Transport-Free Module Binding for Sample Preparation using Microfluidic Fully Programmable Valve Arrays*

Gautam Choudhary[1,2,†], Sandeep Pal[2,†], Debraj Kundu[2], Sukanta Bhattacharjee[3], Shigeru Yamashita[4], Bing Li[5], Ulf Schlichtmann[5] and Sudip Roy[2]

[1]Adobe Research, India, [2]CoDA Laboratory, IIT Roorkee, India, [3]Dept. of CSE, IIT Guwahati, India, [4]CISE, Ritsumeikan University, Japan, and [5]Chair of EDA, Technical University of Munich, Germany

***Abstract*—Microfluidic fully programmable valve array (FPVA) biochips have emerged as general-purpose flow-based microfluidic lab-on-chips (LoCs). An FPVA supports highly re-configurable on-chip components (modules) in the two-dimensional grid-like structure controlled by some software programs, unlike application-specific flow-based LoCs. Fluids can be loaded into or washed from a cell with the help of flows from the inlet to outlet of an FPVA, whereas cell-to-cell transportation of discrete fluid segment(s) is not precisely possible. The simplest mixing module to realize on an FPVA-based LoC is a four-way mixer consisting of a 2×2 array of cells working as a ring-like mixer having four valves. In this paper, we propose a design automation method for sample preparation that finds suitable placements of mixing operations of a mixing tree using four-way mixers without requiring any transportation of fluid(s) between modules. We also propose a heuristic that modifies the mixing tree to reduce the sample preparation time. We have performed an extensive simulation and examined several parameters to determine the performance of the proposed solution.**

I. Introduction

Microfluidic biochips or Lab-on-Chips (LoCs) are highly integrated analogue of the traditional biochemical laboratory that can perform thousands of experiments using nano-/pico-liter volume of reagents. LoCs redefine the way biological experiments are performed by offering a low-cost automated platform for point-of-care diagnosis [1], sample preparation [2], genomics [3], and drug discovery [4], [5]. Among several LoCs, flow-based microfluidic biochips (FMBs) manipulate fluids in a network of micro-channels by actuating pressure driven micro-valves [6].

A fully programmable valve array (FPVA) offers a general purpose FMB. An FPVA biochip (Fig. 1) is a two-dimensional array of fluid chambers which is surrounded by up to four microvalves [7]. Valves can be configured to create arbitrary shaped flow paths, mixers, and fluid storage in a programmable fashion (Figs. 1(a)-(b)). The FPVA also supports on-chip components such as fluid metering unit, storage cell, and mixer [7] (Fig. 1(c) and 1(d)).

Any biochemical assay can be modelled as a sequence graph and the process of realizing it on any microfluidic platform is termed as the synthesis of that bioassay. An important step of any bioassay synthesis is module binding, where nodes of the sequence graph are bound with the instances of on-chip components. Sample preparation is a biochemical assay, where two or more fluids are required to be mixed in a desired ratio [8]. LoC-based sample preparation models the bioassay as a mixing tree to achieve the desired target ratio. Irrespective of any microfluidic platform, storage cells and mixers are required for realizing any mixing tree.

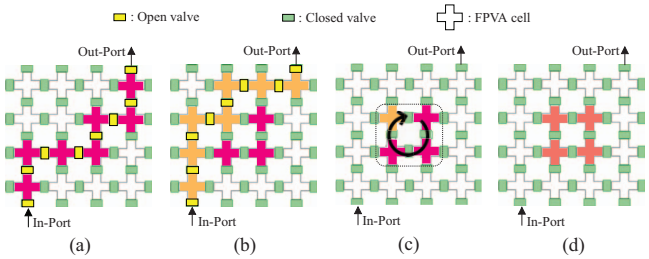

Fig. 1: Paths from in-port to out-port is created by changing valve states before loading of (a) fluid A (pink) and (b) fluid B (orange). (c) Mixing operation of fluids A and B in (3:1) ratio, respectively (one (three) cell(s) is (are) filled with fluid B(A)). (d) storage operation.

