This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2016.2585622, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

Adaptation of Biochemical Protocols to Handle Technology-Change for Digital Microfluidics

Sukanta Bhattacharjee, Sharbatanu Chatterjee, Ansuman Banerjee, *Member, IEEE*, Tsung-Yi Ho, *Senior Member, IEEE*, Krishnendu Chakrabarty, *Fellow, IEEE*, and Bhargab B. Bhattacharya, *Fellow, IEEE*

***Abstract*—Advances in digital microfluidic (DMF) technologies offer a promising platform for a variety of biochemical applications, ranging from massively parallel DNA analysis and computational drug discovery to toxicity monitoring and medical diagnosis. In this paper, we address the migration problem that arises when the technology undergoes a change in the context of digital microfluidics. Given a bio-chemical reaction synthesized for actuation on a given DMF architecture, we discuss how the same bio-chemical reaction can be ported seamlessly to an enhanced architecture, with possible modifications to the architectural parameters (e.g., clock frequency, mixer size, and mixing time) or geometric changes (e.g., change in reservoir locations or mixer positions, inclusion of new sensors or other physical resources). Complete resynthesis of the protocol for the new architecture may often become either inefficient or even infeasible due to scalability, proprietary, security, or cost issues. We propose an adaptation method for handling such technology-changes by modifying the existing actuation sequence through an incremental procedure. The foundation of our method lies in symbolic encoding and satisfiability (SAT)-solvers, enriched with pertinent graph-theoretic and geometric techniques. This enables us to generate functionally correct solutions for the new target architecture without necessitating a complete resynthesis step, thereby enabling the utilization of these chips by users in biology who are not familiar with the on-chip synthesis tool-flow. We highlight the benefits of the proposed approach through extensive simulations on assay benchmarks.**

***Index Terms*—Bioassays, digital microfluidic biochips, formal techniques, physical design, SAT-solvers, synthesis.**

I. Introduction

RECENT advances in microfluidic technologies enabled point-of-care diagnostics [1–3], protocol evaluation [4–7], drug discovery [8], sample preparation [9, 10], and tissue engineering [11]. New-generation microfluidic technologies allow us to carry out these complex procedures on a tiny chip fostering the growth of the increasingly pervasive paradigm called lab-on-a-chip (LoC) [12]. Microfluidic technologies have experienced a sustainable growth in the past two decades [13–15]. The world market for microfluidic devices is forecast to grow to 5.7 billion by 2018 [16] and newer microfluidic systems are being developed continuously [5–7]. LoC devices can now be conveniently integrated to smartphone based platforms for data processing and communication with applications to public health monitoring and diagnostics [17].

Among several candidate technologies, digital microfluidic biochips (DMFB) have emerged as a device of choice for low-cost and accurate implementation of many systems biology and biochemical procedures [1]. DMFBs offer a programmable platform on which several biochemical assays can be performed through the electrical actuation of an electrode array. These systems, equipped with fluidic resources such as reservoirs, mixers, and sensors, can precisely control nano/pico-liter volumes of droplets on a chip for implementing a reaction flow. Given a bio-chemical assay as input, a DMFB designer has to derive a sequence of actuations that drive the fluid droplets on-chip such that the end objective of the reaction is correctly attained.

In this paper, we study the impact of technology-change on protocol realization in the context of a DMFB. While the problem of handling technology-change (or ramp) has been an active area of research in the classical VLSI design paradigm [22], or in software industry, this problem has still remained unexplored in the microfluidics community. Given a bio-chemical reaction synthesized for execution on a target microfluidic architecture, can we easily and efficiently adapt the execution sequence on a modified architecture? In the context of DMFB platform, the changes can be either parametric (in terms of clock frequency, mixing time), or geometric (e.g., change in reservoir locations, sensor positions, or addition of physical resources).

In order to tackle the impact of technology-change migration for a given protocol realization in the current DMFB scenario, the synthesis process needs to be repeated for the enhanced architecture for every bio-protocol implemented on the chip. However, full synthesis may not always be convenient because of the following four reasons: (i) scalability limitations, (ii) the need to protect intellectual property (IP), (iii) security-needs, and (iv) application-specific issues. The associated IP's for complex bioprotocols are often protected by the developers [23–27]. Also, the protocols and chips may

Manuscript received December 28, 2015; revised April 26, 2016; accepted June 5, 2016. The work of A. Banerjee, T-Y. Ho, and B. B. Bhattacharya were supported by a special grant to Nanotechnology Research Triangle from Indian Statistical Institute, Kolkata, India and by DST Indo-Taiwan Collaborative Research Project. This paper was recommended by Associate Editor Y. Wang.

S. Bhattacharjee, A. Banerjee and B. B. Bhattacharya are with the Advanced Computing and Microelectronics Unit, Indian Statistical Institute, Kolkata, India 700108. E-mail: {sukanta_r, ansuman, bhargab}@isical.ac.in

S. Chatterjee is with the Department of Computer Science and Engineering, Indian Institute of Technology, Kanpur, India. E-mail: sharbatc@iitk.ac.in

T.-Y. Ho is with the Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan. Email: tyho@cs.nthu.edu.tw

K. Chakrabarty is with the Department of Electrical and Computer Engineering, Duke University, USA. E-mail: krish@ee.duke.edu

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier xx.xxxx/TCAD.2016.xxxxxxx

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2016.2585622, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

TABLE I
IMPACT OF TECHNOLOGY-CHANGE

Parametric/geometric	Changes on the new DMFB	Consequence on existing implementation of bioassay
Actuation rate	Increases in the enhanced DMF platform.	Droplet transportation speed is increased; hence, the number of mixing cycles may need to be adjusted to maintain homogeneous mixing; otherwise, fluidic constraints may be violated on an enhanced DMF platform.
Electrode architecture	Change in size or aspect-ratio of electrodes [18] or use of different architectures [19, 20].	Old actuation sequence may fail to implement desired bioassay correctly.
Mixing time	For the same sequencing graph, mixing time needs to be adjusted depending on the fluidic properties of the input reagents [21].	Existing implementation may cause violation when different input fluids are used as viscous fluids may need longer mixing time.
Reservoirs	The number of on-chip reservoirs may increase and their locations may change on an enhanced chip [15].	Incorrect fluid dispensing may result on the enhanced DMF platform.
Dimension	Increases in the enhanced DMF platform.	Droplet routing may become erroneous on the enhanced platform.
Other on-chip resources, e.g., sensors	Their number, location, or type may change [15] because of downscaling.	Droplets may be routed to incorrect destinations; erroneous outcome on the enhanced platform.

be provided to the user by different vendors. When chips are upgraded, the protocols are to be modified to suit the new architecture. From a user's perspective, the underlying biochemical steps for an IP protocol may be unknown; the user is provided only with the actuation sequence (a control sequence consisting of binary words), and hence, the execution of a resynthesis tool may not be possible. Even if the sequence is known, because of the parametric/geometric enhancement in the new chip, the resynthesis approach may suffer from scalability bottlenecks as witnessed in earlier experimental findings [28]. On the other hand, the proposed incremental solution can re-extract the details of the protocol from an actuation sequence and is more scalable and useable for large assays, since we attempt to reuse the synthesis solution produced earlier for a small-size chip. The scalability issue also arises at the protocol-suppliers' end. For new arrivals of chips in the market, it may not be convenient for them to re-run a complete and time-consuming synthesis process for every existing bioassay and provide a patch to the user.

Both from the supplier's and the user's end, the security issues have also become very important. As reported in recent work on cyber-security [29], outside attackers may tamper the actuation sequence for a bio-protocol and thus render its outcome erroneous. In order to protect against such attacks, appropriate encryption techniques need to be deployed while producing actuation sequences. Such a scheme makes the re-extraction of the protocol even harder from a user's perspective, and complete resynthesis more complex from a protocol-supplier's perspective.

The problem of resynthesis for a protocol may also arise when fluidic parameters change. Typically, a bioassay needs to be repeated for different kinds of reagents for sample preparation This requires a complete resynthesis of the same sample preparation assay with different input parameters, such as mixing time. Recently, several techniques have been reported [30–32] that exploit the variability in the droplet transportation speed in order to reduce overall assay completion time. Also, droplet mixing times depend on the type of fluid droplets to be mixed [33]. In each of the application-specific cases above, the existing actuation sequence that runs correctly on an old DMF platform, may not emulate the bioassay correctly on the changed platform or when the fluidic properties change.

Table I summarizes the impact of various technology changes that may arise for the DMF technology. To handle such changes in the current setting, the same sequencing graph needs to pass through a complete resynthesis step [28] each time the underlying parameters change.

This paper takes a completely different solution perspective based on synthesis adaptation. In view of the fact that the given bio-chemical reaction has already been synthesized once for the older architecture, we adapt the existing actuation sequence so as to execute it on the new architecture with suitable modifications to the operation schedule, resource binding, droplet movement, and resource instantiations. It does not require the user to have detailed knowledge of the biomolecular protocol, hence it is beneficial to both designers and users. The key contributions of this article are summarized as follows.

- The impact of technology-change on DMFBs and its consequences on protocol realizations are discussed.
- We reuse existing synthesis decisions for migrating a bioassay to a new architecture by instruction rescheduling and suitable modifications of existing fluidic instructions.
- We classify different types of technology-change into two broad categories, namely parametric and geometric.
- We use the power of SAT-solvers [34, 35] to solve the symbolic constraints for handling parametric changes;
- Several graph-theoretic and geometric techniques are proposed to handle geometric changes;

Finally, we describe procedures to reconstruct the modified actuation sequence that will run correctly on the new DMFB.

The rest of the paper is organized as follows. Section II describes background material along with related prior work. Section III introduces the technology-change migration problem with a motivating example. Different aspects of technology-ramp and techniques to handle them are elaborated in Section IV and Section V. Experimental results are reported in Section VI. Finally, Section VII concludes the paper.

II. BACKGROUND AND RELATED PRIOR WORK

Given a chemical reaction involving a set of reagents and the specification of the given DMFB architecture (clock speed, area, reservoir and sensor locations), a DMFB synthesis tool [36] typically executes a series of transformations for implementing the input protocol description on the target architecture. First the behavioral description of the bioassay (operations and dependencies between them) is represented using a sequencing graph [37] with appropriate design constraints (e.g., array area, completion time, resource constraints). This is followed by a set of synthesis steps, which include resource binding, scheduling [33, 38], module placement [39], and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2016.2585622, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

droplet routing [40, 41]. Finally, a detailed layout of the DMFB along with the sequence of actuation steps are generated. The problem of synthesizing a given protocol for a target architecture satisfying various optimality issues suffers from scalability bottlenecks [28, 37]. Heuristic methods [36, 42, 43] can overcome the inherent scalability issues.

We present below a brief review of the *SimBioSys* framework, which can be used for symbolic encoding of bio-assays that we developed in [44] and use as the primary building block in this work.

A. The SimBioSys framework

The *SimBioSys* [44] framework takes, as input, an assay description of a bio-chemical protocol, the parameters of the target DMF architecture, and an actuation sequence that can realize the protocol on the target architecture by droplet movement via electrode actuations. *SimBioSys* adopts an incremental constraint-based analysis approach for checking if the actuation sequence implements the assay correctly on the target architecture. The input actuation sequence is a sequence of instructions that describe the order in which the DMF components need to be instantiated with both spatial and temporal co-ordinates, and droplet movement. For the sake of simplicity, we adopt a compact symbolic encoding of the generated actuation sequence in an intermediate language. The semantics of the instruction set is described in Table II. The instruction set considered in Table II is designed to capture the most common operation types in bioassay protocols.

