DTIP 25-27 April 2012, Cannes, France

Droplet-Aware Module-Based Synthesis for Fault-Tolerant Digital Microfluidic Biochips

Elena Maftei, Paul Pop, Jan Madsen
Technical Univ. of Denmark, DK-2800 Kgs. Lyngby
email: paul.pop@imm.dtu.dk

***Abstract*—Microfluidic biochips are replacing the conventional biochemical analyzers, and are able to integrate on-chip all the basic functions for biochemical analysis. On a "digital" biochip liquids are manipulated as discrete droplets on a two-dimensional microfluidic array of electrodes. Basic operations, such as mixing and dilution, are performed on the array, by routing the corresponding droplets on a group of electrodes, forming a virtual device. Initially researchers have ignored the locations of droplets during operation execution, and have considered that all electrodes inside devices are occupied. We have recently proposed a droplet-aware approach for the execution of operations on the microfluidic array, in which the locations of droplets inside devices are known at each time step. In this article we extend the droplet-aware approach to consider the synthesis of biochips which contain defective electrodes on the microfluidic array. We show that for such biochips knowing the exact locations of droplets during operation execution leads to significant improvements in the completion time of applications.**

I. Introduction

Microfluidic biochips represent a promising alternative to conventional biochemical laboratories, and are able to integrate on-chip all the necessary functions for biochemical analysis such as, transport, splitting, merging, dispensing, mixing, and detection, using very small amount of fluids (micro- or nanoliters). Due to the lower cost per bioassay and increased automation and miniaturization compared to biochemical laboratories, biochips are expected to revolutionize areas such as clinical diagnosis, point-of-care diagnosis of diseases and DNA and protein analysis [1].

This article focuses on the synthesis of digital microfluidic biochips (DMBs). Such devices are based on the manipulation of discrete droplets using software-driven electronic control [2].

A typical digital microfluidic biochip is composed of a two-dimensional microfluidic array of identical cells, together with reservoirs for storing the samples and reagents, as shown in Figure 1a. Each cell is composed of two parallel glass plates, see Figure 1b. The top plate contains a single ground electrode, while the bottom plate has several control electrodes. The droplet moves between the two plates using an electrical method called electrowetting-on-dielectric (EWOD). With EWOD, the movement of droplets is controlled by applying voltages to the required electrodes. For example, turning off the middle control electrode and turning on the right control electrode in Figure 1b will force the droplet to move to the right. For the details on EWOD, the reader is directed to [3].

A. Operation Execution

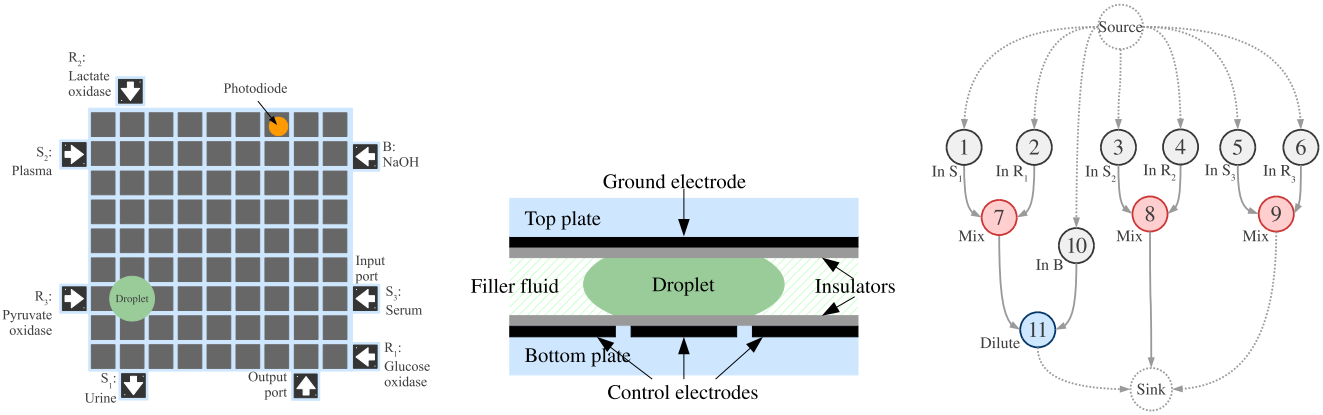
In order to perform a biochemical application on a biochip, its protocol must be known, that is the sequence of basic operations (e.g., dispensing, mixing, dilution, detection) composing the application. Such a protocol will typically be provided by the users of the biochips, e.g., biochemists, and can be modeled using a sequencing graph. For example, Figure 1c describes part of a biochemical application which consists of seven input operations (O_1–O_6, O_{10}), during which droplets are created and dispensed on the array, three mixing operations (O_7, O_8 and O_9) and one dilution operation (O_{11}).

