Recent Research and Emerging Challenges in the System-Level Design of Digital Microfluidic Biochips

Paul Pop, Elena Maftei, Jan Madsen
DTU Informatics, Technical University of Denmark
Kongens Lyngby, Denmark

Abstract—**Microfluidic biochips are replacing the conventional biochemical analyzers, and are able to integrate on-chip all the basic functions for biochemical analysis. The "digital" biochips are manipulating liquids not as a continuous flow, but as discrete droplets on a two-dimensional array of electrodes. Basic microfluidic operations, such as mixing and dilution, are performed on the array, by routing the corresponding droplets on a series of electrodes. The challenges facing biochips are similar to those faced by microelectronics some decades ago. Computer-Aided Design tools for microfluidics are in their infancy, and designers are currently using manual, bottom-up design approaches to implement such biochips. Considering their architecture and the design tasks that have to be performed, the design of digital biochips has similarities to the high-level synthesis of integrated circuits. Motivated by this similarity, a few researchers have recently started to propose approaches for the top-down design of biochips. So far, they have assumed that operations are executing on virtual modules of rectangular shape, formed by grouping adjacent electrodes, and which have a fixed placement on the array. However, operations can actually execute by routing the droplets on any sequence of electrodes on the biochip. In this paper, we outline the original module-based synthesis problem, and then we present recent work which eliminates the concept of virtual modules and allows droplets to move on the chip on any route during operation execution. We discuss the advantages of such an approach, and identify the challenges and opportunities of system-level design of digital microfluidic biochips.**

I. Introduction

Microfluidic biochips represent a promising alternative to conventional biochemical laboratories, and are able to integrate on-chip all the basic functions for biochemical analysis using microfluidics, such as, transport, dispensing, and mixing [7].

Biochips offer a number of advantages over conventional biochemical procedures. By handling small amount of fluids, they provide higher sensitivity while decreasing reagent consumption, hence reducing cost. Moreover, due to their miniaturization and automation, they can be used as point-of-care devices, in remote areas [19].

Due to these advantages, biochips are expected to revolutionize clinical diagnosis, especially immediate point-of-care diagnosis of diseases. Other emerging application areas include drug discovery, DNA analysis, and immuno-assays.

There are two generations of microfludic biochips. The first generation is based on the manipulation of continuous liquid through fabricated micro-channels, using external pressure sources or integrated mechanical micro-pumps [19]. Although adequate for many simple biochemical applications, their integrated micro-structures make continuous-flow biochips unsuitable for more complex applications, requiring complicated fluid manipulation [4]. The second generation, called digital biochips, is based on the manipulation of discrete, individually controllable droplets on a two-dimensional array of identical cells. The actuation of droplets is performed without the need of micro-structures, leading to increased scalability and flexibility compared with continuous-flow biochips [15]. In this paper, we are interested in the second generation, droplet-based Digital Microfluidic Biochips (DMBs).

Researchers have so far considered that the execution of operations is constrained to a group of adjacent electrodes forming a rectangular "virtual device". In this context, DMBs have conceptual similarities to dynamically reconfigurable field-programmable gate arrays (DR-FPGAs). However, in the case of DMBs, devices are *virtual*, as operations can be performed on any sequence of electrodes on the array. The abstraction of using virtual devices has the advantage that the same synthesis techniques used for DR-FPGAs can be adapted for DMBs. Hence, researchers have addressed the same problems: *allocation* of devices from a module library, *binding* of devices to operations, *scheduling*, *placement* and *routing*. However, recent work [11] eliminates the concept of virtual modules and allows the droplets to move on the chip on any route during operation execution. This paper presents the original module-based synthesis problem (Section III) and the typical models used (Section II). We also briefly outline recent related work in this area. Then, in Section IV we present recent work on routing-based synthesis. Finally, in the last section we discuss emerging challenges for the synthesis of DMBs.

