Efficient and Adaptive Error Recovery in a Micro-Electrode-Dot-Array Digital Microfluidic Biochip

Zipeng Li, Kelvin Yi-Tse Lai, John McCrone, Po-Hsien Yu, Krishnendu Chakrabarty, *Fellow, IEEE*, Miroslav Pajic, *Member, IEEE*, Tsung-Yi Ho, *Senior Member, IEEE*, and Chen-Yi Lee

***Abstract*—A digital microfluidic biochip (DMFB) is an attractive technology platform for automating laboratory procedures in biochemistry. In recent years, DMFBs based on a micro-electrode-dot-array (MEDA) architecture have been proposed. MEDA biochips can provide advantages of better capability of droplet manipulation and real-time sensing ability. However, errors are likely to occur due to defects, chip degradation, and the lack of precision inherent in biochemical experiments. Therefore, an efficient error-recovery strategy is essential to ensure the correctness of assays executed on MEDA biochips. By exploiting MEDA-specific advances in droplet sensing, we present a novel error-recovery technique to dynamically reconfigure the biochip using real-time data provided by on-chip sensors. Local recovery strategies based on probabilistic-timed-automata are presented for various types of errors. An on-line synthesis technique and a control flow are also proposed to connect local-recovery procedures with global error recovery for the complete bioassay. Moreover, an integer linear programming (ILP)-based method is also proposed to select the optimal local-recovery time for each operation. Laboratory experiments using a fabricated MEDA chip are used to characterize the outcomes of key droplet operations. The PRISM model checker and three benchmarks are used for an extensive set of simulations. Our results highlight the effectiveness of the proposed error-recovery strategy.**

***Index Terms*—Micro-electrode-dot-array, digital microfluidics, optimization, error recovery, on-line synthesis.**

I. Introduction

OVER the past decade, microfluidic biochips, also referred to as lab-on-a-chip, have been used for various biochemical applications, such as high-throughput DNA sequencing, point-of-care clinical diagnostics, and protein crystallization for drug discovery [1]. While early generations of microfluidic biochips used continuous fluid flow through permanently etched microchannels, more recent biochips, referred to as digital microfluidic biochips (DMFBs), manipulate liquids as discrete droplets of nanoliter and picoliter volumes based on the principle of electrowetting-on-dielectric (EWOD). Compared with the conventional continuous-flow biochip, a DMFB offers the advantages of simple instrumentation, flexible device geometry, reconfigurability and easy coupling with other technologies. To achieve the full potential of scalability and reconfigurability in digital microfluidics, a novel micro-electrode-dot-array (MEDA) architecture [2]–[4] has been proposed recently.

Unlike conventional digital microfluidics, where electrodes of equal size are arranged in a regular pattern, the MEDA architecture is based on the concept of a sea-of-micro-electrodes with an array of identical basic microfluidic unit components called *microelectrode cells* (MCs). Each MC consists of a microelectrode and a control/sensing circuit. A high-voltage shielding layer is inserted between the microelectrode and the control/sensing circuit to ensure the correct operation of the MC. The MEDA architecture allows microelectrodes to be dynamically grouped to form a micro-component (e.g., mixer or diluter) that can perform different microfluidic operations on the chip. Prototypes of MEDA-based biochips have been fabricated using TSMC 0.35 μm CMOS technology [2], and these devices can use a power-supply voltage of only 3.3 V for embedded control circuits [3].

However, as in the case of integrated circuits, continued increase in the density and area of microfluidic biochips will also result in more defects and reduce yield [5]. Fault models can be used to represent the effect of physical defects at some level of abstraction. These models can be used to capture the effect of defects that result in incorrect behavior. Some possible causes of physical defects are as follows.