On a digital microfluidic biochip operations such as mixing and dilution are performed by repeatedly routing the droplets on a group of adjacent electrodes, forming a virtual module. Due to the fact that any electrodes on the chip can be used for such a purpose, we say that these operations are "reconfigurable". A biochemical application may also contain "non-reconfigurable" operations, that are executed on real devices, such as reservoirs or optical detectors. The number and location of non-reconfigurable devices are decided during the design of the biochip and remain fixed after the fabrication of the device.

Initially researchers have ignored the positions of droplets inside modules, considering them as black-boxes inside which operations are executed. In order to avoid the accidental merging of droplets, it was considered that devices are surrounded by segregation areas, containing cells that cannot be used until the operations performing on the devices are completed. For example, the mixing operation O_7 in Figure 2b is executed inside the 2×3 module denoted by M_1. However, due to the 1-electrode segregation area, the device occupies 4×5 electrodes.

In [4] we have proposed a droplet-aware execution of microfluidic operations, in which the exact positions of droplets inside devices are known at each time step. For example, in Figure 3b the mixing operations O_7, O_8 and O_9 are performed by routing the droplets inside the virtual modules, according to the movement patterns described by the corresponding arrows. We avoid the accidental merging of droplets by maintaining a minimum distance between the executing operations, i.e., enforcing the fluidic constraints. The advantage of this approach is a better utilization of the space on the microfluidic array. For example the M_1 device in Figure 3b occupies the same amount of space as the 2×3 M_1 device (4×5

 ISBN:



(a) Biochip: array of cells (b) Cell architecture (c) Application graph

Fig. 1: Biochip architecture and application graph

with segregation cells) in Figure 2b. However, in the droplet-aware approach the electrodes forming the border of the device can be used for performing the operation, which leads to a faster execution time. We have shown in [4] that considering the exact locations of droplets during operation execution leads to improvements in the completion time of applications, compared to the traditional, black-box approach.

B. Fault-Tolerant Module-Based Synthesis

Considering their architecture and the design tasks that have to be performed, the design of digital microfluidic biochips has similarities to the high level synthesis of VLSI systems. Motivated by this similarity, a few researchers have recently started to propose approaches for the top-down design of such biochips. The following are the main design tasks that have been addressed [5]:

- During the design of a digital microfluidic biochip, the bioassay protocols have to be mapped to the on-chip modules. The protocols are (i) *modeled* using sequencing graphs, where each node is an operation, and each edge represents a dependency (see Figure 1c for an example).
- Once the protocol has been specified, the necessary modules for the implementation of the protocol operations will be selected from a module library, such as the one shown in Table I. This is called the (ii) *allocation* step.
- As soon as the (iii) *binding* of operations to the allocated modules is decided, the (iv) *scheduling* step determines the time duration for each bioassay operation, subject to resource constraints and precedence constraints imposed by the protocol.
- Finally, during the chip synthesis, the (v) *placement* of each module on the microfluidic array and the (vi) *routing* of droplets from one module to another have to be determined.