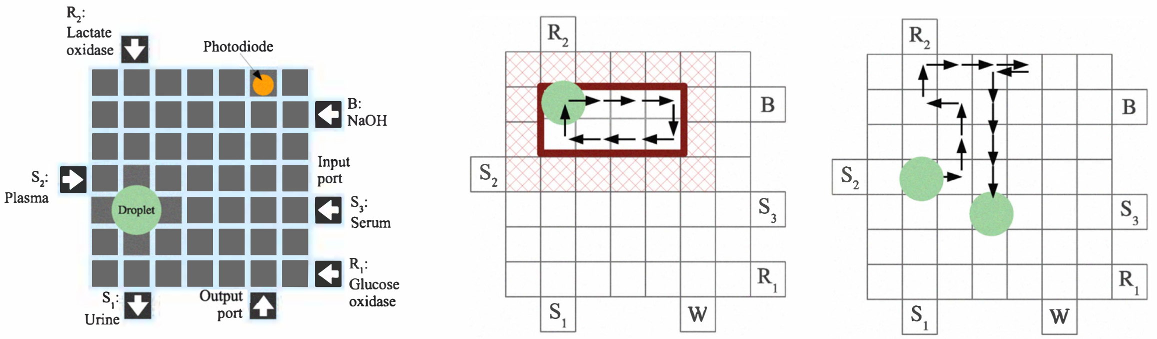
II. System Model

A. Biochip Architecture

In a digital microfluidic biochip the manipulation of liquids is performed using discrete droplets. There are several mechanisms for droplet manipulation [7]. Most of the work on DMBs considers electrowetting-on-dielectric (EWOD) [15], which is

This work was supported by the EU ArtistDesign Network of Excellence on Embedded Systems Design and by the Danish Agency for Science, Technology and Innovation, Grant No. 2106-08-0018 "ProCell".

978-1-4577-1617-1/11/$26.00 ©2011 IEEE



(a) Biochip: array of cells (b) Module-based operation execution (c) Routing-based operation execution

Fig. 1: Biochip architecture and operation execution

a promising technique that can provide high droplet speeds of up to 20 cm/s [15]. A biochip is composed of several cells, see Fig. 1a. The droplet is sandwiched between two glass plates (the top plate and the bottom plate), and moves within a filler fluid. The top plate contains a single ground electrode, while the bottom plate has several control electrodes. With EWOD, the movement of droplets is controlled by applying voltages to the required electrodes, see [15]. Several cells are put together to form a two-dimensional array (an example architecture is presented in Fig. 1a). Using EWOD manipulation, droplets can be moved to any location without the need for pumps and valves, which are required in a continuous-flow biochip. Besides the basic cell discussed previously, a chip typically contains input and output ports and detectors. The detection can be done by using, for example, a light-emitting diode beneath the bottom plate and a photodiode on the top plate.

B. Operation Execution

Using this architecture, and changing correspondingly the control voltages, all the required operations, such as transport, splitting, dispensing, mixing, and detection, can be performed. For example, mixing is done by bringing two droplets to the same location and merging them, followed by the transport of the resulted droplet over a series of electrodes. By moving the droplet, complex internal flow patterns are created (due to the formation of multilaminates), thus leading to a faster mixing [14]. Mixing through diffusion, where the resulted droplet remains on the same electrode, is very slow. The operation can be executed anywhere on the microfluidic array and is not confined to a certain area, thus we say that mixing is a "reconfigurable" operation. Another reconfigurable operation is dilution, which consists of a sequence of mixing and splitting steps. A biochemical application may also contain "non-reconfigurable" operations, that are executed on real devices, such as reservoirs or detectors.

So far, it has been considered that reconfigurable operations are performed inside virtual modules, created by grouping adjacent cells. Such a module is shown in Fig. 1b, where the droplet is routed circularly on a series of electrodes until the mixing operation is completed. The execution time of an operation on a device having a certain area has been so far determined experimentally. Table I shows an example of a characterized module library, based on the experiments performed in [14] for mixing operations.

C. Module- vs. Routing-Based Operations

During module-based operation execution, all cells inside the module are considered occupied, although the droplet uses only one cell at a time. Thus, the remaining cells cannot be used for other operations, which is inefficient since it reduces the potential for parallelism. In addition, in order to prevent the accidental merging of a droplet with another droplet in its vicinity, modules are wrapped in segregation cells (see the hashed area in Fig. 1b for an example). For further details on the fluidic constraints the reader is directed to [22].