- *Dielectric breakdown*: High voltage during actuation causes dielectric breakdown, which can directly expose the droplet to high voltage. In this case, droplet electrolysis happens and the droplet cannot be controlled.
- *Damage to the hydrophobic layer*: The hydrophobic layer can be damaged by chemical reaction or physical scratch. A damaged hydrophobic layer cannot provide sufficient electrowetting force when electrodes are actuated, and then reduced force impedes droplet transportation.
- *Short-circuited miroelectrodes*: A short between two ad-

The work of K. Y.-T. Lai, P.-H. Yu and C.-Y. Lee was supported in part by the Taiwan Ministry of Science and Technology under Grant MOST 103-2221-E-009-191 and 104-2218-E-009-007. The work of T.-Y. Ho was supported in part by the Ministry of Science and Technology of Taiwan, under Grant MOST 105-2221-E-007-118-MY3 and 104-2220-E-007-021, and in part by the Technische Universität München-Institute for Advanced Study, funded by the German Excellence Initiative and the European Union Seventh Framework Programme under grant agreement n^{o} 291763. A preliminary version of this paper appeared in the Proceedings of IEEE/ACM International Conference on Computer Aided Design (ICCAD), 2016.

Z. Li, K. Chakrabarty and M. Pajic are with the Department of Electrical and Computer Engineering, Duke University, Durham, NC, USA (email: zipeng.li@duke.edu; krish@duke.edu; miroslav.pajic@duke.edu).

J. McCrone is with IBM Corporation, Research Triangle Park, NC, USA (email: jvmccron@us.ibm.com).

K. Y.-T. Lai, P.-H. Yu and C.-Y. Lee are with the Department of Electronics Engineering, National Chiao Tung University, Hsinchu, Taiwan (email: kelvin@si2lab.org; yph3135134@hotmail.com; cylee@si2lab.org).

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

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.2017.2729347, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

jacent microelectrodes leads to a "larger" electrode. Once a droplet resides on this larger electrode, it is not able to create the desired variation of interfacial surface tension along the droplet transportation path.

- *Parasitic transistor leakage*: Parasitic transistor leakage in the control circuit can result in a degradation of the high voltage level on the bottom microelectrode. Accordingly, there is not enough voltage difference between the top and the bottom electrode, which fails the droplet actuation.

In addition to defects and imperfections for fabricated MEDA-based biochips, faults may also arise during bioassay execution. For example, excessive actuation voltage may lead to electrode breakdown and charge trapping [6], and DNA fouling may lead to the malfunction of multiple electrodes in the biochip [7]. Faults in biochips may eventually result in errors (e.g., a splitting operation with unbalanced droplets), which can adversely impact the correctness of the entire experiment. However, many biomedical applications (e.g., clinical diagnostics) require high precision for each operation. Therefore, efficient error-recovery strategies are required to ensure robust fluidic operations and high confidence in the outcome of biochemical experiments.

Several error-recovery strategies for digital microfluidics have recently been proposed in [8]–[11]. However, due to the inherent differences between traditional DMFBs and MEDA, existing error-recovery solutions cannot exploit the advantages specific to MEDA-based biochips. For example, in conventional DMFBs, droplets need to be transported to a nearby checkpoint for error detection. However, in MEDA-based biochips, droplets can be detected anywhere on the chip and the response time for sensing is only 10 ms [2]. Moreover, previous strategies use a unified recovery procedure (e.g., roll-back [10], [11]) for all types of errors; this approach can make error recovery inefficient and result in longer recovery time for specific types of errors. Furthermore, prior methods assume that error recovery is always feasible and they neglect the likehood that error-recovery procedures may also fail.

To overcome the above drawbacks, we propose a new probabilistic-timed-automata (PTA)-based strategy for error recovery in MEDA-based biochips. The proposed error-recovery strategy aims to fully exploit the advanced sensing techniques offered by MEDA. The key contributions of this paper are listed as follows:

1) We propose a classification of the outcomes of operations into three categories: *no error*, *minor error*, and *major error*. Each outcome is treated in a different way in the proposed error-recovery strategy.
2) We carry out laboratory experiments using a fabricated MEDA biochip to estimate these outcome probabilities.
3) Instead of utilizing the same error-recovery strategy for all types of errors (e.g., roll-back [10], [11]), we propose different PTA-based error-recovery strategies for different types of local errors. Discrete-time analysis and model checking are also used to compute the probability of success under constraints on the error-recovery time.
4) An on-line synthesis approach is proposed for MEDA biochips. A control flow is also proposed to connect the local recovery procedures to global error recovery for the complete bioassay. We also examine the influence of local recovery procedures on the execution of the bioassay.
5) We propose an ILP-based method for determining the local recovery time for each operation in the bioassay. The method can effectively reduce the time cost with a predefined probability of success for any given bioassay.
6) Simulation results for three benchmarks are derived using the PRISM model checker and compared with the results from previous methods. Our results illustrate the effectiveness of the proposed error-recovery strategy.