In our previous research on droplet-aware module-based synthesis we have considered that all the electrodes on the microfluidic array can be used for the execution of operations. However, this is not always the case, as electrodes on the array can become faulty during the fabrication of the biochip or during its operation.

As biochips are expected to be used for safety-critical applications, it is important that the faults that they can exhibit are well known and that proper actions are taken in order for the devices to function properly. There are two types of faults that can appear in a digital microfluidic biochip [6]: catastrophic and parametric.

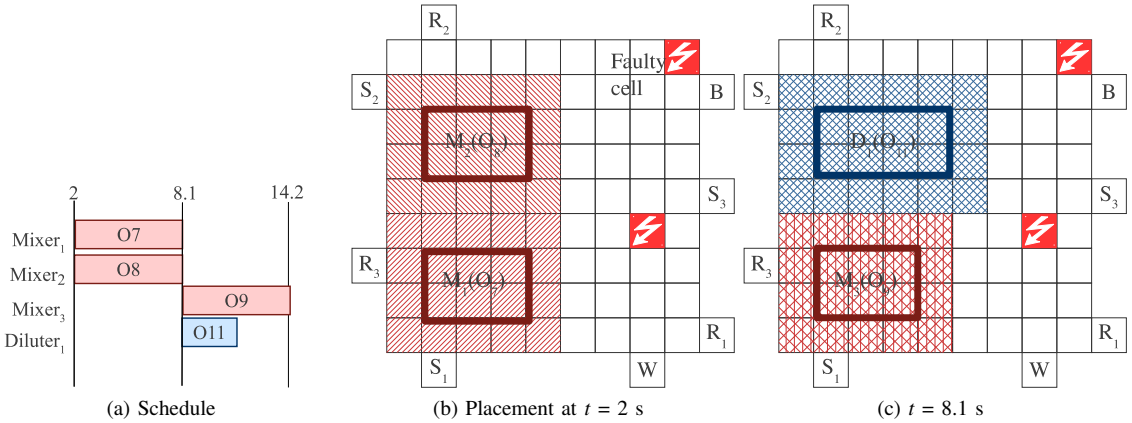
Catastrophic faults are generally caused by physical defects and lead to the complete malfunctioning of part of the biochip. Some examples of causes leading to catastrophic faults are given as follows [6]:

- Dielectric breakdown — occurs when a high voltage applied to an electrode produces the breakdown of the dielectric, creating a short between the electrode and the droplet. As a result, the movement of the droplet resting on the corresponding electrode is affected.
- Degradation of the insulator — happens gradually, during the operation of the biochip. When the degradation level reaches a certain threshold, the movement of the droplet from the corresponding electrode is affected.
- Short between two adjacent electrodes — leads to the formation of one large electrode, occupying the surface of the two electrodes. As the surface of the newly create electrode is too big, the droplet resting on it is not large enough to overlap with the adjacent electrodes. As a result, the droplet can no longer be transported.

Parametric faults do not result in the malfunctioning of the biochip. Rather, they affect the performance of the system. Examples of parametric faults include [6]:

- Increased viscosity of the fluid filler — can result in erroneous concentrations in the case of mixing operations.
- Electrode contamination — caused by certain substances (e.g., proteins, peptides) that tend to adsorb on the electrodes they are routed on. This can lead to the contamination of the droplets that are transported on the same surface, at a later time.

 ISBN:



(a) Schedule (b) Placement at t = 2 s (c) t = 8.1 s

Fig. 2: Black-box module-based synthesis

In this article we are interested in the synthesis of DMBs with catastrophic faults, i.e., chips having one or more electrodes that cannot be used for droplet actuation. Testing methods such as the one proposed in [6] can be used in order to detect faulty electrodes on the microfluidic array. Several fault-tolerant algorithms for module-based synthesis have been proposed so far [7], [8], [9]. These algorithms ignore the positions of droplets inside devices and are based on partial reconfiguration, relocating modules, if needed, in order to ensure that the droplets will not be transported over faulty electrodes. However, the disadvantage of reconfiguration is an increase in the fragmentation of the free space on the microfluidic array. Consider the 9×9 chip shown in Figure 2b, which contains two defective electrodes. The proposed black-box approaches consider that a module cannot be placed on the microfluidic array if it overlaps with a faulty electrode. This will result in a large area on the microfluidic array which cannot be used for operation execution.