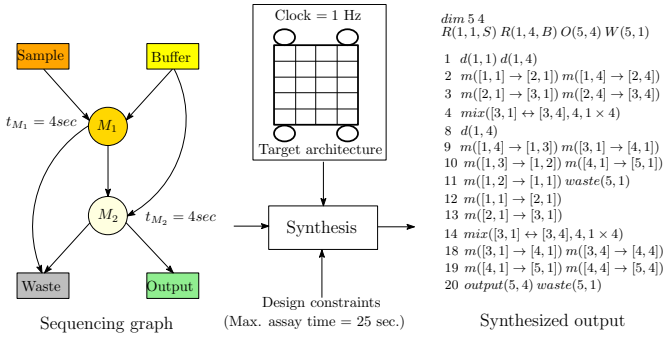

Fig. 1. Synthesis flow and representation of the synthesized output

Fig. 1 illustrates a sample synthesis flow with the generated symbolic encoding of the actuation sequence as obtained as the synthesized output. The first two lines of the symbolic encoding of the actuation sequence define the dimension of the biochip, and reservoir locations respectively. Each instruction line contains one or more symbolic fluidic instructions, separated by space. The instructions occurring on the same line are executed simultaneously. The numeric label prefixing each instruction line stands for the start time (length of each time unit is determined by the clock cycle time of the DMFB controller) of the instructions on that line. We assume that every primitive instruction, except the 'mix' instruction, takes unit time (Table II). The example below shows how the actuation sequence realizes the given bio-chemical reaction.

Example 1. The execution of the synthesized bioassay on a target DMF platform is shown in Fig. 2 where each DMFB

TABLE II
FLUIDIC INSTRUCTIONS AND THEIR INTERPRETATIONS

Encoding	Architectural description
$dim(r, c)$	Dimension of the biochip, r rows and c columns, each cell is addressed as (i, j), where $1 \leq i \leq r$ and $1 \leq j \leq c$
$R(r, c, Name)$	Reagent reservoir from which droplets can be dispensed. Droplet dispensed from reagent reservoir at (r, c) and $Name$ denotes reagent's name
$O(r, c)/W(r, c)$	Output/waste reservoir, where (r, c) denotes the cell from which a droplet can be sent to output/waste reservoir
Instruction	**Fluidic operation description**
$d(r, c)$	Dispense droplet from reservoir at location (r, c)
$m([r_1, c_1] \rightarrow [r_2, c_2])$	Transport droplet at location (r_1, c_1) to (r_2, c_2). It moves a droplet at (r_1, c_1) to one of its 4-neighbors[a] $(N_4(r_1, c_1))$ i.e., $(r_2, c_2) \in N_4(r_1, c_1)$
$mix([r_1, c_1] \leftrightarrow [r_2, c_2], t_{mix}, mtype)$	Mix two droplets at (r_1, c_1) and (r_2, c_2) for t_{mix} time steps. Initial locations are determined by the type of mixer $(mtype)$. At the end of the mixing operation, two droplets are generated and stored at locations (r_1, c_1) and (r_2, c_2) respectively
$waste(r, c)/output(r, c)$	Dispense droplet at (r, c) to its adjacent waste/output reservoir

[a] adjacent left, right, top and bottom cells

snapshot results from the execution of the fluidic instructions shown in Fig. 1. Each time step in the synthesized output (Fig. 1) is represented using t_{init}. The first two lines of the synthesized output in Fig. 1 show the DMFB size and reservoir allocations respectively. The remaining lines show the actuation sequence for each time step represented using t_{init} in Fig. 2. For a compact representation, multiple snapshots over a time interval are merged, e.g., Fig. 2(a) shows two droplets on $(3, 1)$ and $(3, 4)$ that were dispensed at $t_{init} = 1$ and moved to a cell downwards at each time step $t_{init} = 2$ and $t_{init} = 3$. Fig. 2(b) shows the mixing of two droplets on $(3, 1)$ and $(3, 4)$ by a 1×4 mixer during $[4, 7]$. Figures 2(c)-(g) depict the execution of the remaining actuations. ■

III. TECHNOLOGY-CHANGE: MOTIVATING EXAMPLES

We now introduce the problem of technology-change with an example of a parametric enhancement. Consider the sequencing graph shown in Fig. 1 where mixers M_1 and M_2 are scheduled to run for 4 seconds, i.e., $t_{M_1} = t_{M_2} = 4$ sec to attain homogeneous mixing. The synthesis tool generates the actuation sequence for a target DMFB of size 5×4 running with a 1 Hz clock. Thus, to realize the given protocol on the given architecture, 4 cycles will be needed for homogeneous mixing. It is evident from Fig. 2 that the synthesized actuation sequence implements the input sequencing graph correctly and the bioassay execution completes in 20 seconds.

Consider now an enhanced DMF platform with a 2 Hz clock. If we run the existing actuation sequence on the enhanced platform, the mixing time will be halved (2 sec.) since with a faster clock, the actuation rate is doubled. Depending on the fluidic properties, inhomogeneous mixing may occur in some mixing nodes. However, droplet routing can safely be performed at higher actuation rate [31]. In order to maintain the homogeneity of mixing, we therefore need to double the number of mixing cycles, i.e., 8 cycles are now necessary to

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2016.2585622, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

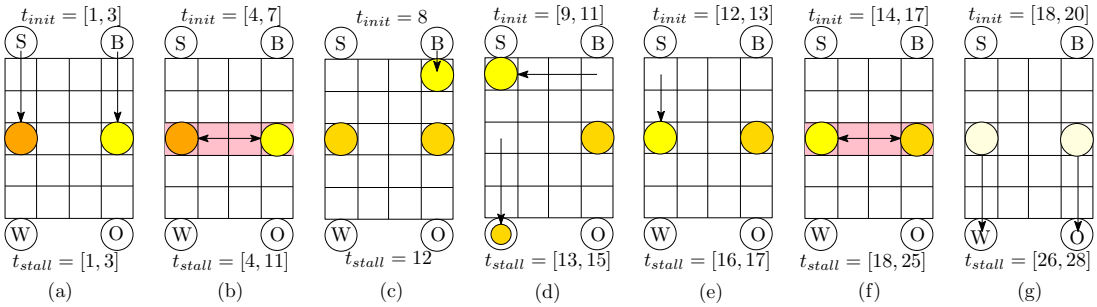

Fig. 2. Execution of the synthesized bioassay on a DMF biochip

achieve homogeneous mixing on the enhanced DMF platform. Simply increasing the mixing duration from 4 cycles to 8 does not work, since the entire actuation sequence needs to be adjusted to incorporate the effect of the increased number of cycles spent in the mixing stage; otherwise, the resulting actuations may cause an incorrect implementation of the input bioassay. The following example explains this issue.

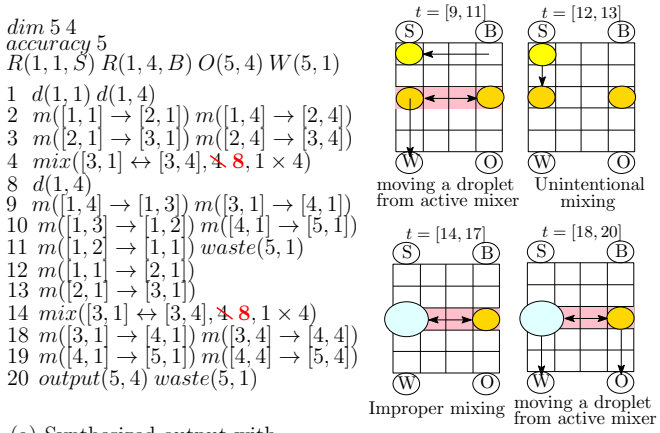

Fig. 3. Consequence of the scaling of mixing time in the actuation sequence on an enhanced DMFB

Example 2. With reference to the above example, Fig. 3(a) shows the actuation sequence after modifying the mixing duration to 8 cycles. Fig. 3(b) illustrates how several errors may creep in as a result of this adjustment, since the other instructions in the actuation sequence are still unchanged. For example, the mixer instantiated at $t = 4$, runs for eight steps on the enhanced DMFB, and now finishes at $t = 11$. However, at $t = 9$, the actuation sequence shows that a droplet will move out from the active mixer. Thus, the mixing will in effect be incomplete and lead to violations and finally, incorrectly synthesized output. Many other similar errors may as well be introduced due to this isolated adjustment of the mixing duration, without considering its global effects. ■

A. Naive corrective measure (baseline approach)

As a first solution to handle the above kind of parametric-change without resynthesis, a simple baseline strategy could be the insertion of *stall* or no-activity operations (after increasing the mixing cycles) following each mix operation in the actuation sequence such that the increased latency resulting from the mixing operation is taken care of and no further adjustments need to be made elsewhere. Note that in Example 2, we needed to increase only the mixing duration to ensure homogeneous mixing but all droplet movement instructions still remain unchanged. If we stall all instructions that are executed after the mixing operation with the increased latency (in this case, 8 - 4 = 4 cycles) of each mixing operation (M_i), we can easily overcome violations resulting out of the actuation sequences following M_i. The following example describes this stall insertion strategy.

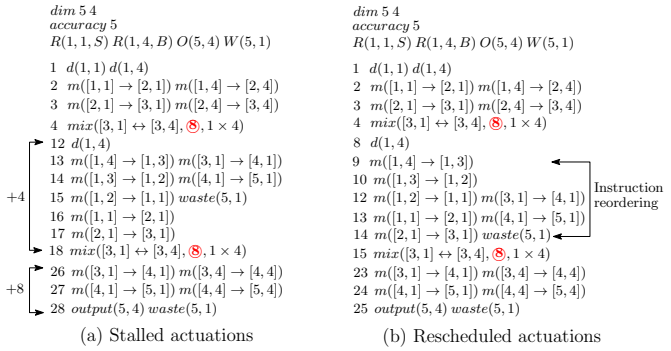

Fig. 4. Actuation sequence after (a) inserting stall cycles, (b) rescheduling fluidic operations

Example 3. In Fig. 3, the number of mixing cycles is increased by 4 for ensuring homogeneous mixing. Hence, the actuation steps following each mixing operation need to be delayed by 4 cycles for preventing any incorrect fluidic operation. Analogously, the fluidic operations following the second mixing operation need to be delayed by an additional 4 cycles to ensure correctness of the stalled sequence on the enhanced DMFB. Fig. 4(a) shows the modified actuation sequence after inserting the required number of stall cycles that take care of the increased mixing time. Additionally, Fig. 2 shows a snapshot of the modified bioassay with time steps t_{stall}. Note that the stall insertion strategy successfully takes care of the increased latency and does not have any error due to the increased mixing time. The bioassay finishes in 28 cycles, i.e, after 14 seconds on a 2 Hz clock. ■

B. Instruction reordering

A naive (baseline) approach may delay the overall assay completion time significantly. In this paper, we propose a more efficient solution based on *instruction reordering*. The main motivation behind the reordering strategy is to utilize the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2016.2585622, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

idle cycles present in different paths of the sequencing graph without compromising the correctness of fluidic operations as used during the code generation phase of compilers [45]. Fig. 4(b) shows a possible reordering of the fluidic instructions of the same assay, which finishes in 25 cycles (12.5 seconds) on the enhanced DMFB without any constraint violations.