FPVA biochips can perform multiple active mixing in a re-configurable fashion [7]. Several design automation solutions for testing [9] and washing [10] have been proposed by leveraging the above mentioned capabilities of the FPVAs. Although an FPVA supports re-configurable mixer creation and storage of fluids, transportation of precise amounts of fluids from one cell to another cell is challenging (see sec II-B). Note that, with the existing synthesis algorithms for implementing a bioassay on FPVA it is difficult as they require transportation of fluids between cells [11], [12].

In this paper, we propose a practical design automation solution for sample preparation on the FPVA biochips that does not require any transportation of fluids between cells. To the best of our knowledge, the proposed solution considers the constraint of transportation between cells in the FPVA for the first time during sample preparation. We also present a heuristic algorithm to reduce the bioassay completion time. The main contributions of the proposed work are summarized as follows: (1) Transport-free module binding of a mixing tree on an FPVA; and (2) Modifying the initial mixing tree by a heuristic to reduce overall bioassay completion time.

II. Background

A. Sample Preparation on Microfluidic Biochips

Sample preparation is a process of mixing two or more input reagents in a desired ratio. A sequence of mixing steps,

[†]Equal contribution

*This work by D. Kundu and S. Roy was supported partially by the early career research award (grant #: ECR/2016/001921) sponsored by SERB, Govt. of India. This work by B. Li, U. Schlichtmann and S. Roy was supported partially by the Indo-German joint research project (grant #: INT/FRG/DAAD/P-21/2018) sponsored by DST, India and DAAD, Germany.

which is commonly represented using a mixing tree [13], is used to denote sample preparation on microfluidic biochips. An internal (leaf) node in the mixing tree represents a mix operation (input reagent) and an edge weight represents the volume of fluid shared between the nodes. In the literature, several sample preparation algorithms exist for different microfluidic platforms such as digital microfluidic biochips [8], [14] and FMBs [15]. FMB-based sample preparation platforms are generally equipped with one or more on-chip rotary mixers and storage-cells [13]. A storage-cell is required to store intermediate fluids for subsequent use.

B. Fluid Transport Constraint for FPVA

In general, an FMB uses carrier fluids such as silicone oil for transporting the payload fluid inside a microchannel. This transportation of a precise volume of fluid requires sophisticated microstructure called microfluidic latch, a partially closed valve that allows the carrier fluid to pass while blocking the payload fluid [16]. The fluid segmentation within the microchannel of an FMB is a well-known problem while transporting a fluid segment using a carrier fluid [17]. It also poses a significant challenge for an FPVA. In order to move a fluid segment within FPVA from the source cell(s) to the target cell(s) (Fig. 2(a)), a career fluid is required to give the pressure through a flow channel from in-port to out-port connecting the source cell(s) to the target cell(s) (Fig. 2(b)). As the flow channel has several right-angle turns, it results into a non-uniform pressure. Hence, the fluid segment may have breakages as shown in Fig. 2(b). For fluid transportation, a microfluidic latch valve is used at the outlet of the target cell (Fig. 2(b)) that blocks the payload fluid while bypassing the carrier fluid. Thus, in this paper, we consider the transportation of a fluid segment is not possible within an FPVA.

C. Motivation of the Work

So far, existing sample preparation algorithms for custom FMB mandate transportation of precise volume of fluids. Unfortunately, the FPVA does not support precise transportation of fluids between cells. Therefore, existing design automation solutions for sample preparation are no longer usable for the FPVAs as they need to transport fluids between cells. However, each cell in the FPVA can store unit volume or takes part in a mixing operation, which can be realized as a rotary mixer by grouping multiple cells together in a re-configurable fashion. In this work, we consider the constraint of transportation between cells in the FPVA, for the first time, and propose

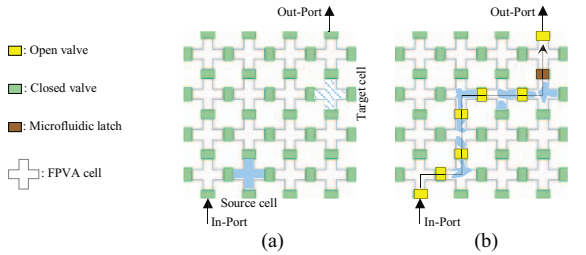

Fig. 2: Fluid transport constraint for an FPVA. (a) Requirement of fluid transportation from source to target cells, and (b) fluid segmentation while transporting a fluid using an immiscible fluid.