In this paper, we consider a droplet-aware module-based synthesis, in which the location of droplets during operation execution is known at each time step. The advantage of this approach during the synthesis of faulty DMBs is a more efficient use of the microfluidic array. By controlling the movement of droplets we can avoid transporting them over faulty cells, without reconfiguring a large number of electrodes on the microfluidc array. This is shown in Figure 3b, where the whole surface of the chip is used for executing microfluidic operations, including the area that contains the defective electrodes.

Our synthesis method starts from a biochemical application modeled as a sequencing graph and a given biochip array containing a number of defective electrodes, and determines a complete synthesis of the application on the biochip. Compared to previous works regarding fault-tolerant DMBs, in this paper we consider the movement of droplets during operation execution. We show that by using our approach, significant improvements can be obtained in the application completion time, allowing us to use smaller area biochips and thus reduce costs.

The article is organized in five sections. Section II formulates the fault-tolerant synthesis problem for DMBs and illustrates the differences between black-box and droplet-aware operation execution. The proposed approach is presented in Section III and evaluated in Section IV. The last section presents our conclusions.

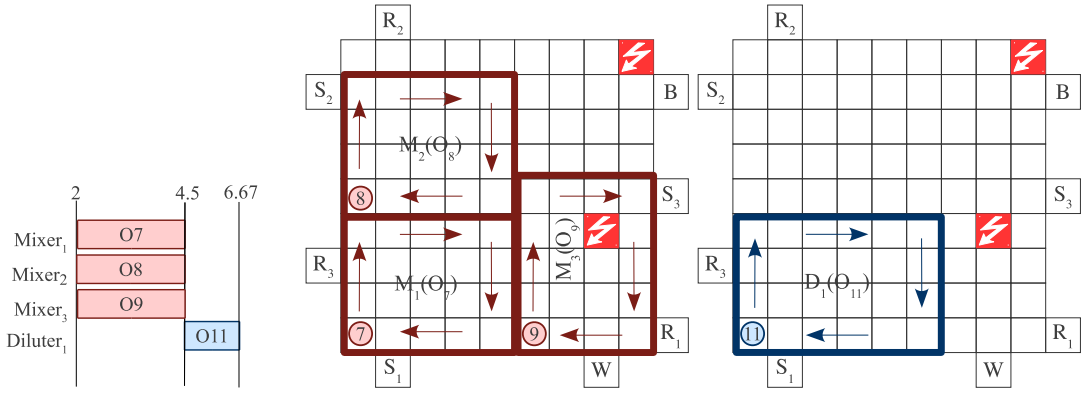
II. Problem Formulation

The problem we are addressing in this paper can be formulated as follows. Given (1) a biochemical application modeled as a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, (2) a biochip with a two-dimensional $m \times n$ array $\mathcal{C}$ of cells, (3) a characterized module library $\mathcal{L}$, and (4) the location of faulty electrodes on the array, we are interested in determining that implementation Ψ, which minimizes the completion time of the application (i.e., finishing time of the sink node, t_{sink}^{finish}).

Synthesizing an implementation $\Psi = < \mathcal{A}, \mathcal{B}, \mathcal{S}, \mathcal{P} >$ means deciding on: (1) the allocation $\mathcal{A} \subset \mathcal{L}$, 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, and (4) the placement $\mathcal{P}$ of the modules on the $m \times n$ array.