Technology-enhancement in digital microfluidics may include a change in performance parameters (such as actuation rate, dispense time, detection time), or geometric parameters (such as grid size, sensors locations, reservoir numbers/locations, addition of resources), or both. We begin by presenting our instruction reordering method used for handling parametric-changes in Section IV and then in Section V we discuss various techniques for handling geometric-changes.

IV. Handling Parametric Changes

The main idea behind the implementation of parametric-changes is to reschedule the existing fluidic instructions so that the new sequence implements the existing assay on the new chip correctly. We use the current placement and routing decisions to help the scheduler generate a more precise schedule compared to the baseline approach of using a stalled schedule. To this effect, we design a specialized instruction reordering/scheduling technique that can schedule the fluidic instructions by reusing the existing placement and routing decisions depending on the modified assay parameters (mixing time, detection time) on the new platform. Note that optimal instruction reordering is computationally hard [46] because of the large search space. We model the instruction reordering and re-scheduling problem as an instance of the Boolean satisfiability and harness the power of modern SAT-solvers to solve this without violating any fluidic constraints.

A. Overview of the solution process

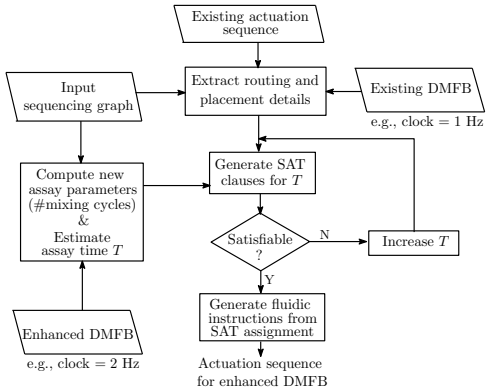

Fig. 5. The proposed parametric enhancement technique

Fig. 5 summarizes the overall methodology. We start with the existing actuation sequence, architectural description and input sequencing graph and extract the placement and routing information obtained from the first synthesis step, as obtained for the previous architecture. Additionally, we compute the modified mixing/detection times of the operations in the input sequencing graph depending on the fluidic properties, actuation rate and sensing time of the enhanced DMFB platform. More specifically, given an assay execution time T, we generate a symbolic formula to check whether there exists a valid schedule that finishes the bioassay on the new chip in T time steps. We pass on the symbolic encoding to a SAT-solver [34, 35] and if it answers in the affirmative, which indicates a valid solution exists, we can generate the new actuation sequence from the satisfying assignment. If the SAT-solver answers in the negative, it implies that no valid schedule exists for T since some constraint is violated when the existing actuation is ported with the new parameters. We increase T by 1, with the relevant changes in the symbolic encoding and check for satisfiability again. This process continues till we get a valid solution when the SAT-solver terminates with an affirmative answer. The lower bound of T can be set as the length of the critical path in the input sequencing graph after modifying the performance parameters (mixing time, detection time) of each node. Moreover, we also have an upper bound on the assay execution time on the new platform, which can be obtained based on the naive stall insertion strategy.

B. Modeling formalism

We define a set of Boolean variables to indicate the existence of a droplet/mixer on cell (x, y) at any time $t \in [1, T]$. Just using occupancy indicators for each cell is not enough, since we further need to distinguish between individual droplets and mixers on a particular cell and therefore, we assign a unique identifier (id) to each of them.

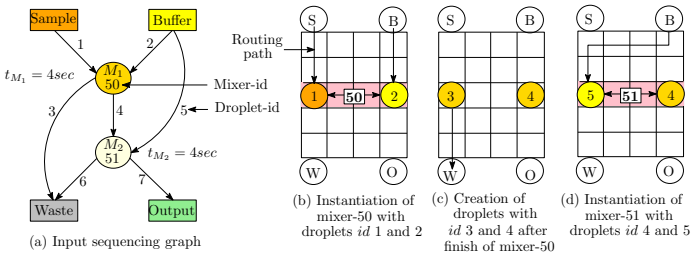

Fig. 6. Assignment of id with each droplet and mixer

Example 4. Fig. 6(a) shows the id associated with each droplet and mixer in the sequencing graph. Additionally, Figs. 6(b)-(c) show the routing path of droplets with ids $1, 2, 3, 4$ and the instantiation of a 1×4 mixer with $id = 50$ which takes in droplets with $id = 1$ and $id = 2$ as inputs and generates two droplets with $id = 3$ and $id = 4$. ■

Modeling Notation: We denote the set of all droplet and mixer identifiers as $\mathcal{ID}_{droplet}$ and $\mathcal{ID}_{mixer}$ respectively. The routing path for a droplet $i \in \mathcal{ID}_{droplet}$ is denoted as $\mathcal{P}_i$, and is simply a sequence of cell numbers through which the droplet is transported during the execution of the bio-assay. For simplicity of illustration, we consider only two types of mixers in this discussion, namely, 1×4 and 4×1 mixers[1]. We denote each mixer $i \in \mathcal{ID}_{mixer}$ as $M_i = \{(x_1, y_1), (x_2, y_2)\}$, where mixer i takes its input droplets from locations (x_1, y_1) and (x_2, y_2) and produces two droplets after mixing which reside on the same locations. Also, for any valid cell (x, y) on the DMFB plane, let $\mathcal{ID}(x, y)$ denote the set of all droplet

[1] A 1×4 (4×1) mixer is a linear array mixer that uses four horizontally (vertically) adjacent cells for mixing two droplets.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2016.2585622, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

ids that appear on the cell (x, y) during the execution of the bioassay. The example below illustrates the notation.

Example 5. Consider the sequencing graph shown in Fig. 6(a) where a unique id is assigned to each droplet and mixer. In this case, $\mathcal{ID}_{droplet} = \{1, 2, 3, \ldots, 7\}$ and $\mathcal{ID}_{mixer} = \{50, 51\}$. The routing paths for droplets with ids $1, 2, 3, 5$ are shown in Fig. 6(b)-(d) and a 1×4 mixer with $id = 50$ is shown in Fig. 6(b) that uses four cells $(3, 1), (3, 2), (3, 3)$, and $(3, 4)$. As per our notation, $M_{50} = \{(3, 1), (3, 4)\}$. The routing paths of the droplet with $id = 1$ and $id = 5$ are $\mathcal{P}_1 = \{(1, 1), (2, 1), (3, 1)\}$ and $\mathcal{P}_5 = \{(1, 4), (1, 3), (1, 2), (1, 1), (2, 1), (3, 1)\}$ respectively. As an example of the cell to droplet association set as defined above, the set of droplet ids that can appear on the cell $(1, 1)$ during execution of the bioassay is $\mathcal{ID}(1, 1) = \{1, 5\}$. ■

We define Boolean variables of the form $c_i^t(x, y)$ for indicating the presence of droplet $i \in \mathcal{ID}_{droplet}$ / mixer $i \in \mathcal{ID}_{mixer}$ on location (x, y) at t.

$$c_i^t(x, y) = \begin{cases} 1 & \text{droplet or mixer } i \text{ is present on } (x, y) \text{ at } t \\ 0 & \text{otherwise} \end{cases}$$

We assume the bioassay completion time is T. For each droplet $i \in \mathcal{ID}_{droplet}$ with its routing path $\mathcal{P}_i$, we define a set of Boolean variables $c_i^t(x, y)$, where $(x, y) \in \mathcal{P}_i$ and $1 \leq t \leq T$. Similarly, for each mixer $M_i, i \in \mathcal{ID}_{mixer}$ with the positions of its two input/output droplets, a set of Boolean variables $c_i^t(x, y)$, where $(x, y) \in M_i$ and $1 \leq t \leq T$, are defined.

Example 6. Consider Example 5. The Boolean variables for droplet $id = 1$ are $c_1^t(1, 1), c_1^t(2, 1), c_1^t(3, 1)$ where $1 \leq t \leq T$ capturing its routing path. For the mixer with $id = 50$, the variables are $c_{50}^t(3, 1)$ and $c_{50}^t(3, 4)$ where $1 \leq t \leq T$. ■

Modeling for checking constraint violations: There are a set of correctness conditions that need to guarantee that the solution returned by the SAT-solver (DMFB snapshot) is consistent, i.e., free from any violation of design rules [44]. We present a brief discussion of the modeling for each such correctness criteria below.

To ensure correct fluidic operation, we need to first guarantee that at most one droplet can appear on a particular location at any time instant t. The set of all locations where a droplet can appear during the execution of the bioassay is denoted by $\cup_{i \in \mathcal{ID}_{droplet}} \mathcal{P}_i$. For each droplet $i \in \mathcal{ID}_{droplet}$, we need to ensure that at most one droplet can appear on a location $(x, y) \in \cup_{i \in \mathcal{ID}_{droplet}} \mathcal{P}_i$ at any time instant t throughout assay execution (expressed as $1 \leq t \leq T$). Therefore, the required consistency condition can be encoded as:

$$\forall i \in \mathcal{ID}_{droplet}, \forall (x, y) \in \mathcal{P}_i$$
$$c_i^t(x, y) \implies \sum_{j \in \mathcal{ID}(x,y) \setminus \{i\}} c_j^t(x, y) = 0, \qquad 1 \leq t \leq T \quad (1)$$

Moreover, we need to make sure that each droplet i appears on at most one cell for a particular time instant t. We also need to express the constraint that each droplet i can appear on at most one cell at time t on its routing path $\mathcal{P}_i$ throughout assay execution. The following constraint captures this requirement for all droplets.

$$\sum_{(x,y) \in \mathcal{P}_i} c_i^t(x, y) \leq 1, \qquad 1 \leq t \leq T; \forall i \in \mathcal{ID}_{droplet} \quad (2)$$

The consistency conditions for droplets in active mixers can be modeled in a similar manner.