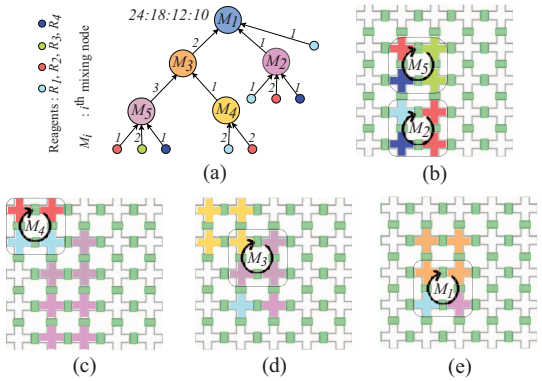

Fig. 3: (a) Mixing tree T for 24:18:12:10 obtained by *genMixing*. (b)-(e) Module binding and scheduling of T in four time cycles.

a design automation approach for sample preparation using FPVA. We leverage the advantage of re-configurable mixing in the FPVA to perform the sample preparation that does not require transportation of fluids between cells.

Example 1: Consider the mixing tree shown in Fig. 3(a) for mixing four input reagents (R_1, R_2, R_3, R_4) with a target ratio $R_1 : R_2 : R_3 : R_4 = 24 : 18 : 12 : 10$. The binding of scheduling of mix operations at four different timestamps are shown in Figs. 3(b)-(e). At $t = 1$, two mixing operations M_5 and M_2 are bound (Fig. 3(b)). Since M_3 mix fluids, which are produced by M_5 and M_4, we need to bind M_4 in a suitable location such that no transportation is required for the mix operation M_3. Fig. 3(c) shows the binding of M_4 at $t = 2$. Fig. 3(d) and 3(e) show the module binding of the mix operations M_3 and M_1 at $t = 3$ and $t = 4$, respectively.

III. Transport-Free Module Binding for FPVA

For a given mixing tree, the mix operations are scheduled and bound to 2×2 modules on the FPVA using an iterative algorithm called as *No Transport Mixing* (*NTM*), which does not require any transportation from the storage cells to the corresponding mixers. We propose another method called as *Heuristic for Distribution Algorithm (HDA)*, which transforms the mixing tree for reducing the overall completion time of sample preparation. In this work, we consider *genMixing* and *FloSPA-M* to determine the mixing trees as they can exploit all the mixing models for a 2×2 mixer [15].

A. Problem Formulation

The problem is formulated as follows: **Inputs**: a mixing tree T, and an FPVA; **Output**: scheduling and binding of nodes (mix operations) of T with on-chip 2×2 modules; **Constraint**: cell-to-cell transportation of a fluid after storage is not allowed. **Objectives**: to minimize (i) the completion time τ of sample preparation, (ii) the total number of cells used C, (iii) the chip-area (estimated as the total number of cells within the minimum bounding rectangle covering C), and (iv) the total number of valves used V, while executing T on the FPVA.

B. No Transport Mixing: NTM

The proposed module binding algorithm *NTM* schedules and binds the mixing nodes of a mixing tree T on the given FPVA. Without loss of generality, the xy-Cartesian coordinate system