Let us consider the synthesis of the application shown in Figure 1c on the 9×9 biochip from Figure 1a. For simplicity, in this example, we consider that the input operations are already assigned to the corresponding reservoirs as follows: O_1 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 R_3, and O_{10} to B. For the other operations in Figure 1c, the mixing operations (O_7, O_8, and O_9) and the dilution operation (O_{11}) the synthesis will have to allocate the appropriate virtual modules, bind operations to them and perform the scheduling and placement.

Let us assume that the reconfigurable operations are bound to modules as follows: O_7, O_8, and O_9 to 2×3 mixers and O_{11} to a 2×4 diluter. We ignore the positions of droplets inside the modules and wrap the devices in segregation cells. Considering this allocation, Figure 2a presents the binding of operations to modules and the optimal schedule. The schedule is depicted



(a) Schedule (b) Placement at t = 2 s (c) t = 4.5 s

Fig. 3: Droplet-aware module-based synthesis

as a Gantt chart, where, for each module, we represent the operations as rectangles with their length corresponding to the duration of that operation on the module. For example, operation O_7 is bound to module $Mixer_1$ (denoted in Figure 2b by M_1), starts at t_7^{start} = 2 s (after the droplets correspoding to input operations O_1 and O_2 are dispensed on the array) and takes 6.1 s, finishing at t_7^{finish} = 8.1 s.

The placement for the solution is indicated in Figures 2b–2c. Note that although the mixing operations O_7, O_8, and O_9 could potentially be executed in parallel, this is not possible due to the existence of the faulty electrodes. As module $Mixer_3$ bound to operation O_9 cannot be placed on the array such that it does not overlap with the defective electrodes, the execution of O_9 will have to be postponed until t = 8.1 s, see Figure 2a.

The schedule in Figure 2a is optimal for the given allocation considering that the positions of droplets during operation execution are unknown. However, the solution can be further improved (see Figure 3a) by taking into account the location of droplets inside virtual modules. Let us consider the placement at time t = 2. After the modules bound to operations O_7 and O_8 have been placed on the microfluidic array as shown in Figure 3b, we would like to schedule the mixing operation O_9. In black-box module-based synthesis did was not possible, as we could not ensure that the operation will not be executed on the faulty electrodes. However, in droplet-aware module-based synthesis we can execute O_9 by repeatedly moving the droplet according to the pattern described in Figure 3b. Although module $Mixer_3$ bound to operation O_9 contains a faulty electrode, we can ensure the correct functioning of the operation by avoiding the transportation of the droplet on the faulty electrode.

Operation	Area (cells × cells)	Operation time (s)
Mixing/Dilution	2 × 4	2.9
Mixing/Dilution	1 × 4	4.6
Mixing/Dilution	2 × 3	6.1
Mixing/Dilution	2 × 2	9.95
Dispense	–	2
Detection	1 × 1	30

TABLE I: Module library

We use the execution time calculation method proposed by us in [10] to compute the completion time of the operations in Figure 3. The method takes into account the exact movement pattern of a droplet inside a device to give a safe conservative estimate of the operation completion time.

As we can see, by considering the location of droplets during operation execution we can significantly improve the completion time of the application presented in Figure 2, 6.67 s compared to 14.2 s. There are several reasons for this reduction. Compared to the solution in Figure 2, in Figure 3 we can schedule O_9 to be executed in parallel with O_7 and O_8 at t = 2. Moreover, in droplet-aware module-based synthesis we can ensure the fluidic constraints without using segregation electrodes. This allows us to consider these electrodes as part of the functional area of the devices, which leads to faster operation execution times [10].

III. Droplet-Aware Fault-Tolerant Approach

In this article we extend the droplet-aware module-based synthesis algorithm proposed by us in [4] in order to take into account faulty electrodes existent on the microfluidic array. The algorithm takes as input: (1) a biochemical application modeled as a graph, (2) a biochip consisting of a two-dimensional array of cells, (3) a module library characterizing the execution of operations, and (4) a list of faulty electrodes on the microfluidic array, and synthesizes an implementation which minimizes the completion time of the application on the biochip.