Fluidic constraint modeling: To ensure unintentional mixing of droplets, we define a set of Boolean clauses which express the fact that when a droplet $i \in \mathcal{ID}_{droplet}$ is present on a location (x, y), no other droplet should appear in any of its 8-neighborhood cells $N_8(x, y)$ all through assay execution on the routing path of each droplet. This is expressed as follows:

$$\forall i \in \mathcal{ID}_{droplet}, \forall (x, y) \in \mathcal{P}_i$$
$$c_i^t(x, y) \implies \sum_{\substack{(x',y') \in \\ N_8(x,y)}} \sum_{\substack{j \in \\ \mathcal{ID}(x',y')}} c_j^t(x', y') = 0, \ 1 \leq t \leq T \quad (3)$$

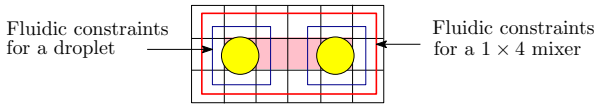

Fig. 7. Fluidic constraints on an active mixer

For simplicity, we have considered only 1×4 and 4×1 mixer. Fig. 7 pictorially marks the neighborhood box for fluidic constraints for a 1×4 mixer along with the fluidic constraints on its two input/output droplets. Such constraints can be expressed as:

$$\forall i \in \mathcal{ID}_{mixer}, \forall (x, y) \in M_i$$
$$c_i^t(x, y) \implies \sum_{\substack{(x',y') \in \\ N_8(x,y)}} \sum_{\substack{j \in \\ \mathcal{ID}(x',y')}} c_j^t(x', y') = 0, \ 1 \leq t \leq T \quad (4)$$

Modeling droplet movement: We now address the issue of modeling the movement of droplet $i \in \mathcal{ID}_{droplet}$ along the routing paths $\mathcal{P}_i$. Let us assume that the length of $|\mathcal{P}_i|$ as k and, $\mathcal{P}_i = \{(x_1^i, y_1^i), (x_2^i, y_2^i), \ldots, (x_k^i, y_k^i)\}$. When a droplet i is created on a cell (x_j^i, y_j^i), it is either dispensed from a reservoir or generated from a mixer. We model this condition using the following set of clauses. If a droplet is dispensed from a reservoir, it must not have been present at that location beforehand. Hence,

$$c_i^t(x_1^i, y_1^i) \implies \sum_{t'=1}^{t-1} c_i^{t'}(x_1^i, y_1^i) = 0, \quad 2 \leq t \leq T \quad (5)$$

Moreover, if a droplet i was generated as a result of the mix operation M_m then,

$$c_i^t(x_1^i, y_1^i) \wedge \neg c_i^{t-1}(x_1^i, y_1^i) \implies \wedge_{(x,y) \in M_m} c_m^{t-1}(x, y)$$
$$2 \leq t \leq T \quad (6)$$

In order to ensure that each droplet i follows its routing path $\mathcal{P}_i$, we define the following set of clauses. If at any time $t \in [2, T]$, a droplet i appears on cell (x_j^i, y_j^i), where $2 \leq j \leq k$, then at $t-1$, the droplet i must be either in the same location (x_j^i, y_j^i) or it was moved to (x_j^i, y_j^i) from its neighborhood location on its routing path $\mathcal{P}_i$. For each droplet $i \in \mathcal{ID}_{droplet}$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2016.2585622, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

with its routing path $\mathcal{P}_i$, we model this as follows:
For $j = 2, \ldots, k-1$

$$c_i^t(x_j^i, y_j^i) \implies c_i^{t-1}(x_{j-1}^i, y_{j-1}^i) \vee c_i^{t-1}(x_j^i, y_j^i) \vee c_i^{t-1}(x_{j+1}^i, y_{j+1}^i), \quad 2 \leq t \leq T \tag{7}$$

and for $j = k$,

$$c_i^t(x_j^i, y_j^i) \implies c_i^{t-1}(x_{j-1}^i, y_{j-1}^i) \vee c_i^{t-1}(x_j^i, y_j^i) \quad 2 \leq t \leq T \tag{8}$$

The first condition takes care of the fact that it was either at the same location or came from any of its neighborhood for any internal cell, while the following condition expresses the requirement for a boundary location. Additionally, when a droplet i reaches its destination, it either waits for mixer instantiation or is dispensed to an output/waste reservoir.

When a droplet i is used as an input to a mixer $M_m = \{(x_k^i, y_k^i), (x_l^j, y_l^j)\}$, which mixes droplets i and j, droplet i needs to wait at (x_k^i, y_k^i) until droplet j reaches at (x_l^j, y_l^j). For $2 \leq t \leq T$, this can be modeled as

$$c_i^{t-1}(x_k^i, y_k^i) \wedge \neg c_i^t(x_k^i, y_k^i) \implies c_j^{t-1}(x_l^j, y_l^j) \tag{9}$$

Note that if droplets i and j are transported to (x_k^i, y_k^i) and (x_l^j, y_l^j) at time $t-1$ respectively, mixer M_m is instantiated at time t, which is encoded in implication 10.

Droplet mix modeling: The mix operation takes two droplets as input, mixes them for a predefined number of cycles and generates two mixed droplets of equal size. A mixer can only be considered to be instantiated when both the input droplets reach the desired mixing location.

Consider the instantiation of a mixer $m \in \mathcal{ID}_{mixer}$ which receives its input droplets i_1, i_2 from locations $(x_1, y_1), (x_2, y_2)$ respectively. After running for t_m cycles, the mixer m generates the output droplets o_1 o_2 on the same locations (x_1, y_1),(x_2, y_2) respectively. The two input droplets i_1 and i_2 disappear as soon as mixer m is instantiated. Fig. 8 shows the instantiation and termination of a 1×4 mixer. Therefore, the instantiation of mixer m can be encoded as

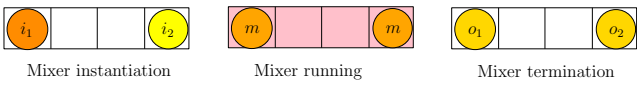

Fig. 8. Instantiation, running, and termination of a 1×4 mixer m with input droplets i_1, i_2 and output droplets o_1, o_2

follows. For $1 \leq t \leq T - t_m + 1$,

$$\begin{aligned} c_{i_1}^t(x_1, y_1) \wedge c_{i_2}^t(x_2, y_2) \implies & \\ & \wedge_{(t+1 \leq t' \leq t+t_m+1)} c_m^{t'}(x_1, y_1) \wedge c_m^{t'}(x_2, y_2) \\ & \wedge \neg c_{i_1}^{t+1}(x_1, y_1) \wedge \neg c_{i_2}^{t+1}(x_2, y_2) \\ & \wedge c_{o_1}^{t+t_m+2}(x_1, y_1) \wedge c_{o_2}^{t+t_m+2}(x_2, y_2) \end{aligned} \tag{10}$$

Note that the left hand side of the implication denotes the existence of the two input droplets (that are to be mixed) on the desired mixing location at time instant t. This implies the instantiation of the mixer during $[t+1, t+t_m+1]$ and the disappearance of the input droplets at $t+1$ and creation of the output droplets at the end of mixing i.e., at $t+t_m+2$.

C. Solving the problem

Before passing the set of constraints to a SAT-solver, we need to enforce dispensing of input droplets, instantiation of mixers and dispensing of output/waste droplets to the output/waste reservoirs. In the case of mixer instantiation, we need to ensure that both the input droplets of a mixer arrive at the mixing location at a time step t during the execution of the bioassay. Hence, for each $m \in \mathcal{ID}_{mixer}$, presence of its two input droplets i_1^m, i_2^m on $(x_1^m, y_1^m), (x_2^m, y_2^m)$ at a time step t ensures instantiation of m. Therefore, $\forall m \in \mathcal{ID}_{mixer}$

$$\sum_{1 \leq t' \leq T - t_m + 1} c_{i_1^m}^{t'}(x_1^m, y_1^m) \wedge c_{i_2^m}^{t'}(x_2^m, y_2^m) = 1 \tag{11}$$

The above clauses represent the fact that two input droplets of each mixer appear in the mixing locations exactly once during the assay period. This implies the instantiation of the mixer, which is modeled with implication 10 stated above. Moreover, for ensuring proper dispense operations to the output and waste reservoirs, we enforce that at the end, i.e. at time instant $t = T$, no droplet should be present on any cell on the DMFB grid. This constraint can be modeled by ensuring no output/waste droplet on its routing path. For all output and waste droplets i with its routing path $\mathcal{P}_i$, let (x_e^i, y_e^i) denote the last cell of $\mathcal{P}_i$ which is the dispense location corresponding to output/waste reservoir. Hence, for all output/waste droplets i,

$$\bigwedge_{\forall i} \neg c_i^T(x_e^i, y_e^i) \tag{12}$$

Algorithm 1 summarizes the proposed SAT modeling for a given assay time T.

Algorithm 1: $genSAT(T)$

Input: T: Assay time
Output: ψ: SAT instance that can schedule assay within T time steps on enhanced DMFB
1 Create Boolean variables and add to ψ;
2 Add consistency and fluidic constraint clauses to ψ;
3 Add droplet movement and mixing clauses to ψ;
4 Add mixer instantiation clauses to ψ;
5 Enforce flushing of all droplets at T and add to ψ;
6 **return** ψ;

V. Handling Geometric Changes

In the previous subsection, we have addressed the possible migration problems that may arise when performance parameters of a given DMF platform are upgraded, and discussed a SAT-based instruction-rescheduling solution. However, the new biochip may have a different geometrical layout of physical resources, or it may contain some additional resources such as reservoirs and sensors. Without any loss of generality, we assume that in the enhanced platform (dimension: $r_n \times c_n$), resources such as the reservoirs, sensors or operational area (in both X and Y dimensions), either increase in number or in size (or remain the same) compared to the old chip (dimension: $r_o \times c_o$). However, their relative locations may change. Although in geometric-migration, our objective is to execute the existing bioassay on the new platform, different placement of physical resources (sensors, reservoirs) in the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2016.2585622, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

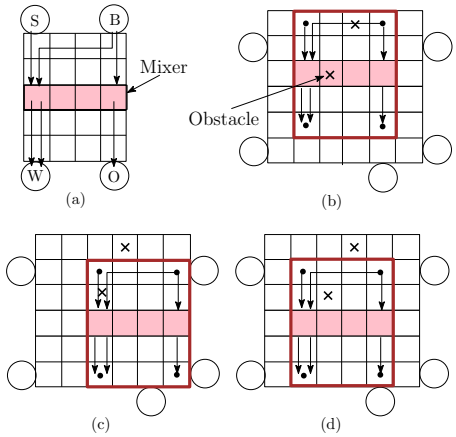

Fig. 9. (a) Existing DMF platform (b) Violation of routing path and placement (c) Preservation of routing (d) Preservation of routing and placement

new DMFB prevents precise emulation. Hence, we identify a sub-grid of size $(r_o \times c_o)$ in the new DMFB, where the old chip can be emulated by preserving most of the existing routing and placement solutions. The initial sub-grid can be chosen as an isothetic rectangle of size $(r_o \times c_o)$ containing the minimum number of obstacles (physical resources) in the new DMFB. Note that such sub-grid can be located on the new DMFB in $O(n \log n)$ time [47], where n is the number of obstacles. If multiple sub-grids that contain the minimum number of obstacles are found, we choose the one that is likely to reduce the total re-routing and re-placement overhead, as explained below. Note that mixer re-placement overhead is much higher than re-routing since in addition to finding a place the mixer, we need to determine the routing paths to and from the mixer. We assign a cost $c \in \mathbb{Z}^+$ for each routing path that needs to be re-routed and $4c + c = 5c$ for each relocated mixer ($4c$ for re-routing four input-output droplet-paths, and c for finding new location for the mixer). In our experiment we choose $c = 1$, for reducing the total number of re-routing and re-placement on the new chip. Example 7 explains this selection strategy in more detail.

Example 7. Fig. 9(a) shows an existing synthesis solution on a 5×4 DMFB, and Figs. 9(b)-(d) show its emulation on a 6×6 DMFB with different choices of the initial sub-grid. Note that mixer cells are highlighted and a cell marked with $\times$ represents an obstacle. Reservoir assignment can be done after choosing the initial sub-grid (detailed discussion provided in Section V-A). Due to the presence of two obstacles, the choice of the sub-grid in Fig. 9(b) causes both routing and placement violations when the old solution is emulated. Note that there exist three different 5×4 sub-grids on a 6×6 DMFB (two of them are shown in Fig. 9(c)-(d)) that contain only one (in this case, minimum) obstacle. The choice of sub-grid shown in Fig. 9(c) needs two re-routing, i.e., the cost is $2 * c$, whereas Fig. 9(d) does not require any re-routing or re-placement, i.e., the cost is 0. Hence, the sub-grid shown in Fig. 9(d) is chosen for emulating the existing synthesis on the new chip. ■

For a given choice of sub-grid, the migration problem has the following three subproblems: (i) reservoir assignment, (ii) re-routing of droplet pathways, and (iii) re-placement of virtual

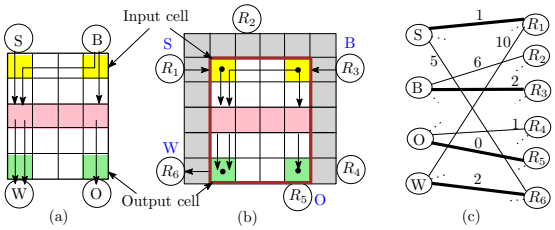

Fig. 10. (a) Existing DMF platform (b) Enhanced DMF platform after reservoir assignment (c) Weighted bipartite graph

resources such as instantiation of mixers. We discuss these problems in details below.