An alternative to modules, proposed by us in [11], is routing-based operation execution. As mixing is performed by routing, an operation can be executed anywhere on the array, unconstrained by a rectangular shape representing a virtual module. This characteristic of the mixing operation is shown in Fig. 1c, where the droplet is routed freely on a sequence of electrodes, according to the shown route.

For routing-based operation execution, the completion times are not fixed, as for modules, and depend on the route taken by the droplet. In [11] we have proposed an analytical method for determining the percentage of operation completion for any given route. Our method provides safe estimates by decomposing the modules of a given module library.

D. Biochip Application Model

We model a biochemical application using an abstract model consisting of a sequencing graph [5]. The graph $\mathcal{G}(\mathcal{V},\mathcal{E})$ is directed, acyclic and polar (i.e., there is a *source node*, which

Operation	Area (cells)	Time (s)
Mixing	2 × 4	2.9
Mixing	1 × 4	4.6
Mixing	2 × 3	6.1
Mixing	2 × 2	9.95
Dispensing	–	2
Detection	1 × 1	30

TABLE I: Module library

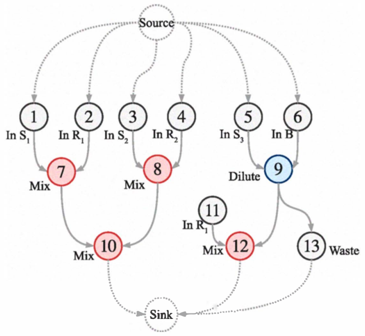


Fig. 2: Application graph

is a node that has no predecessors and a *sink node* that has no successors). Each node $O_i \in \mathcal{V}$ represents one operation.

An edge $e_{i,j} \in \mathcal{E}$ from O_i to O_j indicates that the output of operation O_i is the input of O_j. An operation can be activated after all its inputs have arrived and it issues its outputs when it terminates.

In Fig. 2 we have an example of an application graph with thirteen operations, O_1 to O_{13}. The application consists of four mixing operations (O_7, O_8, O_{10} and O_{12}), one dilution operation (O_9), six input operations (O_1, O_2, O_3, O_4, O_5, O_6 and O_{11}) and one output operation (O_{13}).

III. Synthesis Problem

The synthesis problem for DMBs can be formulated as follows. Given (1) a biochemical application modeled as a graph $\mathcal{G}(\mathcal{V},\mathcal{E})$; (2) a biochip consisting of a two-dimensional $m \times n$ array C of cells; and (3) a library $\mathcal{L}$ characterizing the completion time of operations, we are interested to synthesize that implementation Ψ, which minimizes the completion time of the application $\delta_{\mathcal{G}}$. Let us first illustrate the module-based synthesis, followed by our proposed routing-based synthesis approach, presented in Section IV-A.

The module-based synthesis problem consists in determining an implementation $\Psi = <\mathcal{A}, \mathcal{B}, \mathcal{S}, \mathcal{P}, \mathcal{R}>$, which means deciding on: (1) the allocation $\mathcal{A}$, which determines what modules from the library $\mathcal{L}$ should be used; (2) the binding $\mathcal{B}$ of each operation $O_i \in \mathcal{V}$ to a module $M_k \in \mathcal{A}$; (3) the schedule $\mathcal{S}$ of the operations, which contains the start time t_i^{start} of each operation O_i on its corresponding module; (4) the placement $\mathcal{P}$ containing the locations at which operations will be executed on the $m \times n$ array; and (5) the routes $\mathcal{R}$ taken by the droplets between modules and between modules and input/output ports.

A. Example

Let us consider the graph shown in Fig. 2, which we assume to be part of a larger biochemical application. We would like to implement the operations on the 7×7 biochip from Fig. 1a. We consider the current moment of time as being $t = 0$. For simplicity, in this example, we consider that the input operations are already assigned to the corresponding input ports. Thus, O_1 is assigned to S_1, O_2 to R_1, O_3 to S_2, O_4 to R_2, O_5 to S_3, O_6 to B and O_{11} to R_1. For the other operations in Fig. 2, the mixing operations (O_7, O_8, O_{10} and O_{12}) and the dilution operation (O_9) the module-based synthesis will have to allocate the appropriate virtual modules, bind operations to them and perform the placement and scheduling.