For each operation to be executed, the algorithm performs the following steps:

- binds the operation to a device from the module library, using a Tabu Search metaheuristic [11];
- schedules the execution of the operation, using a List Scheduling heuristic [12];
- places the device to which the operation is bound on the microfluidic array using the "keep all maximal empty rectangles" (KAMER) algorithm proposed in [13];

- performs the operation by routing the droplet inside the module, while avoiding the transportation of the droplet on faulty electrodes.

Let us explain the algorithm by using the example in Fig. 3b at time $t = 2$. After the droplets corresponding to the input operations O_1–O_6 are dispensed on the array, there are three operations that are ready to execute: O_7, bound to $Mixer_1$, O_8, bound to $Mixer_2$ and O_9 bound to $Mixer_3$. We use a List Scheduling heuristic to determine the order in which the operations will be executed. The List Scheduling is based on a sorted priority list containing the operations $O_i \in \mathcal{V}$ which are ready to be scheduled. The priorities of the operations are computed according to the bottom-level values of the nodes in the graph. According to these, the priority of an operation is defined as the length of the longest path from the operation to the sink of the node. In our case the mixing operation O_7 has the highest priority, as its path to the sink node includes both its operation on the 4×5 module as well as the execution of its successor, operation O_{11}, on the 4×6 diluter. Therefore the algorithm chooses O_7 to be scheduled first.

We use the KAMER algorithm proposed in [13] to determine the location of the 4×5 module bound to operation O_7 on the microfluidic array. The algorithm divides the free space on the biochip into a list of overlapping rectangles and then selects the smallest empty rectangle that accommodates the module M_i to be placed. As initially the microfluidic array is empty, the placement algorithm places the 4×5 module at the left bottom corner of the array (see Fig. 3b). Once the device is placed on the array, the operation is executed by repeatedly routing the droplet inside the device, using the movement pattern shown by the arrows.

We use a greedy approach for deciding the direction in which droplets are moved inside virtual devices at each time step. For each droplet we consider all the possible directions in which it can be moved, while ensuring that the accidential merging with neighboring droplets is avoided. We use the analytical method proposed by us in [10] for deciding which of the possible moves leads to the highest percentage of operation execution and we move the droplet in the corresponding direction. For more details regarding the algorithm the user is directed to [4].

The advantage of the presented approach compared to the conventional, black-box synthesis, is a decrease in the space fragmentation in the case of faulty electrodes. As explained in Section I-B, in the black-box approach a module cannot be placed on the microfluidic array if it overlaps with a faulty electrode. However, by using the proposed droplet-aware approach, we can place a device anywhere on the array, as long as we make sure that the droplet inside the device is not routed on a faulty electrode.

IV. Experimental Evaluation

In order to evaluate our approach we have used one real-life application and three synthetic TGFF-generated benchmarks. The algorithm was implemented in Java SE 6, running on SunFire v440 computers with UltraSPARC IIIi CPUs at 1.062 GHz and 8 GB of RAM. The module library used for all the experiments is shown in Table I.