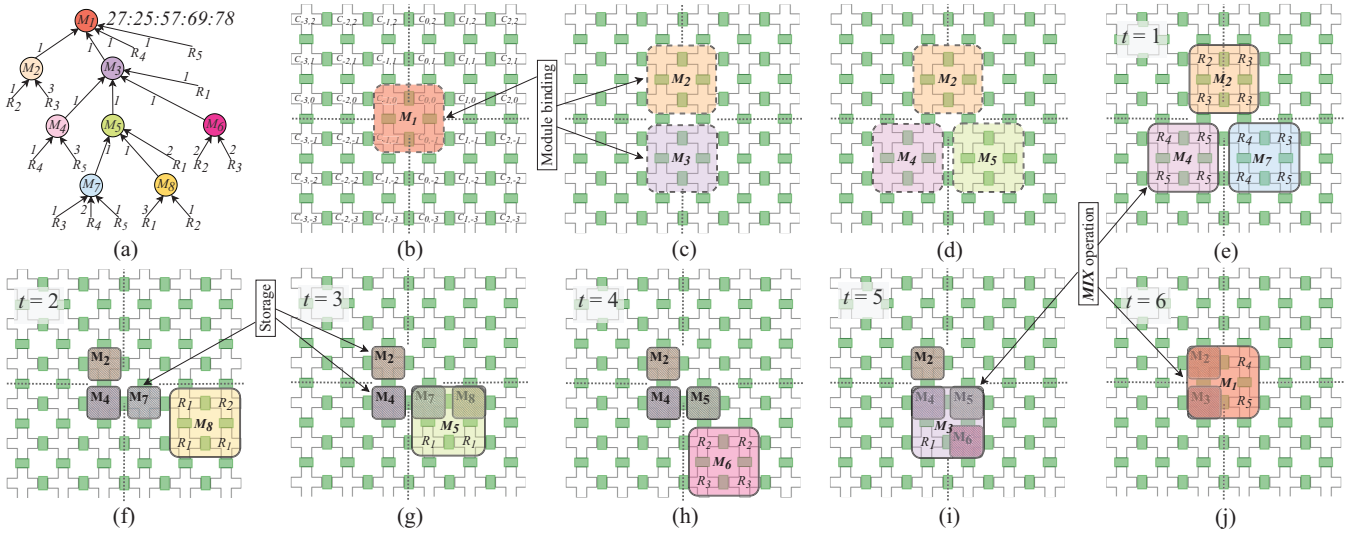

Fig. 4: For a target ratio $27 : 25 : 57 : 69 : 78$, (a) mixing tree T obtained by genMixing [15]. (b)-(j) Module binding and time-stamp assignment of the mixing nodes in T on a 6×6 FPVA using *NTM*.

is assumed on the given FPVA cells as shown in Fig. 4(b). For a no-transport mixing we divide the FPVA into quadrants and bind the modules in different quadrants around the origin FPVA cell $C_{0,0}$. A 2×2 mixers can realize (1:3), (1:1), (1:2:1), and (1:1:1:1) mixing models for on-chip sample preparation [15]. So, we consider only 2×2 mixing modules for the module binding. Note that, the four cells corresponding to a 2×2 mixing module $M_{x,y}$, is determined by the coordinates of the top-right cell of the module, i.e., (x, y).

The first step of *NTM* is to reorient internal nodes of T, which is denoted as *Left Factoring*. Given the mixing tree T, *Left Factoring* reorients the child nodes (mixing node only) of each parent node in decreasing order of their edge weights from left to right in the same level. After the *Left Factoring* of the mixing tree shown in Fig. 5(a), M_3 appears first and then M_2. Similarly, M_5 appearing first and then M_4 (Fig. 5(b)). This transformation facilitate the transport-free module binding without changing the target ratio.

In *NTM*, we bind the mixing nodes with modules using the following two criteria. **1. Module binding criteria (MBC)**: A mixing node is allowed to bound only in the quadrants in which its parent node was bound. While binding, the main objective is to bind maximum number of children of a parent node in disjoint quadrants. Children bound to disjoint quadrants can be executed in parallel. **2. Time-stamp assignment criteria (TAC)**: When all the inputs of different mixing nodes bound to disjoint quadrants are available, a time-stamp is assigned to the corresponding on-chip modules. All mixing modules with the same time-stamp are executed in parallel.

After left factoring, *NTM* traverse the mixing nodes of the tree in BFS order and places them to the FPVA according to MBC. Each time when the TAC is satisfied we assign the time-stamps to the concerned modules. We describe the module binding procedure using the following example.