A. *Reservoir assignment*

Given a sub-grid on the enhanced DMF platform, we need to reassign the input and output reservoirs so that the total length of routing paths between reservoirs and input/output cells of the sub-grid is minimized. Fig. 10(a) and Fig. 10(b) highlight the input/output cells on the new and old platforms, respectively. Our objective is to find a shortest-path routing of the droplets from reservoirs to the input/output cells of the sub-grid using the unused cells on the new chip only. We formulate this problem in terms of matching in a weighted bipartite graph [48]. Construct a weighted bipartite graph $G(V, E)$, $(V = V_1 \cup V_2$ and $E = V_1 \times V_2)$, where $V_1(V_2)$ is the set of reservoirs in the existing DMF platform (enhanced platform). Note that an edge $e = (v_1, v_2)$, where $v_1 \in V_1$ and $v_2 \in V_2$, denotes the assignment of reservoir v_1 of the old chip to reservoir v_2 of the new chip. The weight assigned to each edge (v_1, v_2) indicates the total length of all routing paths of the droplet(s) that was(were) dispensed to(from) the reservoir v_2 on the new platform. For example, let us assume that two droplets are dispensed from v_1 in the old assay, and v_2 is the corresponding reservoir on the new chip. Let the length of routing paths (using only unused cell of the new chip) from v_2 to the input cells of the sub-grid on the new chip be l_1 and l_2 respectively. Then $w(v_1, v_2) = l_1 + l_2$. Note that in the presence of obstacles in the new chip, a routing path may not exist through the unused cells alone. In such cases, a large cost is assigned to (v_1, v_2), since we need to re-route droplets using the cells from the sub-grid. Once G is constructed, a minimum-cost bipartite matching gives the desired reservoir assignment. The following example illustrates the procedure.

Example 8. Let us consider an existing DMF platform of size 5×4 along with synthesis results shown in Fig. 10(a). The enhanced DMF platform of dimension 6×6 and a sub-grid of size 5×4 are also shown in Fig. 10(b), where the entry and exit points to the sub-grid are highlighted and marked with black dots. The unused cells on the new chip are shaded. We construct a weighted bipartite graph (Fig. 10(c)) for assignment of reservoirs. Note that the old (new) platform has 4 (6) reservoirs. Hence, $V_1 = \{S, B, O, W\}$ and $V_2 = \{R_1, R_2, \ldots, R_6\}$. Few edges along with their weights are shown in Fig. 10(c). Note that $w(B, R_3)$ is 2, which corresponds to the total path length for two buffer droplets. However, $w(S, R_6)$ is 5 because only one sample droplet is transported through five electrodes. A minimum-cost matching

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2016.2585622, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

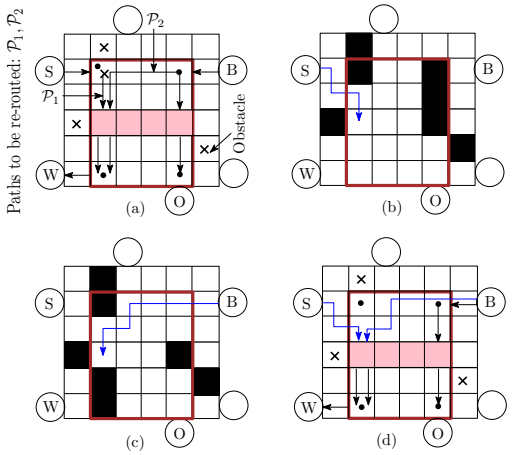

Fig. 11. (a) Enhanced DMFB after reservoir assignment (b) Combined snapshot of the new chip during the time interval $[1, 3]$ (b) Combined snapshot of the new chip during the time interval $[8, 13]$ (d) Enhanced DMFB after re-routing of obstructed paths

is shown using bold edges on Fig. 10(c), which provides the desired reservoir assignment that minimizes input/output droplet-routing time. ■

B. Re-routing

In this subsection, we discuss obstacle-avoidance droplet routing techniques between a pair of source and destination cells. Initially, a heuristic routing technique is applied for finding an obstacle-avoidance droplet routing path of a droplet from its source to its destination. When routing paths in the old chip are mimicked into the new sub-grid, re-routing of some of them may become necessary due to the presence of the obstacle(s) on the routing path. Moreover, droplet transportation between reservoirs and the sub-grid may require re-routing.

Example 9. Consider Fig. 11(a), which shows the emulation of the old synthesis solution on a (5×4) grid-size (Fig. 1 and Fig. 2) on the enhanced DMF platform with obstacles $(\mathcal{C} = \{(1, 2), (2, 2), (4, 1), (5, 6)\})$. Note that the two droplet-paths from the sample reservoir $(\mathcal{P}_1)$ and buffer reservoir $(\mathcal{P}_2)$ are obstructed as the cell $(2, 1)$ is occupied by an obstacle on the new chip. However, the other paths remain unchanged. We need to re-route the paths $\mathcal{P}_1$ and $\mathcal{P}_2$. ■

Let us assume that we need to re-route a routing path $\mathcal{P}_i$ that transports droplet i from its source location (x_s^i, y_s^i) to its destination cell (x_e^i, y_e^i). We emulate the old synthesis solution on the sub-grid of the new chip. Therefore, for each time instant, a snapshot of DMFB can be found from the existing actuation sequence by simulating it on the predefined sub-grid of the new chip. Moreover, the lifetime of each droplet (time interval between creation and deletion of a droplet) is known. Let for droplet i, its lifetime is $[t_s^i, t_e^i]$. For re-routing of droplet i, we can combine DMFB snapshots of the new chip between $[t_s^i, t_e^i]$ and find the cell usage. Additionally, the cells corresponding to routing cells of droplet i are removed from the combined snapshot. A shortest path between (x_s, y_s) and (x_e, y_e) can be used as new routing path of droplet i.

Example 10. Consider Example 9, where the starting location of routing path $\mathcal{P}_1$ i.e., $(2, 2)$ on the sub-grid of the new chip is obstructed. Hence, we need to re-route the droplet from the sample reservoir $(2, 1)$ to destination cell $(4, 2)$. Moreover the lifetime of the droplet $(id = 1)$ is $t_{init} = [1, 3]$ and the combined snapshots on the old DMFB is shown in Fig. 2(a). Fig. 11(b) shows the combined snapshot of the old synthesis on the sub-grid of the new chip during the time interval $[1, 3]$. The new routing path for droplet 1 becomes $\mathcal{P}_1 = \{(2, 1), (3, 1), (3, 2), (4, 2)\}$. In case of re-routing of $\mathcal{P}_2$, the combined snapshot is shown in Fig. 11(c), which is generated from the snapshots of the old synthesis (Figs. 2(c)-(e)) on the new chip during the time interval $[8, 13]$. The re-routed path and the complete re-routing on the new platform are shown in Fig. 11(c) and Fig. 11(d), respectively. ■

The proposed heuristic re-routing technique may fail if there does not exist any path from (x_s^i, y_s^i) to (x_e^i, y_e^i) on the combined snapshot. In this case, we re-route the routing path of droplet i using the SAT-based routing technique. Given a set of obstacle cells $\mathcal{O}$ in the sub-grid, we modify the clauses representing the routing paths as given in Section IV-B for obstacle-free movement [41] of a droplet $i \in \mathcal{ID}_{droplet}$ from its source (x_s, y_s) to its destination (x_e, y_e). Note that the constraints $(1)-(5)$, described in Section IV-B, restrict droplet movement on its routing path. We relax the clauses so that a droplet can move to any of its four-neighborhood cells avoiding obstacles, or remain at its location. Moreover, the routing path can use any cell on the new DMF platform. Let the dimension of the new DMF platform be $(r_n \times c_n)$. In order to implement obstacle-free movement for droplet i, we define Boolean variables $c_i^t(x, y)$, where $1 \leq t \leq T$, for all obstacle-free cells $(\mathcal{C})$ in the new DMFB, i.e., $(x, y) \in \mathcal{C}$, where $\mathcal{C} = (\{1, 2, \ldots, r_n\} \times \{1, 2, \ldots, c_n\}) \setminus \mathcal{O}$. Additionally, constraints $(3) - (4)$ are replaced by the following constraints to ensure obstacle-free movement of i. Thus, for $(x, y) \in \mathcal{C} \setminus \{(x_s^i, y_s^i)\}$,

$$c_i^t(x, y) \implies \bigvee_{(x', y') \in \{(x,y)\} \cup (N_4(x,y) \cap \mathcal{C})} c_i^{t-1}(x', y') \qquad (13)$$

where $1 \leq t \leq T$.

Moreover, the consistency and fluidic constraints (Section IV-B) need to be modified.

C. Re-placement

Re-placement of virtual resources (e.g., mixers) are necessary when a mixing location on the sub-grid of the enhanced chip contains one or more obstacles. Furthermore, when a mixer in the sub-grid needs to be relocated, it is necessary to modify the corresponding routing operations depending on the mixer location. Given the architectural description of the old chip and the actuation sequence, it is easy to identify which mixing operations and its corresponding routing operations need to be modified. We present below a heuristic re-placement technique that helps to mitigate the overhead of exact synthesis [28] for mixer re-placement.