Let us assume that the available module library is the one captured by Table I. We assume the same execution time for mixing and dilution operations. We have to select modules from the library while trying to minimize the application completion time and place them on the 7×7 chip. A solution to the problem is presented in Fig. 3, where the following modules are used: three 1×4 mixers ($Mixer_1$, $Mixer_2$, $Mixer_3$), one 2×4 mixer ($Mixer_4$) and one 2×4 diluter ($Diluter_1$).

Considering this allocation, Fig. 3a presents the binding of operations to modules and the optimal schedule. The schedule is depicted as a Gantt chart, where, for each module, we represent the operations as rectangles with their length corresponding to the duration of the operation. Due to space reasons, we do not show the schedule of input operations, however, the starting times of the reconfigurable operations shown in Fig. 3a do take into consideration the time required for droplet dispensing. The routing times needed for merging the inputs of the operations are represented as hashed rectangles in the schedule. For example, operation O_{12} is bound to module $Mixer_4$, starts after operation O_9 is completed ($t_9^{finish} = 9.58$) and after its inputs, $e_{9,12}$ and O_{11}, are merged on the microfluidic array, thus $t_{12}^{start} = 9.60$ s. The operation takes 2.9 s, finishing at time $t_{12}^{finish} = 12.50$ s.

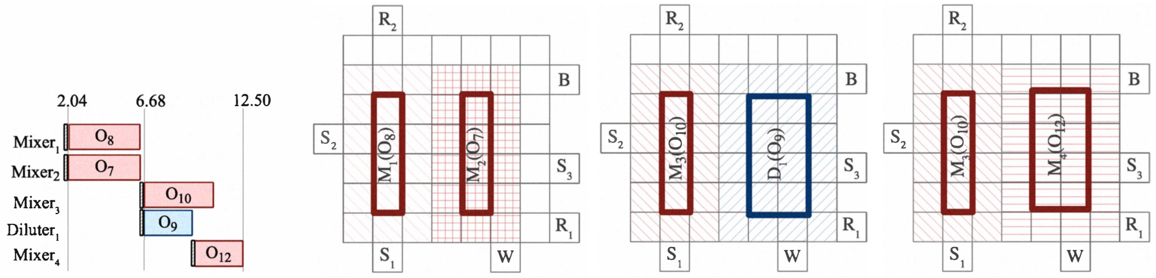
The placement for the solution is as indicated in Fig. 3b–d. Note that only two virtual devices can be placed on the biochip due to space constraints, thus only two operations can execute in parallel. In our case O_7, O_8 and O_9 could potentially be executed in parallel. If we decide to select smaller areas to increase parallelism, such as a 2×2, the execution time is much larger, e.g., 9.95 s for a 2×2, which eliminates the potential gain obtained through parallelism.

B. Related Work

The DMB synthesis problem is NP-complete [6]. Researchers have used several strategies to solve this problem. For a survey of the design tools proposed so far for digital biochips the reader is directed to [3].

A unified high-level synthesis and module placement methodology has been proposed in [17], where the focus has been on deriving an implementation that can tolerate faulty cells in the biochip array. Their algorithm was modified in [20] to include droplet-routing-aware physical design decisions. Yuh et al. [21] have proposed a synthesis and placement algorithm which uses a tree-based topological representation and is able to improve on the results from [17].

Routing has been addressed so far as a post-synthesis step, following the placement of modules on the array. Several techniques have been proposed for finding the routes on which droplets move. An A* search algorithm is presented in [2], where droplets are routed on optimal paths on the array, in the order given by assigned priorities. In [18], a modified Lee algorithm is proposed for finding the routes on which droplets



(a) Schedule (b) Placement for $t \in [2.04, 6.64]$ (c) $t \in (6.68, 9.58]$ (d) $t \in (9.60, 12.50]$

Fig. 3: Module-based synthesis example

are transported, while minimizing the number of used cells. A variant of the Open Shortest Path First network protocol is presented in [8], while a network-flow based routing algorithm is proposed in [22]. The results in [22] are improved in [6], by performing bypassability analysis while routing. Thus, at each time, the droplet less likely to block the movement of the other droplets is chosen to be scheduled.

