0$;
            else if $C_i^{t-1} == 1$ and $C_i^{t+1} == 1$ then
                $c = 1$;
            else
                $c = C_i^{t-1}$;
    Find the different states of valves between $C^t$ and $C^{t+1}$;
    Set $Current$ to be the first different index of changed valve;
    while switching is not finished do
        Find the index of changed valve with nearest Hamming-distance with
          $Current$;
        Set $Next$ to be the above index;
        Set $Tmp$ to be the number of valves to change state from $t$ to $t+1$;
        Switch the multiplexer according to the change from $C_{Current}^t$ to $C_{Next}^t$;
        Set $M_t$ to be the above switching solution of multiplexer;
        $Current = Next$;
        $Res = Res + Tmp$;
    $t = t + 1$;
$Frequency = \frac{Res}{T_{end} - T_{begin}}$;
```

Algorithm 1: Hamming-distance-based switching order optimization algorithm.

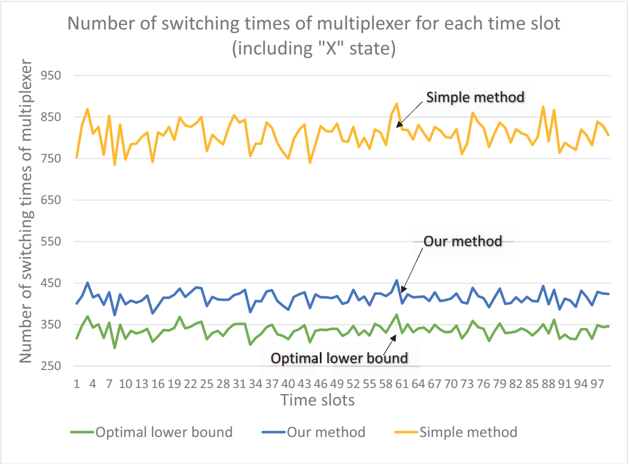


Figure 7: Number of switching times among optimal lower bound, our method, and simple method for b1.

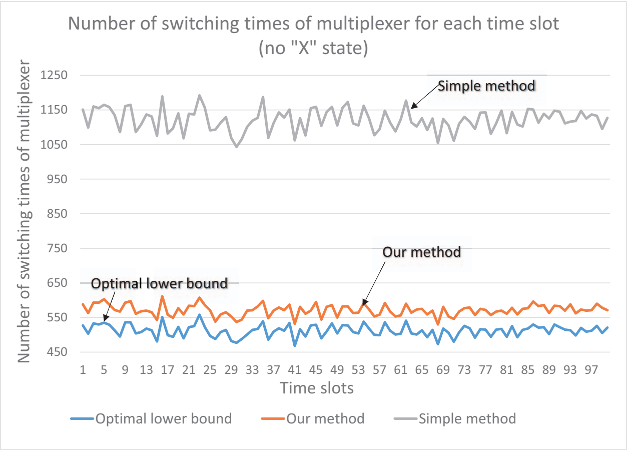


Figure 8: Number of switching times among optimal lower bound, our method, and simple method for c1.

4. HAMMING-DISTANCE-BASED VALVE-SWITCHING STRATEGY

According to the definition in [15], the Hamming distance between two strings of equal length is the number of positions at which the corresponding symbols are different. In another way, it measures the minimum number of substitutions required to change one string into the other, or the minimum number of errors needed to transform one string into the other. For example, the Hamming distance between "001001" and "101101" is 2. Algorithm 1 gives the details of our algorithm.

In Algorithm 1, we first deal with the "X" (don't-care) states in actuation sequences of valves from input. For each time step t, the method compares the states of valves between time step $t-1$ and $t+1$. The "X" state at time step t is assigned according to the comparison results to reduce the switching times of valves. Then, for each time loop, the different states of valves between C^t and C^{t+1} are found afterwards and recorded. Next, the switching order of changed valves are determined according to their Hamming distances. The time step moves forward one step until the assay is finished.

Now we analyze the time complexity of Algorithm 1. The time loop executes for $(T_{end} - T_{begin})$ times. For each time loop, the loop for actuation sequences preprocessing and switching operations both take maximum n times. The computation of minimum Hamming-distance takes $O(n)$ time, where n is the number of valves. Therefore, the overall time complexity of our algorithm is $O((T_{end} - T_{begin}) \times n^2)$.