Let $M_1, M_2, \ldots, M_n$ be n mixers instantiated during the assay operations. Let the start and end time of M_i be $t_s(M_i)$ and $t_e(M_i)$, respectively. Out of n mixers, let k of them be replaced, i.e., $M_{i_1}, M_{i_2}, \ldots, M_{i_k}$ where $t_s(M_{i_1}) \leq t_s(M_{i_2}) \leq \ldots \leq t_s(M_{i_k})$. Assume that mixers $M_{i_1}, M_{i_2}, \ldots, M_{i_{j-1}}$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2016.2585622, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

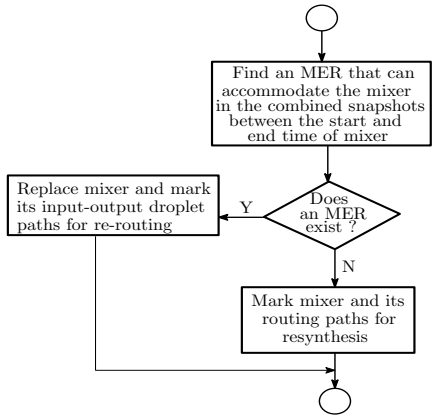

Fig. 12. Re-placement of an obstructed mixer

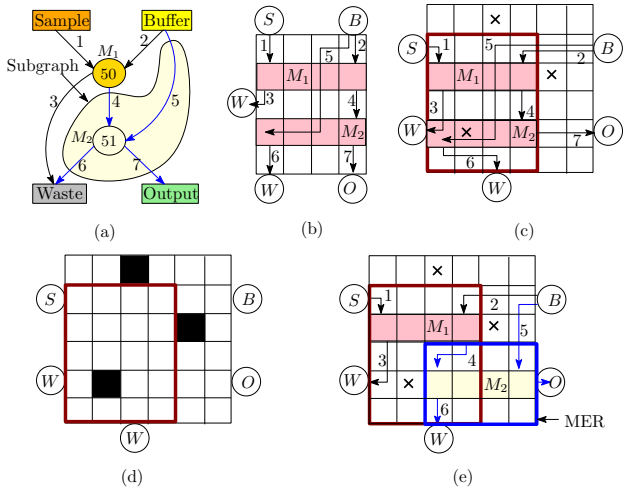

Fig. 13. (a) Subgraph of the input sequencing graph to be re-synthesized (b) Existing synthesis for the old chip (c) Reservoir assignment (d) Combined snapshot of the new chip during $[t_s(M_2), t_e(M_2)]$ (e) Re-routing and synthesis for the new chip

are already processed, and we need to relocate M_{i_j}, where $i_j \leq i_k$. As discussed in the re-routing step of geometric migration, we can combine DMFB snapshots of the new chip between $[t_s(M_{i_j}), t_e(M_{i_j})]$ for finding the used cells during the concerned time interval. Additionally, the cells corresponding to mixer M_{i_j} location and its incoming and outgoing droplet paths are removed from the combined snapshot. Note that the unoccupied cells in the combined snapshot can be used for replacing of M_{i_j}. Next, a maximal-empty-rectangle (MER) that can accommodate M_{i_j} is determined for relocating M_{i_j}. All MERs in such an environment can be identified using a plane-sweep technique in $O(n \log n + \mathcal{R})$ time [49], where $\mathcal{R}$ is the number of reported MERs and n is the number of blocked cells. In the unlikely event when no such MER is found, relocation of M_{i_j} is performed by invoking exact synthesis [28]. The outline of the mixer re-placement technique is summarized in Fig. 12. The remaining mixers are processed in a similar fashion. Note that an MER-based mixer re-placement strategy can reduce the number of mixers to be relocated, and effectively reduce the performance bottleneck in remapping the assay on the new chip. Let us consider the following illustrative example.

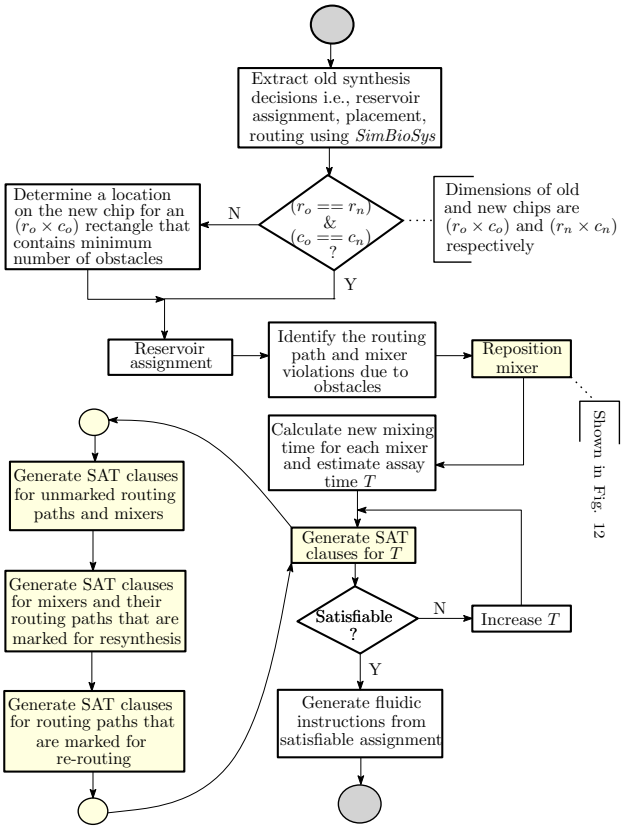

Fig. 14. Overall flow for the technology-change process on DMFB.

Example 11. Consider the input sequencing graph and its synthesis result for a 5×4 DMFB as shown in Fig. 13(a) and Fig. 13(b), respectively. A new DMFB of size 6×6 along with a 5×4 sub-grid, where the existing synthesis (Fig. 13(b)) is deemed to be emulated, is shown in Fig. 13(c). Note that after reservoir assignment, the emulation of entire assay cannot be completed as one of the cells in mixer M_2 is occupied by an obstacle. Hence, we need to relocate M_2 and re-route its incoming and outgoing droplets (mixer M_2 and its corresponding droplets are shown as a subgraph of the sequencing graph in Fig. 13(a)). The combined snapshot during $[t_s(M_2), t_e(M_2)]$ along with obstacles is shown in Fig. 13(d). Fig. 13(e) shows an MER where mixer M_2 and its input/output droplet paths are relocated. ■

The outline of the full migration flow is shown in Fig. 14.

VI. Experimental Evaluation

We implement the proposed methodology using Python 2.7. Clauses are represented using a Python library Z3Py that can handle the logic for cardinality constraints using the SMT solver Z3 [34]. As an implementation platform, we have used 2 GHz Intel Core i5 workstation with 16 GB of memory running 64 bit Ubuntu 14.04 LTS.

In our experiments, we consider 36 variants of six bioassays to be run on enhanced-DMFBs of different sizes. We assume that the old actuation sequence and the layout descriptions are known. Table III provides the attributes of several benchmarks that we have used, the size (in terms of the number of different operations) of their corresponding sequencing graphs, as well

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2016.2585622, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

TABLE III
BENCHMARK ASSAYS USED IN EXPERIMENT

Assay name	Number of assay operations				#routing paths	Length of the critical path	Old chip size	#new chips considered	#cycles in old synthesis
	Input	Mix-split	Detection	Output					
A1. In-vitro[a][37]	8	4	4	8	16	16	6×6	2	40
A2. Dilution using *REMIA* [9]	4	5	0	4	14	35	5×5	2	55
A3. Linear dilution [50]	8	11	0	8	30	47	8×13	2	70
A4. Protein [37]	12	11	2	12	34	51	6×6	2	82
A5. PCR mixture [51]	8	7	0	7	22	40	8×15	2	62
A6. PCR mixture droplet streaming [51]	13	15	0	13	43	48	8×15	2	150

[a] we have considered 2 input samples and 2 reagents

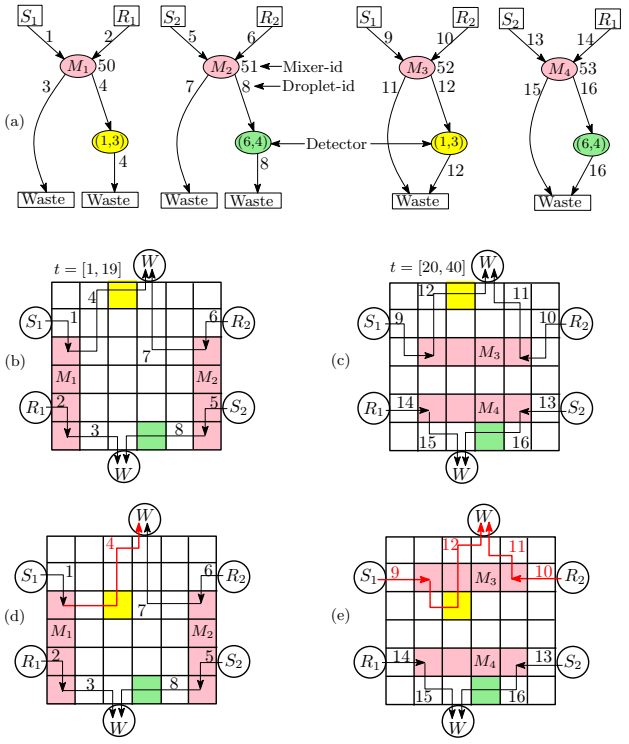

Fig. 15. (a) Sequencing graph of in-vitro multiplexed assay (b)-(c) synthesis results on an existing DMFB of dimension 6×6 (c) proposed solution for handling technological changes (sensor relocation) on a new DMFB

as the grid-size on which the tasks were realized.

A. Comparison of the proposed technique with exact synthesis

In the first set of experiments, we consider two bioassays (first two assays in Table III) with a simple set of assay operations, and perform several experiments for demonstrating parametric and geometric migration of the existing assays on some representative new platforms with a set of arbitrary technology changes. We illustrate a complete migration flow for the in-vitro assay [37]. The sequencing graph extracted from the existing synthesis result is shown in Fig. 15(a) along with the id assignment for each droplet and mixer. The synthesis outcomes (routing path, mixer placement and detector assignment) on the existing DMFB for the first two subgraphs and last two subgraphs in Fig. 15(a) are shown in Fig. 15(b) and Fig. 15(c), respectively, where t denotes the time steps in the existing actuation sequence. We consider a new platform (Fig. 15(d)) on which the position of a sensor is changed (e.g., previously it was on $(1, 3)$ but now it is relocated to $(3, 3)$), and the clock frequency is increased by 2X, i.e., the number of mixing and detection cycles are doubled. Note that the path of droplet $id = 4$ i.e., $\mathcal{P}_4$ needs to be re-routed through the new sensor location $(3, 3)$ on the new platform. The modified routing path obtained by our re-routing technique (Section V-B) is shown in Fig. 15(d). Moreover, we need to relocate mixer M_3 along with its associated four routing paths, as the corresponding mixing cell is occupied by the sensor on the new chip. A MER-based re-placement technique is used (Section V-C) to determine the new position of the mixer as shown in Fig. 15(e). After re-placement of M_3, we need to re-route the routing paths for droplets with $id = 9, 10, 11, 12$. The desired re-routing is shown in Fig. 15(d). Note that only five routing paths (out of sixteen) and only one mixer (out of four mixers) need to be relocated while handling geometric-changes in the new platform. After such incremental re-routing and mixer re-placement, we need to update the performance parameters (mixing time, detection time) as applicable to the new platform and perform parametric-changes (adapting with higher clock frequency) for obtaining the new actuation sequence. Table IV reports comparative results for the proposed technique and exact synthesis [28] for different DMF platforms. The results show the advantages of the proposed migration technique over exact synthesis in terms of CPU-time while giving a small penalty in the the number of cycles. The first row of Table IV does not consider any enhancement on the new chip. Nevertheless, it can speed-up the existing actuation sequence; note that old actuation takes 40 cycles, whereas our solution requires 30 cycles. As the proposed technique reduces assay-completion time for a given placement and routing, this can be as a fast optimizer for actuation codes as well.

Consider the highlighted row in Table IV; let each mixing(detection) operation takes 4(2) cycles in the existing synthesis, respectively. An exact synthesis tool [28] finishes the input bioassay within 32 time steps (#cycles), which is minimum on the enhanced platform. Note that in the new platform, mixing and detection cycles need to be doubled because of the increase in the clock frequency by 2X. However, our proposed migration technique modifies the old assay to execute it correctly on the new platform within 35 time steps.