All these methods consider that routing is performed between virtual devices whose position on the array is determined during the placement step, thus the routes have predefined fixed start- and end-points. In addition, the assumption is that the operation is executed within the virtual device.

IV. Synthesis with Reconfigurable Operations

A. Routing-Based Synthesis

Let us consider the same synthesis problem for DMBs in the case where we remove the concept of "virtual device" and allow operations to execute by routing the droplets on any sequence of electrodes on the array. Similar to the problem formulation for module-based synthesis, we want to synthesize an implementation $\Psi = <\mathcal{A},\mathcal{B},\mathcal{S},\mathcal{P},\mathcal{R}>$, deciding the allocation, binding, scheduling, placement and routing. However, there are differences when performing routing-based synthesis. The allocation, binding and placement need to be performed only for non-reconfigurable operations, such as input and detection operations. For reconfigurable operations, such as mixing and dilution, the synthesis is determined by the routes $\mathcal{R}$. For each reconfigurable operation O_i we have to determine a time-ordered list containing electrodes on which O_i is executed (i.e., a route). Thus, for reconfigurable operations, the synthesis problem is transformed into a routing problem.

Let us consider the same example presented in Section III-A. Fig. 4 shows the synthesis of the application on the 7 × 7 array. The allocation and binding of physical modules to *non-reconfigurable* operations is the same as the one presented in Section III-A. The characterization of droplet moving is performed by decomposing the module library shown in Table I, based on the analytical method in [11]. We have to find the routes $\mathcal{R}$ for all the reconfigurable operations such that the application completion time $\delta_{\mathcal{G}}$ is minimized.

Fig. 4 shows a complete solution for synthesizing the application $\mathcal{G}$ in Fig. 2 to the 7 × 7 chip. Before the reconfigurable operations O_7, O_8 and O_9 can start, we route their inputs to the locations depicted in Fig. 4b. In order to simplify the visual representation of the solution, we assume a repetitive route for the operations: the droplets corresponding to O_7, O_8 and O_9 in Fig. 4c are repeatedly routed on the shown paths for 13.58 iterations, until the mixing is completed.

After completion, the droplets resulted from the mixing operations O_7 and O_8 are routed to a common location on the chip, where they merge, forming the droplet corresponding to operation O_{10} (Fig. 4d). The dilution operation O_9 continues by splitting the mixed droplet into two droplets of intermediate concentration and equal volume, corresponding to $e_{9,12}$ and the output operation O_{13}.

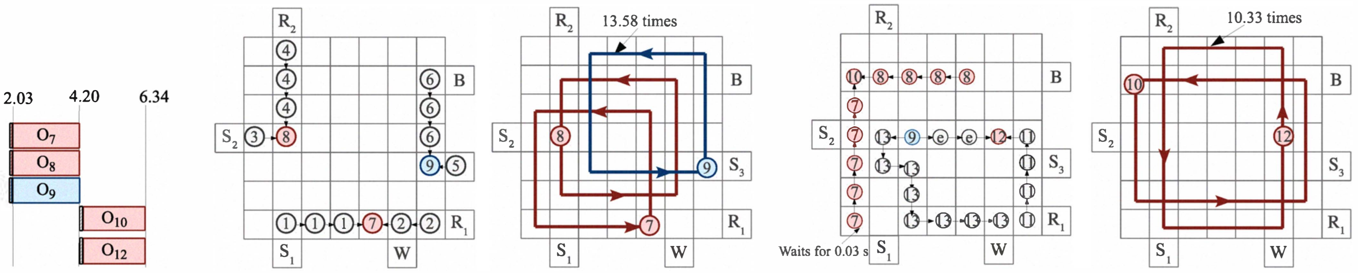
Because of simplicity reasons, in this example, the paths on which the droplets are routed while operations are executed are of rectangular shape. However, in routing-based synthesis any sequence of electrodes can be used as a path, see Fig. 1c.