5. EXPERIMENTAL RESULTS

We have implemented the proposed Hamming-distance-based method for flow-based microfluidic biochips in C++ and tested it on a 2.60GHz 32-core Intel Xeon Linux workstation with 132GB memory. Only a single thread is used. We have synthesized 60 benchmarks to evaluate the performance of our method. For benchmarks b1 (c1) to b10 (c10), the number of valves is 1024 and the total number of time slots is 100. For benchmarks b11 (c11) to b20 (c20), the number of valves is 2048 and the total number of time slots is 100. For benchmarks b21 (c21) to b30 (c30), the number of valves is 2048 and the total number of time slots is 200. Furthermore, benchmarks b1 to b30 include "X" (don't-care) states, and benchmarks c1 to c30 do not include "X" states. To validate the effectiveness of the proposed method, we have implemented a simple baseline method, where the decision of switching order is based on the order of valve's relative position. Here we use total switching times to denote switching frequency because the total time slots are the same. For valves in control layer, each change of states results in at least one switching time of control-valves in the multiplexer. Thus, the optimal lower bound is the total number of changed states of valves from the beginning time step to the end time step.

Figure 7 to 10 present the comparison results among optimal lower bound, our method, and the simple method above. Experimental results show that the solution of our method is very close to the optimal lower bound, and there is a significant improvement compared with the simple method. Figure 7 and 8 show the number of switching times of control valves in each time slot among optimal lower bound, our method, and simple method for benchmarks b1 and c1. The figures show that our method can obtain the close-to-optimal solution for each time slot. Figure 11 gives the runtime of the two methods, and it shows that the overhead is acceptable.

6. CONCLUSION

Based on the software-programmable microfluidic platform proposed by [14], the number of off-chip control pins in flow-based microfluidic biochips can be reduced dramatically. In this platform, time division multiplexer is the key component, which determines the functionality and reliability of a chip. However, the high switching frequency of multiplexer makes it vulnerable. To solve this problem, decreasing the switching times of control-valves in the multiplexer should be a major concern in the design flow. In this work, we present a switching order optimization method for control-valves in the multiplexer of flow-based microfluidic biochips. In this method, the Hamming-distance-based strategy is applied and the switching times of control-valves is optimized to enhance the reliability of the multiplexer. Experimental evaluations show that our method is effective and efficient.

7. ACKNOWLEDGEMENTS

The work of H. Yao was supported by Tsinghua University Initiative Scientific Research Program (20141081203) and the National Natural Science Foundation of China (61674093). The work of T. Ho was supported in part by the Ministry of Science and Technology of Taiwan, under Grant MOST 105-2221-E-007-118-MY3 and 104-2220-E-007-021 and in part by the Technical University of Munich-Institute for Advanced Study, funded by the German Excellence Initiative and the European Union Seventh Framework Program under grant agreement no 291763. The work of B. Li and U. Schlichtmann was supported by the IGSSE Project FLUIDA of Technical University of Munich (TUM). The work of Y. Cai was supported by the National Natural Science Foundation of China (61274031).

8. REFERENCES

[1] T. Thorsen, S. J. Maerkl, and S. R. Quake, “Microfluidic Large-Scale Integration,” *Science*, vol. 298, no. 5593, pp. 580-584, 2002.

[2] G. M. Whitesides, “The Origins and the Future of Microfluidics,” *Nature*, vol. 442, no. 7101, pp. 368-373, 2006.

[3] D. Mark, S. Haeberle, G. Roth, F. von Stetten, and R. Zengerle, “Microfluidic Lab-on-a-Chip Platforms: Requirements, Characteristics and Applications,” *Chemical Society Reviews*, vol. 39, no. 3, pp. 1153-1182, 2010.

[4] Y.Huang, P.Castrataor, C.-C. Lee, and S. R. Quake, “Solvent resistant microfluidic DNA synthesizer,” *Lab on a Chip*, vol.7, no. 1, pp. 24-26, 2006.

[5] N. N. Watkins, U. Hassan, G. Damhorst, H. Ni, A. Vaid, W. Rodriguez, and R. Bashir, “Microfluidic CD4+ and CD8+ T lymphocyte counters for point-of-care HIV diagnostics using whole blood,” *Science Translational Medicine*, vol. 5, no. 214, pp. 214ra170, 2013.