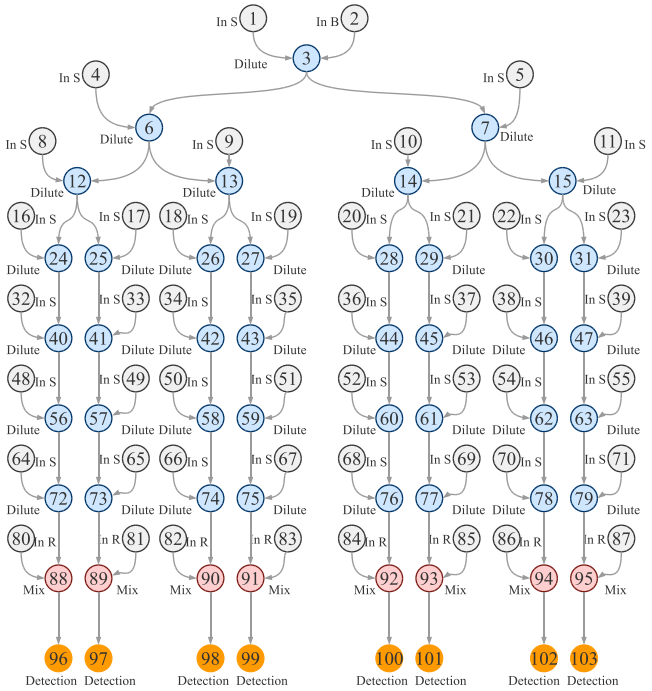


Fig. 4: Colorimetric Protein Assay

In our experiments we were interested to determine the improvement in completion time that can be obtained by considering the position of droplets inside devices when executing applications on devices with faulty electrodes. For this purpose, we have considered two approaches for the synthesis problem of fault-tolerant DMBs: a fault-tolerant droplet-aware operation execution approach (Fault-Tolerant Droplet-Aware Synthesis, FT-DAS) and a fault-tolerant black-box execution approach (Fault-Tolerant Black-Box Synthesis, FT-BBS). We have extended the droplet-aware synthesis and black-box module-based synthesis developed by us in [4] and [14], respectively, to allow the synthesis of applications on digital biochips with defective electrodes.

Table II presents the results obtained by using FT-DAS and FT-BBS on a real-life application, the colorimetric protein assay, a procedure used for determining the concentration of a certain protein in a solution. The protocol of the application consists of 103 microfluidic operations (see Fig. 4) and is based on the reaction between the protein of interest and a dye. Before being mixed with the dye, the sample is first diluted with a NaOH buffer using the mixing-splitting scheme proposed in [15]. The protocol finishes with detection operations, in which the protein concentration for the resultant solution is measured [16].

Table II shows the best and the average application completion time and the standard deviation out of 20 runs for the colorimetric protein assay, using the FT-DAS and the FT-BBS approaches. The comparison is made for two areas, using a time limit of 10 minutes for each run. In order to determine the

#faults	Area	Best		Average		Standard dev.	
		FT-DAS	FT-BBS	FT-DAS	FT-BBS	FT-DAS	FT-BBS
1	12 × 12	104.94	127.00	109.94	145.87	2.90	8.29
2		106.68	144.00	110.73	159.00	2.75	10.28
1	13 × 13	103.43	117.00	105.67	121.90	1.23	3.59
2		103.04	123.00	105.89	131.00	1.70	6.07

TABLE II: Results for the colorimetric protein assay

Nodes	#faults	$Area_1$	Best		Average		$Area_2$	Best		Average	
			FT-DAS	FT-BBS	FT-DAS	FT-BBS		FT-DAS	FT-BBS	FT-DAS	FT-BBS
20	1	8 × 8	41.65	46.00	42.84	49.22	9 × 9	41.10	43.00	41.46	46.10
	2		41.91	48.00	42.99	51.17		41.39	45.00	41.70	49.60
40	1	8 × 8	75.86	80.00	81.81	91.35	9 × 9	48.13	52.00	48.76	57.15
	2		76.03	81.00	83.20	92.75		48.41	62.00	49.38	69.95
60	1	9 × 9	82.29	88.00	86.43	92.63	10 × 10	83.16	84.00	83.86	88.42
	2		84.19	110.00	87.68	119.09		83.43	88.00	84.37	95.04

TABLE III: Results for the synthetic benchmarks

quality of the synthesis on biochips with defective electrodes, we have randomly generated a number of faulty electrodes on the microfluidic array (see column 1 of Table II).

As we can see, using the droplet-aware approach during operation execution improves the completion time of biochemical applications, compared to the black-box synthesis, particularly when the number of faulty electrodes on the array increases. For example, the improvement in the best completion time obtained out of 20 runs using FT-DAS compared to FT-BBS for the 12 × 12 area increases from 17.37% in the case of one faulty electrode to 25.91% for two faulty electrodes.