The schedule of the operations is presented in Fig. 4a, where we notice that the completion time of the application is significantly reduced compared to the module-based schedule presented in Fig. 3a, 6.34 s compared to 12.50 s.

There are several reasons for this reduction. Compared to the solution in Fig. 3, operation O_9 can be executed in parallel with O_7 and O_8 in Fig. 4c. Routing-based synthesis leads to an increase in parallelism due to a more efficient use of the microfluidic array. In module-based synthesis the entire module area, including the segregation cells, is considered occupied by the operation. In routing-based operation execution we know the actual position of the droplets, therefore all the other cells can be used, as long as the droplets are not too close to each other (i.e., the microfluidic constraints from [22] are enforced). For example, in Fig. 4d the droplet corresponding to O_7 must be kept on the initial position shown from time 4.20 s until time 4.23 s, in order to prevent the accidental merging with the droplet discarded to the output reservoir (corresponding to the operation O_{13}).

B. Area-Constrained Routing

The synthesis approaches mentioned so far do not address the problem of cross-contamination of samples during the biochemical application execution. However, some biochemical applications contain liquids that adsorb on the substrate on



(a) Schedule (b) Placement at t = 2.03 (c) $t \in (2.03, 4.20]$ (d) t = 4.28 (e) $t \in (4.28, 6.34]$

Fig. 4: Routing-based synthesis example

which they are transported. Consequently, the purity of the droplets routed on the microfluidic array can be affected by the contaminated electrodes and this may lead to an erroneous outcome of the performed biochemical assay. Wash droplets are typically used in such cases to clean contaminated sites on the chip, between successive transportations of droplets [13].

In the routing-based synthesis approach presented in the previous section we have considered that operations are executed by transporting the corresponding droplets on any route on the microfluidic array. However, by constraining the execution of operations to a certain area on the microfluidic array, the contamination can be decreased. Therefore, in this section we present an approach in which droplet routes are constrained to a given area during operation execution.

Consider the example presented in Section III-A where the mixing operations O_7, O_8, O_{10} and O_{12} and the dilution operation O_9 must be performed on the microfluidic array. We consider that during the execution of an operation, the route is constrained to a certain area on the microfluidic array.

For example, in Fig. 5 the operations are performed by transporting the droplets as follows: O_7, O_8 and O_9 inside 3×4 modules, O_{10} and O_{12} inside 3×6 modules. The routes during operation execution are decided in a greedy fashion, such that the completion times of operations are minimized. The analytical method proposed in [11] is used to determine the execution time of an operation inside the area where it is executed. In our example, considering the shown movement patterns, Fig. 5a presents the schedule obtained for area-constrained routing.

V. Experimental Evaluation

In order to evaluate our proposed approaches, we have used a real-life example consisting of a *colorimetric protein assay* (103 operations) [9], utilized for measuring the concentration of a protein in a solution. In the first experiment we were interested to determine the improvement that can be obtained by using Routing-Based Synthesis (RBS) compared to Module-Based Synthesis (MBS). For MBS, we have used the Tabu Search (TS)-based synthesis approach we have proposed in [10], while the implementation of RBS is based on the Greedy Randomized Adaptive Search Procedure (GRASP) algorithm presented in [11]. The module library used in the experiments is shown in Table I. As we can see from Table II, eliminating the concept of "virtual modules" and allowing the operations to perform on any route on the microfluidic array can lead to improvements in terms of application completion time, allowing us to use smaller areas and thus reduce costs.