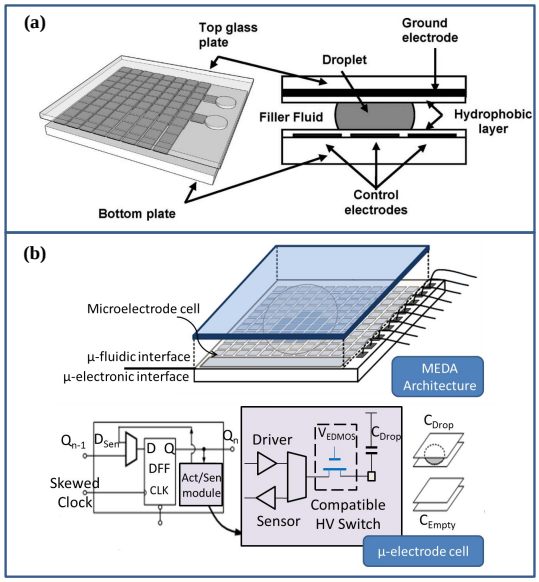


Fig. 1. (a) Illustration of a conventional DMFB [12]. (b) Illustration of the MEDA architecture and microelectrode cell (μ-electrode cell).

The remainder of the paper is organized as follows. Section II presents background material on digital microfluidics and the MEDA architecture. Section III presents the PTA-based error-recovery strategy in detail. Section IV presents the method to select the optimal time limit for local error recovery. Section V describes the proposed method for global error recovery. Section VI presents experimental results. Finally, Section VII concludes the paper.

II. DIGITAL MICROFLUIDICS AND MEDA

A DMFB utilizes electrowetting-on-dielectric (EWOD) to manipulate and move nanoliter or picoliter droplets containing biological samples on a two-dimensional electrode array. MEDA extends this basic architecture by adding more flexibility, as described in Section I. A comparison between a conventional DMFB and a MEDA-based biochip is presented in Fig. 1. A typical MEDA-based biochip consists of two plates: a top plate, which serves as a reference electrode, and a bottom plate with patterned microelectrodes.

The size of the microelectrodes can be 10 times smaller (e.g., 100 μm [13] in length) than conventional electrodes. Each microelectrode cell (MC) consists of a microelectrode, an activation circuit, and a sensing circuit. MEDA-based DMFBs allow the dynamic grouping of microelectrodes to form different shapes and fluidic modules.

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.2017.2729347, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

A dielectric layer and a hydrophobic layer are deposited above the microelectrodes, and droplets are sandwiched between the two plates. Once the electrodes are actuated, the EWOD force induces droplet motion. In this way, fluidic operations (e.g., cutting and mixing) on droplets with different sizes can be achieved on the chip.

Different with conventional DMFBs, MEDA biochips can provide a real-time sensing mechanism to detect the property (droplet-property sensing) and the location (droplet-location sensing) of the droplet. Droplet sensing results are presented in the sensing map. The sensing technique on MEDA biochips can provide chip-users with detailed information about the outcomes of on-chip operations. Therefore, any errors can be detected in a real-time manner and the error-recovery technique can be applied to correct these errors.

III. Error Recovery For Local Errors

In this section, we first describe the errors that are being targeted in this paper. Instead of regarding the detection outcome for each operation as either "success" or "failure", we classify the outcome into three categories: "no error", "minor error" and "major error". A probabilistic timed automata (PTA)-based error recovery approach is proposed for each type of error. PTA is a formalism for modelling systems whose behaviour incorporates both probabilistic and real-time characteristics [14]. PTAs are similar to finite-state machines (FSMs) that use clocks to capture time, which can be used to specify guard conditions and invariants as well as probabilistic edge transitions. The probability of success (POS) for error recovery is also considered. The outcome classification and the PTA-based approach distinguish this work from previous error-recovery strategies for conventional DMFBs.