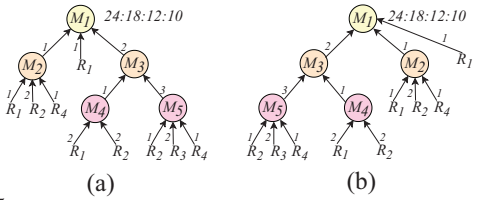

Fig. 5: (a) Before and (b) after applying *Left Factoring* on T.

Example 2: Consider the mixing tree shown in Fig. 4(a). Initially, M_1 is bound at $M_{0,0}$ (Fig. 4(b)), then its child nodes M_2 and M_3 are bound according to the MBC at $M_{0,1}$ and $M_{1,-1}$, respectively (Fig. 4(c)). Next, M_4 and M_5 (child nodes of M_3) are bound at $M_{-1,-1}$ and $M_{0,-1}$, respectively (Fig. 4(d)). Although M_6 is a child node of M_3 it cannot be bound along with M_4 and M_5. Only M_7 (child node of M_5) is bound at $M_{1,-1}$ (Fig. 4(e)). At this stage the TAC is satisfied, so time-stamp $t = 1$ is assigned to M_2, M_4 and M_7 (Fig. 4(e)). Then M_8 (child node of M_5) is bound at $M_{2,-1}$ and since TAC is satisfied time-stamp $t = 2$ is assigned to M_8 (Fig. 4(f)). Now, according to the MBC no other modules can be bound and as time-stamp $t = 3$ is assigned to M_5 (Fig. 4(g)). Eventually, M_6 is bound at $M_{1,-2}$ and time-stamp $t = 4$ is assigned to M_6 (Fig. 4(h)). Next, $t = 5$ is assigned to M_3 (Fig. 4(i)) and finally, $t = 6$ is assigned to M_1 (Fig. 4(j)).

C. Heuristic for Distribution Algorithm: HDA

In this section, we propose a heuristic method, called *HDA*, which transforms the mixing tree and minimizes the sample preparation time as compared to *NTM*. The idea is to transform the input mixing tree into another one preserving the output ratio such that it balances the number of mix operations bound in each quadrant and parallelize the process. The heuristic is based on the following transformations of a mixing tree [8]: (a) the nodes (mix operations or input reagents) appearing at the same level of the tree can be permuted, and (b) a node corresponding to an input reagent can be replaced by a mixing node with four copies of the same reagent as its child nodes.

When *NTM* is applied directly to the mixing tree shown in Fig. 6(a), the third quadrant is heavily loaded with mix operations while other quadrants are mostly idle (Gantt chart in Fig. 6(c)). This is due to the fact that the sub-tree rooted at

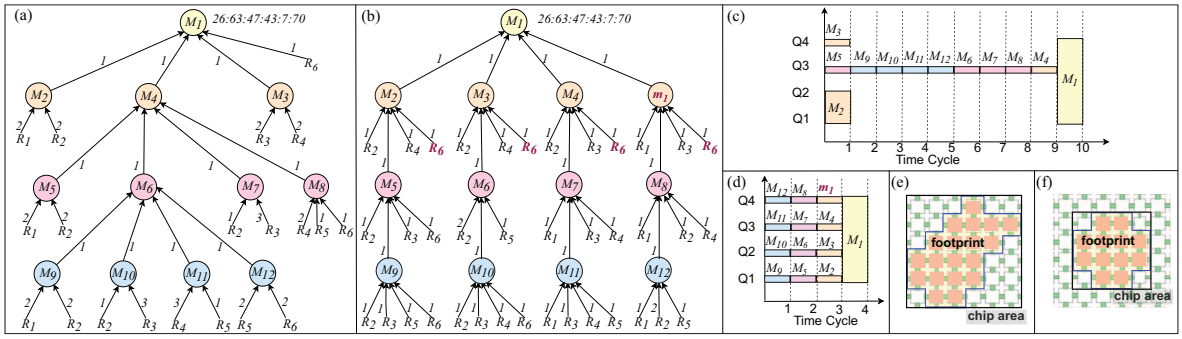

Fig. 6: For a target ratio $26 : 63 : 47 : 43 : 7 : 70$, mixing tree obtained by (a) *genMixing* [15] and (b) *HDA*. Gantt chart, footprint and chip area for (c,e) *NTM* and (d,f) *HDA+NTM* for the mixing tree shown in Fig. 6(a).