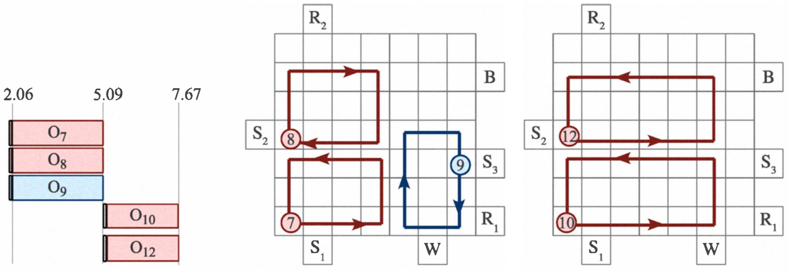
In a second experiment we were interested to determine the suitability of routing-based synthesis when contamination is a concern. Therefore, we have considered two approaches to the synthesis problem with cross-contamination avoidance: a routing-based synthesis in which droplets are moved freely during operation execution (Routing Based Synthesis with Contamination avoidance, RBSC) and an area-constrained routing-based synthesis (Area-Constrained Synthesis with Contamination avoidance, ACSC). The area-constrained routing is based on a hybrid TS/GRASP algorithm presented in [9]. For both approaches we consider that the chip is partitioned in a number of equals areas, with a wash droplet assigned to clean the contaminated electrodes in each partition. As we can see from the results presented in Table III, when synthesizing applications in which contamination avoidance must be ensured, area-constrained routing leads to better results than transporting the droplets freely on the array.

VI. Emerging Challenges

A. Synthesis with Reconfigurable Operation Execution

One of the challenges in the synthesis of biochips is the transition from the typical algorithms, used in the electronic design automation area, to synthesis approaches that take into account the reconfigurable nature of DMBs. Thus, we have proposed a routing-based model of operation execution, and we have developed several associated synthesis approaches, which progressively relax the assumption that operations execute inside fixed rectangular modules. The proposed synthesis approaches consider that (i) modules can dynamically move during their execution [10] and (ii) can have non-rectangular shapes [12]. (iii) We have relaxed the assumption that all electrodes are occupied during the operation execution, by taking into account the position of droplets inside modules [9].

Finally, (iv) we have eliminated the concept of virtual modules and have allowed the droplets to move on the chip on any route [11], as presented in Section IV-A. In this context, we have also shown how contamination can be avoided (Section IV-B). The experiments show that by considering the



(a) Schedule (b) $t \in (2.06, 5.09]$ (c) $t \in (5.18, 7.67]$

Fig. 5: Area-constrained operation execution

Area	RBS	MBS	δ %
11 × 11	113.63	184.06	38.26
11 × 10	114.33	185.91	38.50
10 × 10	115.65	208.90	44.63

TABLE II: RBS vs. MBS

Area	ACSC	RBSC	δ %
15 × 15	155.81	173.79	10.34
14 × 14	162.25	188.31	13.83
13 × 13	188.92	188.15	-0.4

TABLE III: ACSC vs. RBSC

reconfigurable nature of microfluidic operations, significant improvements can be obtained, decreasing the biochemical application completion times, reducing thus the biochip area and implementation costs. More issues remain to be solved with routing-based synthesis, such as pin-constrained synthesis and synthesis of fault-tolerant implementations.

B. Adaptive Online Synthesis

Biochip operation execution is in the order of seconds, whereas specialized heuristics for the synthesis problems can potentially obtain good results in milliseconds [16]. An interesting possibility in this context is to perform the synthesis online, while the biochemical application is running, and not offline, as it has been done so far. Such an online approach has the advantage of adaptivity, to faults in the architecture or variations in the operation execution. In addition, it opens up the possibility of fully portable point-of-care devices. Although no research has been done so far in this area, recent research [1], [23] has shown how an implementation can react to faults by switching online to recovery schedules pre-synthesized offline.

Top-down synthesis methods and the associated Computer-Aided Design tools will reduce the development costs, increase the design productivity and yield, and are the key to the further growth and market penetration of biochips.

References

[1] M. Alistar, E. Maftei, P. Pop, and J. Madsen. Synthesis of biochemical applications on digital microfluidic biochips with operation variability. In *Proceedings of the Symposium on Design Test Integration and Packaging of MEMS/MOEMS*, pages 350–357, 2010.

[2] K. F. Bohringer. Towards optimal strategies for moving droplets in digital microfluidic systems. *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1468–1474, 2004.

[3] K. Chakrabarty, R. B. Fair, and J. Zeng. Design tools for digital microfluidic biochips: Towards functional diversification and more than Moore. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 29(7):1001–1017, 2010.