We have also compared the exact synthesis method [28] with our proposed method for the sequencing graph produced by the REMIA [9] dilution algorithm. The detailed experimental setup including input sequencing graph, existing routing path, mixer placements, and new DMFBs can be found in [52]. As evident from Table IV, the proposed migration method outperforms the exact synthesis method [28] for 12 test cases

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2016.2585622, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

TABLE IV
COMPARATIVE RESULTS OF THE PROPOSED TECHNIQUE WITH EXACT SYNTHESIS [28]

Parametric enhancement								
Assay	Clock frequency (f_{clock})	New chip size	Exact synthesis [28]			Proposed		
			CPU time	#cycles	#vars	CPU time	#cycles	#vars
A1.	1 Hz	6×6	2250 sec.	26	19240	32 sec.	30	1800
	2 Hz		2908 sec.	32	23680	33 sec.	33	2160
A2.	1 Hz	5×5	1359 sec.	42	20454	63 sec.	47	2538
	2 Hz		3875 sec.	62	30194	102 sec.	67	3618

Geometric enhancement													
Assay	f_{clock}	New DMFB		Corrections				Exact synthesis [28]			Proposed		
		Dimension	#Obstacles	Reservoir assignment	Re-routing		Re-placement	CPU time	#cycles	#vars	CPU time	#cycles	#vars
					#paths	SAT-based?							
A1.	1 Hz	6×6	2	×	5	×	1	2100 sec.	25	18500	26 sec.	26	1508
	1 Hz	7×7	3	✓	12	×	0	3358 sec.	27	27000	33 sec.	30	2100
A2.	1 Hz	6×6	0	✓	0	×	0	4180 sec.	40	27680	78 sec.	50	2900
	1 Hz	6×6	1	✓	3[a]	✓	0	3230 sec.	40	26920	320 sec.	50	7000

Parametric and geometric enhancement													
Assay	f_{clock}	New DMFB		Corrections				Exact synthesis [28]			Proposed		
		Dimension	#Obstacles	Reservoir assignment	Re-routing		Re-placement	CPU time	#cycles	#vars	CPU time	#cycles	#vars
					#paths	SAT-based?							
A1.	2 Hz	6×6	2	×	5	×	1	2940 sec.	32	23680	32 sec.	35	2030
	2 Hz	7×7	3	✓	12	×	0	5385 sec.	33	33000	46 sec.	42	2940
A2.	2 Hz	6×6	0	✓	0	×	0	9967 sec.	60	44400	107 sec.	70	4060
	2 Hz	6×6	1	✓	3[a]	✓	0	9658 sec.	60	43200	587 sec.	70	9800

[a] SAT based re-routing is used for all three paths

TABLE V
PERFORMANCE SUMMARY

Parametric enhancement							
Assay	Clock frequency (f_{clock})	New chip size	Stall insertion (#cycles)	Proposed			
				CPU time	#cycles	#vars	Speedup
A3.	1 Hz	8×13	70	165 sec.	51[a]	7700	27.14%
	2 Hz		102	251 sec.	71	10720	30.39%
A4.	1 Hz	6×6	82	181 sec.	58[a]	7250	26.82%
	2 Hz		110	251 sec.	84	10500	23.63%
A5.	1 Hz	8×15	62	93 sec.	47[a]	5734	24.19%
	2 Hz		86	152 sec.	63	7686	26.74%
A6.	1 Hz	8×15	150	657 sec.	74[a]	18324	50.66%
	2 Hz		182	917 sec.	98	24460	46.15%

Parametric and geometric enhancement													
Assay	New DMFB		Corrections				Proposed (f_{clock} = 1Hz)			Proposed (f_{clock} = 2Hz)			
	Dimension	#Obstacles	Reservoir assignment	Re-routing		Re-placement	CPU time	#cycles	#vars	CPU time	#cycles	#vars	
				#path	SAT-based?								
A3.	8×13	1	✓	3	×	0	98 sec.	51	6986	231 sec.	71	9726	
	8×13	2	✓	7	×	1	120 sec.	51	7190	222 sec.	71	10010	
A4.	6×6	2	×	5	×	1	277 sec.	60	7610	354 sec.	86	11094	
	8×8	2	✓	2	×	0	605 sec.	63	10016	1404 sec.	89	14150	
A5.	10×17	0	✓	0	×	0	100 sec.	50	6950	123 sec.	66	9174	
	8×15	1	×	8	×	2	77 sec.	47	5734	128 sec.	63	7686	
A6.	8×15	2	×	5[b]	✓	1	7105 sec.	72	34200	8472 sec.	96	45600	
	10×17	3	✓	5	×	1	720 sec.	76	19760	1019 sec.	100	26000	

[a] used as code optimizer; [b] two paths are re-routed using SAT based re-routing technique

in terms of both CPU-time and the number of variables used, though it pays a small penalty in terms of the number of cycles in assay-completion time.

B. Scalability evaluation

In order to study the scalability and performance of the proposed method, we run our experiments on 24 additional test cases of real-life biochemical protocols (Table V) for which an exact synthesis approach [28] fails to produce solutions within reasonable time limits. The details of experimental set-up such as sequencing graphs, routing paths, mixer placements, detector assignments for both the old and enhanced architectures can be found in [52]. For parametric-changes, we have considered changes in actuation rate, detection time, mixing (detection) cycles. The first part of Table V demonstrates that the SAT-based instruction-reordering technique outperforms the baseline stall-insertion strategy. In order to handle geometric-changes, we may need to perform re-placement, rerouting, and reservoir assignment as described in Section V. The second part of Table V shows that the proposed algorithms can tackle geometric-changes quite satisfactorily even for large-size target architectures. Note that the CPU-time increases only for a few cases where a SAT-based engine had to be invoked to solve the re-routing problem during geometric-migration.

VII. CONCLUSION

We propose an automated migration technique for adapting an existing bioassay to an enhanced DMF platform. We show that prior synthesis results can be effectively utilized to incrementally construct the modified actuation sequence, instead

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2016.2585622, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

of running an expensive resynthesis procedure for the new architecture. We utilize the deductive power of Boolean satisfiability solvers to handle parametric and geometric changes in the new DMF platform, so that existing bioassays can be adapted quickly. To the best of our knowledge, no technique has been reported that can perform such incremental migration. Until now, resynthesis was the only way to address migration issues. Moreover, our experimental results establish the scalability of the proposed technique on a number of test cases. This research may open up further avenues in the context of technology-change for microfluidic biochips. One challenging problem in this area is to characterize the technology-changes that can be handled by an incremental migration approach.

REFERENCES

[1] R. Sista, Z. Hua, P. Thwar, A. Sudarsan, V. Srinivasan, A. Eckhardt, M. Pollack, and V. Pamula, "Development of a digital microfluidic platform for point of care testing," *Lab Chip*, vol. 8, pp. 2091–2104, 2008.

[2] M. Mohammadi, H. Madadi, and J. Casals-Terre, "Microfluidic point-of-care blood panel based on a novel technique: Reversible electro-osmotic flow," *Biomicrofluidics*, vol. 9, 2015.

[3] V. K. Bhutani, M. Kaplan, B. Glader, M. Cotten, J. Kleinert, and V. Pamula, "Point-of-care quantitative measure of glucose-6-phosphate dehydrogenase enzyme deficiency," *Pediatrics*, vol. 136, no. 5, pp. 1268–1275, 2015.

[4] L. Mazutis, J. Gilbert, W. L. Ung, D. A. Weitz, A. D. Griffiths, and J. A. Heyman, "Single-cell analysis and sorting using droplet-based microfluidics," *Nature*, vol. 8, pp. 870–891, 2013.

[5] A. H. C. Ng, M. D. Chamberlain, H. Situ, V. Lee, and A. R. Wheeler, "Digital microfluidic immunocytochemistry in single cells," *Nature Communications*, vol. 6, 2015.

[6] A. P. Aijian and R. L. Garrell, "Digital microfluidics for automated hanging drop cell spheroid culture," *Journal of Laboratory Automation*, vol. 20, no. 3, pp. 283–295, 2015.

[7] F. Piraino, F. Volpetti, C. Watson, and S. J. Maerkl, "A digital-analog microfluidic platform for patient-centric multiplexed biomarker diagnostics of ultralow volume samples," *ACS Nano*, vol. 10, no. 1, pp. 1699–1710, 2016.

[8] P. Neui, S. Giselbrecht, K. Lnge, T. J. Huang, and A. Manz, "Revisiting lab-on-a-chip technology for drug discovery," *Nature*, vol. 11, pp. 620–632, 2012.

[9] J.-D. Huang, C.-H. Liu, and T.-W. Chiang, "Reactant minimization during sample preparation on digital microfluidic biochips using skewed mixing trees," in *Proc. ICCAD*, Nov. 2012, pp. 377–383.

[10] A. E. Kirby and A. R. Wheeler, "Digital microfluidics: An emerging sample preparation platform for mass spectrometry," *Analytical Chemistry*, vol. 85, no. 13, pp. 6178–6184, 2013.

[11] N. W. Choi, M. Cabodi, B. Held, J. P. Gleghorn, L. J. Bonassar, and A. D. Stroock, "Microfluidic scaffolds for tissue engineering," *Nature*, vol. 6, pp. 908–915, 2007.

[12] E. K. Sackmann, A. L. Fulton, and D. J. Beebe, "The present and future role of microfluidics in biomedical research," *Nature*, vol. 507, pp. 181–189, Mar. 2014.

[13] J. Ducrée, R. Zengerle, and J. Newman, *FlowMap: Microfluidics Roadmap for the Life Sciences*. FlowMap Consortium, 2004. [Online]. Available: https://microfluidics-roadmap.com

[14] J. Melin and S. R. Quake, "Microfluidic large-scale integration: the evolution of design rules for biological automation," *Lab Chip*, vol. 36, pp. 213–231, 2007.

[15] X. J. Li and Y. Zhou, Eds., *Microfluidic Devices for Biomedical Applications*. Woodhead Publishing, 2013.

[16] Yole développement. [Online]. Available: http://www.yole.fr/

[17] D. Erickson, D. O'Dell, L. Jiang, V. Oncescu, A. Gumus, S. Lee, M. Mancuso, and S. Mehta, "Smartphone technology can be transformative to the deployment of lab-on-chip diagnostics," *Lab Chip*, vol. 14, pp. 3159–3164, 2014.

[18] P. Paik, V. K. Pamula, and R. B. Fair, "Rapid droplet mixers for digital microfluidic systems," *Lab Chip*, vol. 3, pp. 253–259, 2003.

[19] G. Wang, D. Teng, and S.-K. Fan, "Digital microfluidic operations on micro-electrode dot array architecture," *Nanobiotechnology, IET*, vol. 5, no. 4, pp. 152–160, 2011.

[20] S. Roy, B. B. Bhattacharya, and K. Chakrabarty, "Optimization of dilution and mixing of biochemical samples using digital microfluidic biochips," *IEEE Trans. on CAD*, vol. 29, no. 11, pp. 1696–1708, 2010.

[21] C.-Y. Lee, C.-L. Chang, Y.-N. Wang, and L.-M. Fu, "Microfluidic mixing: A review," *Int. Journal of Molecular Sciences*, vol. 12, no. 5, pp. 3263–3287, 2011.

[22] P.-H. Wu, S.-Y. Bai, and T.-Y. Ho, "A topology-based eco routing methodology for mask cost minimization," in *Proc. ASP-DAC*, 2014, pp. 507–512.