M_4 contains almost all mixing nodes. As a result the sample preparation completes in 10 time cycles. *HDA* transforms the mixing tree shown in Fig. 6(a) into an equivalent tree shown in Fig. 6(b), leveraging ratio preserving transformations. The Gantt chart for *HDA + NTM*, i.e., applying *NTM* to the transformed mixing tree by *HDA* (Fig. 6(d)) shows that the loads of each quadrant are balanced and the sample preparation time decreases to 4 time cycles. The footprint (of the cells used) and the chip area (estimated by the minimum bounding rectangle) of an FPVA for *NTM* and *HDA + NTM* are shown in Figs. 6(e) and 6(f), respectively.

IV. Simulation Results

We perform an extensive simulation to analyze the performance of the proposed algorithm. In our experiments, we consider a set of 50,000 or maximum possible (whichever is lower) ratios for all ratio sum $L = 64, 256$ by varying the number of reactant fluids $K = 6, 7, 8, 9$, and compute six performance parameters O (number of mixing operations), τ (time cycles for completion), C (number of cells used), V (number of valves used), A (total actuations of valves for mixing), and B (area of the minimum bounding rectangle). The proposed *NTM* is run without and with *HDA* over the large set of target ratios to evaluate the parameter values. Table I shows the values of all these parameters for $L = 64$ and $L = 256$. Note that, *HDA+NTM* increases the average number of mixing operations (O_{avg}) as it introduces new mix nodes. Therefore, *HDA+NTM* requires more valve actuations (A_{avg}) for performing the extra mixing operations. However, the average numbers of cycles (τ_{avg}), cells used (C_{avg}), and valves used (V_{avg}) are reduced by *HDA+NTM*. Reductions in the τ_{avg}, i.e., sample preparation time, and C_{avg}, V_{avg} are because *HDA+NTM* can exploit parallelism by distributing mixing nodes into different quadrants of the FPVA. Moreover, *HDA+NTM* requires less chip area (B_{avg}) (estimated as the minimum area rectangular bounding box) compared to *NTM*.

V. Conclusions

We present a technique for design automation for sample preparation on FPVA chip. The proposed method can realize any mixing tree on the FPVA without using any fluid transportation between any two cells. We also propose a heuristic that distributes mixing operations in a mixing tree to reduce the overall assay operation time and chip area for the FPVA.

References

[1] V. Gubala *et al.*, "Point of Care Diagnostics: Status and Future," *Anal. Chem.*, vol. 84, no. 2, pp. 487–515, 2012.

[2] R. H. Liu *et al.*, "Self-Contained, Fully Integrated Biochip for Sample Preparation, Polymerase Chain Reaction Amplification, and DNA Microarray Detection," *Anal. Chem.*, vol. 76, no. 7, pp. 1824–1831, 2004.

[3] H. Yin *et al.*, "Microfluidics for Single Cell Analysis," *Current Opinion in Biotechnology*, vol. 23, no. 1, pp. 110–119, 2012.

[4] C. D. Chin *et al.*, "Microfluidics-Based Diagnostics of Infectious Diseases in the Developing World," *Nat Med.*, vol. 17, no. 8, pp. 1015–1019, 2011.

[5] S. Einav *et al.*, "Discovery of a Hepatitis C Target and its Pharmacological Inhibitors by Microfluidic Affinity Analysis," *Nat Biotechnol.*, vol. 26, no. 9, pp. 1019–1027, 2008.

[6] J. Melin *et al.*, "Microfluidic Large-Scale Integration: The Evolution of Design Rules for Biological Automation," *Annual Review of Biophysics and Biomolecular Structure*, vol. 36, no. 1, pp. 213–231, 2007.