[4] K. Chakrabarty and F. Su. *Digital Microfluidic Biochips: Synthesis, Testing, and Reconfiguration Techniques*. CRC Press, Boca Raton, FL, 2006.

[5] K. Chakrabarty and J. Zeng. Design automation for microfluidics-based biochips. *ACM Journal on Emerging Technologies in Computing Systems*, 1(3):186–223, 2005.

[6] M. Cho and D. Z. Pan. A high-performance droplet routing algorithm for digital microfluidic biochips. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 27(10), 2008.

[7] R. B. Fair. Digital microfluidics: is a true lab-on-a-chip possible? *Microfluidics and Nanofluidics*, 3(3):245–281, 2007.

[8] E. J. Griffith, S. Akella, and M. K. Goldberg. Performance characterization of a reconfigurable planar array digital microfluidic system. *Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 25:340–352, 2006.

[9] E. Maftei. *Synthesis of Digital Microfluidic Biochips with Reconfigurable Operation Execution*. PhD thesis, Technical University of Denmark.

[10] E. Maftei, P. Paul, and J. Madsen. Tabu search-based synthesis of dynamically reconfigurable digital microfluidic biochips. In *Proceedings of the Compilers, Architecture, and Synthesis for Embedded Systems Conference*, pages 195–203, 2009.

[11] E. Maftei, P. Paul, and J. Madsen. Routing-based synthesis of digital microfluidic biochips. In *Proceedings of the Compilers, Architecture, and Synthesis for Embedded Systems Conference*, pages 41–49, 2010.

[12] E. Maftei, P. Paul, and J. Madsen. Tabu search-based synthesis of digital microfluidic biochips with dyanmically reconfigurable non-rectangular devices. *Journal of Design Automation for Embedded Systems*, 14:287–308, 2010.

[13] H. Moon, A. R. Wheeler, R. L. Garrell, J. A. Loo, and C.-J. Kim. An integrated digital microfluidic chip for multiplexed proteomic sample preparation and analysis by MALDI-MS. *Lab on a chip*, 6(9):1213–1219, 2006.

[14] P. Paik, V. K. Pamula, and R. B. Fair. Rapid droplet mixers for digital microfluidic systems. *Lab on a Chip*, 3:253–259, 2003.

[15] M. G. Pollack, A. D. Shenderov, and R. B. Fair. Electrowetting-based actuation of droplets for integrated microfluidics. *Lab Chip Journal*, 2:96–101, 2002.

[16] C. E. Sjøgreen. Synthesis of biochemical applications with operation variability on digital microfluidic biochips. Bachelor Thesis, Technical University of Denmark, 2011.

[17] F. Su and K. Chakrabarty. Unified high-level synthesis and module placement for defect-tolerant microfluidic biochips. In *Proceedings of the Design Automation Conference*, pages 825–830, 2005.

[18] F. Su, W. Hwang, and K. Chakrabarty. Droplet routing in the synthesis of digital microfluidic biochips. In *Proceedings of the Design, Automation and Test in Europe*, volume 1, pages 73–78, 2006.

[19] T. Thorsen, S. Maerkl, and S. Quake. Microfluidic largescale integration. *Science*, 298:580–584, 2002.

[20] T. Xu and K. Chakrabarty. Integrated droplet routing and defect tolerance in the synthesis of digital microfluidic biochips. *ACM Journal on Emerging Technologies in Computing Systems*, 4(3), 2008.

[21] P.-H. Yuh, C.-L. Yang, and Y.-W. Chang. Placement of defect-tolerant digital microfluidic biochips using the T-tree formulation. *ACM Journal on Emerging Technologies in Computing Systems*, 3(3), 2007.

[22] P.-H. Yuh, C.-L. Yang, and Y.-W. Chang. Bioroute: A network-flow-based routing algorithm for the synthesis of digital microfluidic biochips. *IEEE Transactions on CAD of Integrated Circuits and Systems*, pages 1928–1941, 2008.

[23] Y. Zhao, T. Xu, and K. Chakrabarty. Integrating control-path design and error recovery in the synthesis of digital microfluidic lab-on-chip. *ACM Journal on Emerging Technologies in Computing Systems*, 6(3), 2010.