In our second set of experiments we have compared FT-DAS with FT-BBS on three synthetic applications, consisting of 20, 40 and 60 operations. The results in Table III show the best and the average completion time obtained out of 20 runs for FT-DAS and FT-BBS, using a time limit of 10 minutes. The results confirm the conclusion from Table II: as the number of faulty electrodes on the microfluidic array increases, knowing the location of each droplet on the chip becomes more important, and leads to significant improvements. For example, for the synthetic application with 60 operations, in the case of the 9 × 9 array, using FT-DAS we have obtained an improvement of 6.48% in the best completion time in the case of one defective electrode and 23.46% in the case of two defective electrodes.

V. Conclusions

In this paper we have presented a droplet-aware synthesis approach for fault-tolerant microfluidic biochips. The proposed approach considers the location of droplets on the microfluidic array during the execution of operations. One real-life application as well as a set of three synthetic benchmarks have been used to show the effectiveness of the proposed approach. We have shown that by knowing the position of droplets on the array at each time step we can efficiently avoid the defective electrodes on the microfluidic array, improving the completion time of applications compared to the traditional, back-box approach.

References

[1] K. Chakrabarty, "Design automation and test solutions for digital microfluidic biochips," *IEEE Transactions on Circuits and Systems*, vol. 57, pp. 4–17, 2010.

[2] K. Chakrabarty, R. B. Fair, and J. Zeng, "Design tools for digital microfluidic biochips: Towards functional diversification and more than Moore," *Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 29, no. 7, pp. 1001–1017, 2010.

[3] M. G. Pollack, A. D. Shenderov, and R. B. Fair, "Electrowetting-based actuation of droplets for integrated microfluidics," *Lab on Chip*, vol. 2, pp. 96–101, 2002.

[4] E. Maftei, "Synthesis of digital microfluidic biochips with reconfigurable operation execution," Ph.D. dissertation, Technical University of Denmark, 2011.

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

[6] F. Su, S. Ozev, and K. Chakrabarty, "Testing of droplet-based microelectrofluidic systems," in *Proceedings of the International Test Conference*, 2003, pp. 1192–1200.

[7] F. Su and K. Chakrabarty, "Module placement for fault-tolerant microfluidics-based biochips," *ACM Transactions on Design Automation of Electronic Systems*, vol. 11, no. 3, pp. 682–710, 2006.

[8] T. Xu and K. Chakrabarty, "Integrated droplet routing and defect tolerance in the synthesis of digital microfluidic biochips," in *Proceedings of the Design Automation Conference*, 2007, pp. 948–953.

[9] 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*, vol. 3, no. 3, 2007.

[10] E. Maftei, P. Paul, and J. Madsen, "Tabu Search-based synthesis of digital microfluidic biochips with dynamically reconfigurable non-rectangular devices," *Journal of Design Automation for Embedded Systems*, vol. 14, pp. 287–308, 2010.

[11] F. Glover and M. Laguna, *Tabu Search*. Kluwer Academic Publishers, 1997.

[12] G. D. Micheli, *Synthesis and Optimization of Digital Circuits*. McGraw-Hill Science, 1994.

[13] K. Bazargan, R. Kastner, and M. Sarrafzadeh, "Fast template placement for reconfigurable computing systems," *IEEE Design and Test of Computers*, vol. 17, no. 1, pp. 68–83, 2000.

[14] 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*, 2009, pp. 195–203.

[15] H. Ren, V. Srinivasan, and R. B. Fair, "Design and testing of an interpolating mixing architecture for electrowetting-based droplet-on-chip chemical dilution," in *Proceedings of the International Conference on Transducers, Solid-State Sensors, Actuators and Microsystems*, 2003, pp. 619–622.

[16] F. Su and K. Chakrabarty, "High-level synthesis of digital microfluidic biochips," *Journal on Emerging Technologies in Computing Systems*, vol. 3, 2008.