[23] HemoGenix. In vitro predictive toxicity testing and drug development. [Online]. Available: http://www.hemogenix.com/app_InvitroTox.php

[24] Quest Diagnostics. Quest diagnostics receives FDA clearance for new bio-intact PTH parathyroid hormone test kit. [Online]. Available: http://newsroom.questdiagnostics.com/press-releases?item=94459

[25] MB Research Labs. Corrositex - Non-animal dermal corrosivity test for packing group classification. [Online]. Available: http://www.mbresearch.com/corrositex.htm

[26] Trovagene. Trovagene announces plans to develop proprietary test for high risk HPV carrier screening from urine. [Online]. Available: http://trovagene.investorroom.com/2012-05-02-Trovagene-Announces-Plans-to-Develop-Proprietary-Test-for-High-Risk- HPV-Carrier-Screening-from-Urine

[27] Invitae. When technology enables better medical practice. [Online]. Available: https://www.invitae.com/en/assay/

[28] O. Keszocze, R. Wille, T.-Y. Ho, and R. Drechsler, "Exact one-pass synthesis of digital microfluidic biochips," in *Proc. DAC*, 2014, pp. 1–6.

[29] S. S. Ali, M. Ibrahim, O. Sinanoglu, K. Chakrabarty, and R. Karri, "Security assessment of cyberphysical digitalmicrofluidic biochips," *IEEE/ACM Tran. on Computational Biology and Bioinformatics*, 2015 (to appear).

[30] S.-H. Yeh, J.-W. Chang, T.-W. Huang, and T.-Y. Ho, "Voltage-aware chip-level design for reliability-driven pin-constrained EWOD chips," in *Proc. ICCAD*, 2012, pp. 353–360.

[31] J. Fiske, D. Grissom, and P. Brisk, "Exploring speed and energy tradeoffs in droplet transport for digital microfluidic biochips," in *Proc. ASP-DAC*, 2014, pp. 231–237.

[32] Y. Luo, K. Chakrabarty, and T.-Y. Ho, "Biochemistry synthesis on a cyberphysical digital microfluidics platform under completion-time uncertainties in fluidic operations," *IEEE Trans. on CAD*, vol. 33, no. 6, pp. 903–916, 2014.

[33] F. Su and K. Chakrabarty, "High-level synthesis of digital microfluidic biochips," *JETC*, vol. 3, no. 4, 2008.

[34] L. M. de Moura and N. Bjørner, "Z3: An efficient SMT solver," in *Proc. TACAS*, 2008, pp. 337–340.

[35] Y. S. Mahajan, Z. Fu, and S. Malik, "Zchaff2004: An efficient SAT solver," in *Proc. SAT*, 2004, pp. 360–375.

[36] D. Grissom and P. Brisk, "Fast online synthesis of digital microfluidic biochips," *IEEE Trans. on CAD*, vol. 33, no. 3, pp. 356–369, 2014.

[37] K. Chakrabarty and F. Su, *Digital Microfluidic Biochips - Synthesis, Testing, and Reconfiguration Techniques*. CRC Press, 2007.

[38] D. Grissom and P. Brisk, "Path scheduling on digital microfluidic biochips," in *Proc. DAC*, 2012, pp. 26–35.

[39] Y.-H. Chen, C.-L. Hsu, L.-C. Tsai, T.-W. Huang, and T.-Y. Ho, "A reliability-oriented placement algorithm for reconfigurable digital microfluidic biochips using 3-d deferred decision making technique," *IEEE Trans. on CAD*, vol. 32, no. 8, pp. 1151–1162, 2013.

[40] F. Su, W. L. Hwang, and K. Chakrabarty, "Droplet routing in the synthesis of digital microfluidic biochips," in *Proc. DATE*, 2006, pp. 323–328.

[41] O. Keszocze, R. Wille, and R. Drechsler, "Exact routing for digital microfluidic biochips with temporary blockages," in *Proc. ICCAD*, 2014, pp. 405–410.

[42] R. Wille, O. Keszocze, T. Boehnisch, A. Kroker, and R. Drechsler, "Scalable one-pass synthesis for digital microfluidic biochips," *Design Test, IEEE*, vol. 32, no. 6, pp. 41–50, 2015.

[43] F. Su and K. Chakrabarty, "Unified high-level synthesis and module placement for defect-tolerant microfluidic biochips," in *Proc. DAC*, 2005, pp. 825–830.

[44] S. Bhattacharjee, A. Banerjee, K. Chakrabarty, and B. B. Bhattacharya, "Correctness checking of bio-chemical protocol realizations on a digital microfluidic biochip," in *Proc. VLSI Design*, 2014, pp. 504–509.

[45] K. Wilken, J. Liu, and M. Heffernan, "Optimal instruction scheduling using integer programming," in *Proc. PLDI*, 2000, pp. 121–133.

[46] D. Bernstein, M. Rodeh, and I. Gertner, "On the complexity of scheduling problems for parallel/pipelined machines," *IEEE Trans. Comput.*, vol. 38, no. 9, pp. 1308–1313, 1989.

[47] S. C. Nandy and B. B. Bhattacharya, "A unified algorithm for finding maximum and minimum point enclosing rectangles and cuboids," *Comput. and Math. Appl.*, vol. 29, no. 8, pp. 45–61, 1995.

[48] A. V. Goldberg and R. E. Tarjan, "Finding minimum-cost circulations by canceling negative cycles," *J. ACM*, vol. 36, no. 4, pp. 873–886, 1989.

[49] S. C. Nandy, B. B. Bhattacharya, and S. Ray, "Efficient algorithms for identifying all maximal isothetic empty rectangles in VLSI layout design," in *Proc. FSTTCS, LNCS*, vol. 472 (Springer), 1990, pp. 255–269.

[50] S. Bhattacharjee, A. Banerjee, T.-Y. Ho, K. Chakrabarty, and B. Bhattacharya, "On producing linear dilution gradient of a sample with a digital microfluidic biochip," in *Proc. ISED*, 2013, pp. 77–81.

[51] S. Roy, S. Kumar, P. P. Chakrabarti, B. B. Bhattacharya, and K. Chakrabarty, "Demand-driven mixture preparation and droplet streaming using digital microfluidic biochips," in *Proc. DAC*, 2014, pp. 1–6.

[52] Details of experimental setup. [Online]. Available: http://www.iscal.ac.in/∼sukanta_r/experimental_setup.pdf

Sukanta Bhattacharjee received the B.Sc. (Hons.) degree in computer science and the B.Tech. degree in computer science and engineering from the University of Calcutta, India, in 2006 and 2009, respectively. He received the M.Tech. degree in computer science from the Indian Statistical Institute, Kolkata, India, in 2012. He is currently a doctoral student in computer science at the same institute.

His current research interests include algorithms for computer-aided design of microfluidic biochips.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2016.2585622, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

Sharbatanu Chatterjee received the B. Tech. degree in computer science and engineering from the Indian Institute of Technology, Kanpur in 2016.

His current research interests include optimization techniques, machine learning and applications of mathematical and computing knowledge to problems in various fields, particularly neuroscience.

Ansuman Banerjee (M'11) is an Associate Professor at the Advanced Computing and Microelectronics Unit, Indian Statistical Institute Kolkata. He received his B.E. from Jadavpur University, and M.S. and Ph.D. degrees from the Indian Institute of Technology Kharagpur – all in Computer Science. Prior to joining Indian Statistical Institute, he served as a research fellow at NUS Singapore and as a member of management staff at Interra Systems India Pvt. Ltd. His research interests include formal methods, design verification and automata theory.

Tsung-Yi Ho (M'08-SM'12) received his Ph.D. in Electrical Engineering from National Taiwan University in 2005. He is a Professor with the Department of Computer Science of National Tsing Hua University, Hsinchu, Taiwan. His research interests include design automation and test for microfluidic biochips and nanometer integrated circuits. He has been the recipient of the Invitational Fellowship of the Japan Society for the Promotion of Science (JSPS), the Humboldt Research Fellowship by the Alexander von Humboldt Foundation, and the Hans Fischer Fellow by the Institute of Advanced Study of the Technical University of Munich. He was a recipient of the Best Paper Awards at the VLSI Test Symposium (VTS) in 2013 and *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* in 2015. He served as a Distinguished Visitor of the IEEE Computer Society for 2013-2015, the Chair of the IEEE Computer Society Tainan Chapter for 2013-2015, and the Chair of the ACM SIGDA Taiwan Chapter for 2014-2015. Currently he serves as an ACM Distinguished Speaker, a Distinguished Lecturer of the IEEE Circuits and Systems Society, and Associate Editor of the *ACM Journal on Emerging Technologies in Computing Systems*, *ACM Transactions on Design Automation of Electronic Systems*, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, and *IEEE Transactions on Very Large Scale Integration Systems*, Guest Editor of *IEEE Design & Test of Computers*, and the Technical Program Committees of major conferences.

Krishnendu Chakrabarty (F'08) received the B. Tech. degree from the Indian Institute of Technology, Kharagpur, in 1990, and the M.S.E. and Ph.D. degrees from the University of Michigan, Ann Arbor, in 1992 and 1995, respectively. He is now the William H. Younger Distinguished Professor of Engineering in the Department of Electrical and Computer Engineering and Professor of Computer Science at Duke University. Prof. Chakrabarty is a recipient of the National Science Foundation Early Faculty (CAREER) award, the Office of Naval Research Young Investigator award, the Humboldt Research Award from the Alexander von Humboldt Foundation, Germany, the IEEE Transactions on CAD Donald O. Pederson Best Paper award (2015), and 11 best paper awards at major IEEE conferences. He is also a recipient of the IEEE Computer Society Technical Achievement Award (2015) and the Distinguished Alumnus Award from the Indian Institute of Technology, Kharagpur (2014).

Prof. Chakrabartys current research projects include: testing and design-for-testability of integrated circuits; digital microfluidics, biochips, security, and cyberphysical systems; optimization of enterprise systems and smart manufacturing. He is a Fellow of ACM, a Fellow of IEEE, and a Golden Core Member of the IEEE Computer Society. Prof. Chakrabarty served as the Editor-in-Chief of *IEEE Design & Test of Computers* during 2010-2012 and *ACM Journal on Emerging Technologies in Computing Systems* during 2010-2015. Currently he serves as the Editor-in-Chief of *IEEE Transactions on VLSI Systems*. He is also an Associate Editor of *IEEE Transactions on Computers*, *IEEE Transactions on Biomedical Circuits and Systems*, *IEEE Transactions on Multiscale Computing Systems*, and *ACM Transactions on Design Automation of Electronic Systems*.

Bhargab B. Bhattacharya (F'07) is currently professor of computer science and engineering and INAE Chair Professor at the Indian Statistical Institute, Kolkata, India. He received the B.Sc. degree in physics from the Presidency College in 1971, the B.Tech. and M.Tech. degrees in radiophysics and electronics in 1974 and 1976, respectively, and the Ph.D. degree in computer science in 1986, all from the University of Calcutta. He held visiting professorship at the University of Nebraska-Lincoln, and at Duke University, USA, at the University of Potsdam, Germany, at the Kyushu Institute of Technology, Iizuka, Japan, at Tsinghua University, Beijing, China, at IIT Kharagpur and IIT Guwahati, India. His research interest includes design and testing of integrated circuits, nano-biochips, digital geometry, and image processing architecture. He has published more than 300 technical articles, and he holds 10 United States Patents. He is a Fellow of the Indian National Academy of Engineering and a Fellow of the National Academy of Sciences (India). He is currently on the editorial board of the *Journal of Electronic Testing: Theory and Applications* (JETTA).