[6] Rice University. “Quick nano-bio-chip checks for oral cancer,” *ScienceDaily*, www.sciencedaily.com/releases/2010 /04/100405152753.htm.

[7] Qin Wang, Yizhong Ru, Hailong Yao, Tsung-Yi Ho, Yici Cai, “Sequence-pair-based placement and routing for flow-based microfluidic biochips” *Proc. of ASPDAC*, pp. 587-592, 2016.

[8] J. A. Rogers and R. G. Nuzzo, “Recent Progress in Soft Lithography,” *Materials Today*, vol. 8, no. 2, pp. 50-56, 2005.

[9] T. M. Squires and S. R. Quake, “Microfluidics: Fluid Physics at the Nanoliter Scale,” *Reviews of Modern Physics*, vol. 77, pp. 977-1026, 2005.

[10] V. Studer, G. Hang, A. Pandolfi, M. Ortiz, W. F. Anderson, and S. R. Quake, “Scaling Properties of a Low-Actuation Pressure Microfluidic Valve,” *Journal of Applied Physics*, vol. 95, no. 1, pp. 393-398, 2004.

[11] K. Hu, T. A. Dinh, T.-Y. Ho, and K. Chakrabarty, “Control-Layer Optimization for Flow-Based mVLSI Microfluidic Biochips,” *Proc. of CASES*, pp. 16:1-16:10, 2014

[12] E. C. Jensen, B. P. Bhat and R. A. Mathies, “A digital microfluidic platform for the automation of quantitative biomolecular assays,” *Lab on a Chip*, vol. 10, no. 10, pp. 685-691, 2009.

[13] M.-P. Chang and M. M. Maharbiz, “Electrostatically-driven elastomer components for user-reconfigurable high density microfluidics” *Lab on a Chip*, vol. 9, no. 9, pp. 1274-1281, 2009.

[14] L. M. Fidalgo, S. J. Maerkl, “A software-programmable microfluidic device for automated biology,” *Lab on a Chip*, vol. 11, no. 9, pp. 1612-1619, 2011.

[15] R. W. Hamming, “Error detecting and error correcting codes,” *Proc. of Bell System Technical Journal*, vol. 29, no. 2, pp. 147-160, 1950.

[16] Qin Wang, Weiran He, Hailong Yao, Tsung-Yi Ho, Yici Cai, “SVM-Based Routability-Driven Chip-Level Design for Voltage-Aware Pin-Constrained EWOD Chips” *Proc. of ISPD*, pp. 49-56, 2015.

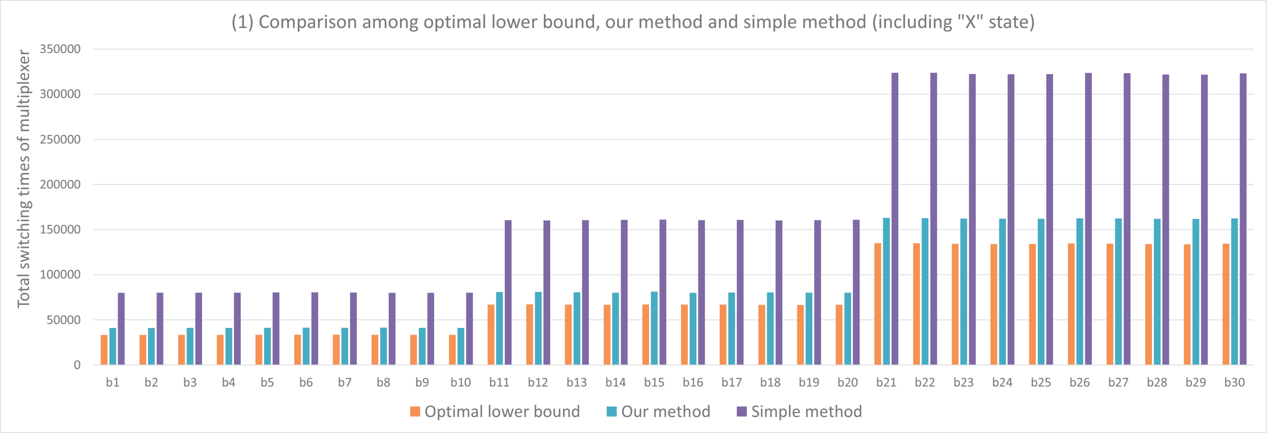


Figure 9: Comparison among optimal lower bound, our method, and simple method for b1 to b30.