[7] L. M. Fidalgo *et al.*, "A Software-Programmable Microfluidic Device for Automated Biology," *Lab Chip*, vol. 11, pp. 1612–1619, 2011.

[8] S. Bhattacharjee *et al.*, *Algorithms for Sample Preparation with Microfluidic Lab-on-Chip*. River Publishers, 2019.

[9] C. Liu *et al.*, "Testing Microfluidic Fully Programmable Valve Arrays (FPVAs)," in *Proc. DATE*, 2017, pp. 91–96.

[10] Y. Lin *et al.*, "Block-Flushing: A Block-Based Washing Algorithm for Programmable Microfluidic Devices," in *Proc. DATE*, 2019, pp. 1531–1536.

[11] T. Tseng *et al.*, "Reliability-Aware Synthesis With Dynamic Device Mapping and Fluid Routing for Flow-Based Microfluidic Biochips," *IEEE TCAD*, vol. 35, no. 12, pp. 1981–1994, 2016.

[12] A. Grimmer *et al.*, "Close-to-optimal placement and routing for continuous-flow microfluidic biochips," in *Proc. ASP-DAC*, 2017, pp. 530–535.

[13] W. Thies *et al.*, "Abstraction Layers for Scalable Microfluidic Biocomputing," *Natural Computing*, vol. 7, no. 2, pp. 255–275, 2008.

[14] S. Roy *et al.*, "Optimization of Dilution and Mixing of Biochemical Samples using Digital Microfluidic Biochips," *IEEE TCAD*, vol. 29, no. 11, pp. 1696–1708, 2010.

[15] S. Bhattacharjee *et al.*, "Dilution and Mixing Algorithms for Flow-Based Microfluidic Biochips," *IEEE TCAD*, vol. 36, no. 4, pp. 614–627, 2017.

[16] J. P. Urbanski *et al.*, "Digital microfluidics using soft lithography," *Lab Chip*, vol. 6, no. 1, pp. 96–104, 2006.

[17] T. Fu *et al.*, "Bubble Formation and Breakup Dynamics in Microfluidic Devices: A Review," *CES*, vol. 135, pp. 343–372, 2015.

TABLE I: *NTM* vs. (*HDA + NTM*) on the basis of O_{avg}, τ_{avg}, C_{avg}, V_{avg}, A_{avg} and B_{avg} for $L = 64$ and 256 over # reagents $K = 6, 7, 8,$ and 9.

K	$L = 64$												$L = 256$											
	NTM						*HDA+NTM*						*NTM*						*HDA+NTM*					
	O_{avg}	τ_{avg}	C_{avg}	V_{avg}	A_{avg}	B_{avg}	O_{avg}	τ_{avg}	C_{avg}	V_{avg}	A_{avg}	B_{avg}	O_{avg}	τ_{avg}	C_{avg}	V_{avg}	A_{avg}	B_{avg}	O_{avg}	τ_{avg}	C_{avg}	V_{avg}	A_{avg}	B_{avg}
6	5.6	3.7	13.0	17.1	22.5	17.9	6.3	3.0	10.6	13.2	25.1	12.7	6.4	4.6	12.4	16.6	25.7	17.1	7.6	4.0	8.8	10.9	30.4	9.7
7	6.3	4.2	14.5	19.2	25.1	20.3	7.0	3.0	12.0	15.0	28.0	14.4	6.7	4.9	13.3	17.8	27.0	18.9	8.1	4.0	9.5	11.9	32.5	11.1
8	6.8	4.6	15.6	20.7	27.3	22.0	7.6	3.0	13.2	16.5	30.4	15.4	7.0	5.1	14.0	18.9	28.3	20.4	8.7	4.0	10.3	12.9	34.9	12.5
9	7.1	4.8	16.1	21.5	28.6	22.6	7.9	3.0	13.9	17.4	31.8	15.9	7.4	5.4	14.7	20.0	29.7	21.8	9.4	4.0	11.2	14.0	37.6	13.9