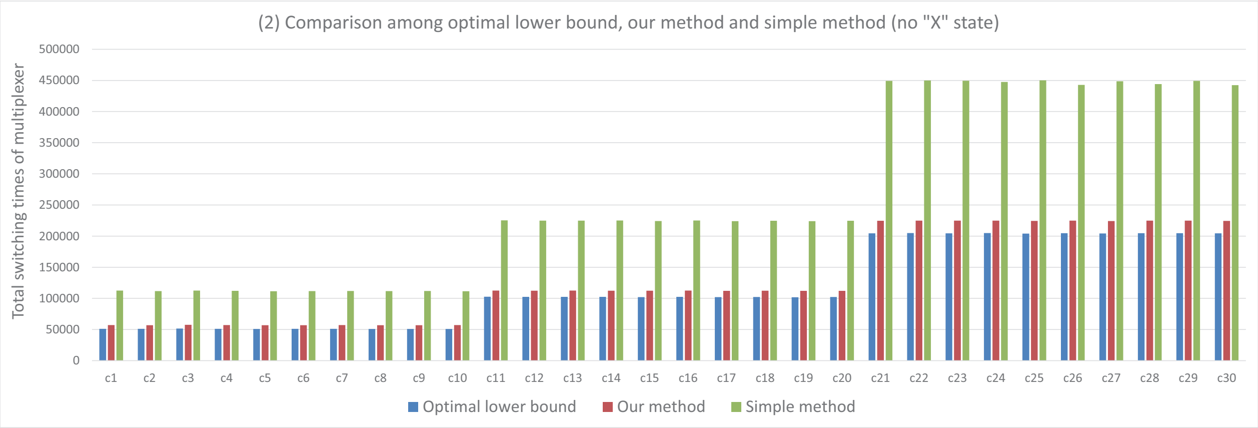


Figure 10: Comparison among optimal lower bound, our method, and simple method for c1 to c30.

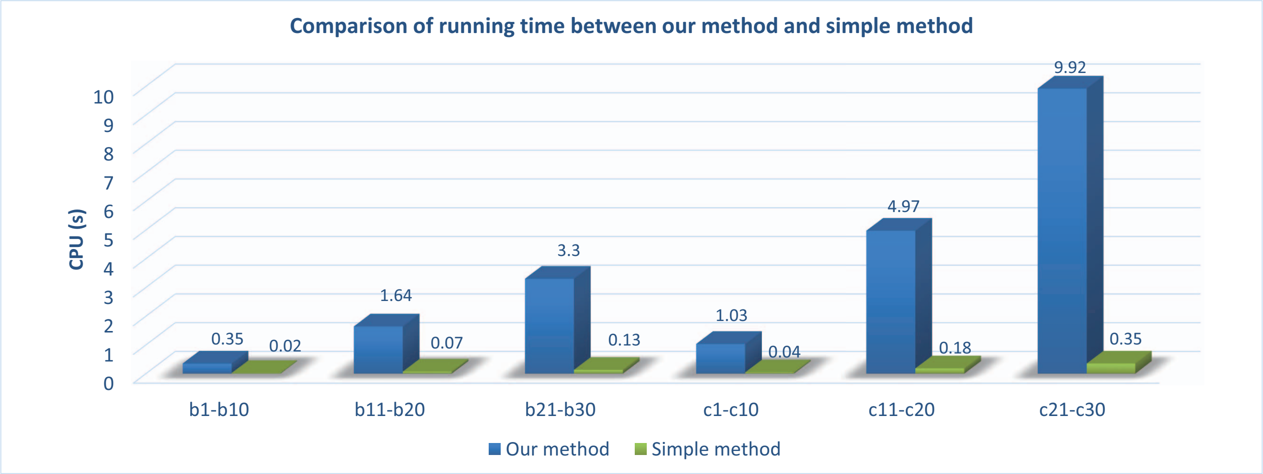


Figure 11: Comparison of running time between our method and simple method for all benchmarks.