A. Target Errors in MEDA

A DMFB is said to have an error if its operation does not match its specified behavior. Errors are typically caused by physical defects in DMFBs. Fault models can be used to represent the effect of physical defects at some level of abstraction. As described in [9], [15], [16], faults in DMFBs can be classified as being either catastrophic or parametric. Catastrophic faults lead to a complete malfunction of the system, while parametric faults cause degradation in the system performance. Physical defects that cause parametric faults include geometrical parameter deviations, which includes deviation in insulator thickness, electrode length and the height between parallel plates [15]. In this paper, we assume that chips have been carefully tested using both functional test methods [16] and structural test methods [15] before they are used for bioassay execution. For example, we assume that droplets will never be stuck during their transportation and droplet dispensing can always be successfully achieved.

However, some manufacturing defects may be latent, and they may produce errors during field operation. Moreover, harsh operational environments and biological samples (e.g., protein) may result in particle contamination and residue on surfaces due to adsorption, which may also result in unexpected errors [17]. These are referred to as *on-line errors*, and they occur after a series of fluidic operations [18]. Such on-line errors can have serious consequences on bioassay results. Therefore, to ensure robust execution of the target bioassay, we propose efficient error-recovery approaches for these on-line errors. More specifically, we target on-line errors related to mixing, splitting, mixing, and dilution.

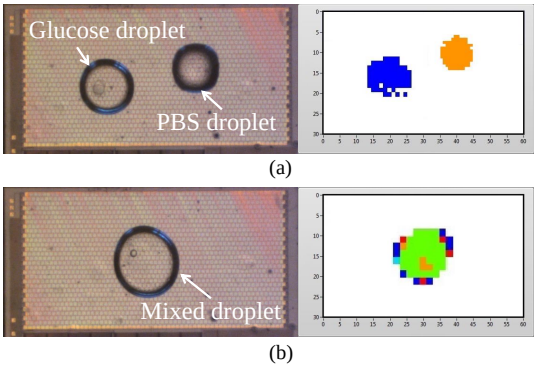


Fig. 2. Illustration of (a) two droplets before mixing and the corresponding droplet-property sensing map, and (b) the mixed droplet and the corresponding droplet-property sensing map.

B. Outcome Classification of Fluidic Operations

Outcomes of error detection can either be "success" or "failure" in a conventional DMFB [8], [11]. However, once an error is detected, there is no way to determine the extent of the error. For example, we can detect that a droplet splitting operation produces two droplets with unbalanced volumes, but we are not able to easily determine the extent of volume imbalance.

However, MEDA provides a practical sensing technique for droplets on the chip, referred to as real-time droplet size sensing [19]. This sensing technique can provide us with detailed information about the outcomes of error detection. Accordingly, we are able to classify the outcomes of error detection into multiple categories, and the most-efficient recovery procedure can be utilized for each type of error.

1) Outcome Classification for Mixing: In addition to droplet-location sensing, droplet-property sensing can also be achieved on MEDA-based biochips [19]. Droplet-property sensing can be used to distinguish between different kinds of droplets based on their permittivities. Since there is a sensing circuit under each microelectrode, droplet-property sensing can be achieved anywhere on the chip in a real-time manner.

A software package can be used to map different permittivity levels to various colors for ease of visualization [19]. An example is shown in Fig. 2(a): the glucose droplet is visualized in the measurement window using blue color while the phosphate-buffered saline (PBS) droplet is orange. Based on the detected permittivity levels, we are able to determine whether two droplets are uniformly mixed. If two droplets are uniformly mixed, there will be only one detected permittivity level. Accordingly, there is only one color, defined as *final color*, in the visualization of the mixed droplet in the measurement window. Otherwise, there are multiple colors associated with the visualization of the mixed droplet; see Fig. 2(b). We next quantify the error factor (F_{mix}) for mixing operation:

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.2017.2729347, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

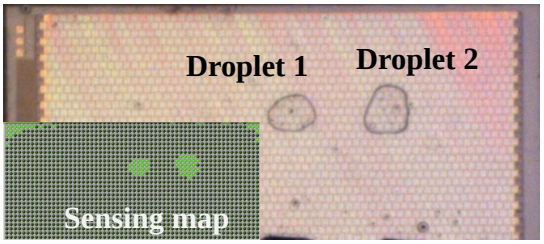


Fig. 3. Illustration of two split droplets on a MEDA-based chip and the corresponding droplet-sensing map.

$$F_{mix} = \frac{N_P}{N_T} \tag{1}$$

where N_P denotes the number of microelectrodes visualized by the *final color* in the measurement window, and N_T represents the total number of microelectrodes occupied by the droplet.

After we calculate F_{mix}, two user-defined thresholds, T_{m1} and T_{m2} ($T_{m1} < T_{m2}$), can be used to distinguish between different outcomes. The outcome of a mixing operation is characterized as (i) *major error* if $F_{mix} \leq T_{m1}$, (ii) *minor error* if $T_{m1} < F_{mix} \leq T_{m2}$, or (iii) *no error* if $F_{mix} > T_{m2}$.

2) Outcome Classification for Splitting: The outcome classification for splitting is based on the size difference between two split droplets. The computation of the error factor for the splitting operation (F_{split}) is described by (2), where N_1 is the number of microelectrodes occupied by one split droplet, and N_2 is the number of microelectrodes occupied by the other split droplet.

$$F_{split} = 1 - \frac{|N_1 - N_2|}{\max\{N_1, N_2\}} \tag{2}$$

An example is shown in Fig. 3. After splitting, Droplet 1 occupies 17 microelectrodes ($N_1 = 17$) and Droplet 2 occupies 25 microelectrodes ($N_2 = 25$). Therefore, the error factor for the splitting operation in Fig. 3 is calculated to be $1 - \frac{|17-25|}{\max\{17,25\}} = 0.68$.

Similar to mixing operations, two user-defined thresholds, T_{s1} and T_{s2} ($T_{s1} < T_{s2}$), can be used to distinguish between different outcomes of splitting operations. The outcome of a splitting operation is characterized as (i) *major error* if $F_{split} \leq T_{s1}$, (ii) *minor error* if $T_{s1} < F_{split} \leq T_{s2}$, or (iii) *no error* if $F_{split} > T_{s2}$.

3) Outcome Classification for Dilution: A dilution operation can be regarded as a mixing operation followed by a splitting operation. Therefore, the outcome classification for any dilution operation is determined by the outcome classifications of the corresponding mixing and splitting operations. The outcome of the dilution operation can be characterized as *no error* if and only if outcomes of both the mixing and splitting operations are characterized as *no error*. If the outcome of either mixing or splitting is characterized as a *major error*, the outcome of the dilution operation is characterized as a *major error*. In other cases, the outcome of the dilution operation is characterized as a *minor error*. The outcome classification for dilution operations is shown in Table I.

TABLE I
OUTCOME CLASSIFICATION OF DILUTION OPERATIONS.

Outcome (Dilution)	Outcome (Mixing)	Outcome (Splitting)
No error	No error	No error
Minor error	No error	Minor error
Minor error	Minor error	No error
Minor error	Minor error	Minor error
Major error	No error	Major error
Major error	Minor error	Major error
Major error	Major error	No error
Major error	Major error	Minor error
Major error	Major error	Major error

C. Experimentally Characterized Outcome Probability

Note that T_{m1}, T_{m2}, T_{s1}, and T_{s2} are user-defined parameters. Higher values of these parameters indicate higher precision requirements on the bioassay outcomes.

The outcome probability of an operation is defined as the probability that the outcome of the operation belongs to that particular category. We estimated the outcome probabilities for mixing and splitting using experiments on a fabricated MEDA-based biochip. The micro-photo of the fabricated chip and the experimental setup are shown in Fig. 4. The splitting operation was performed using a 7×7 deionized (DI) water droplet. The mixing operation was performed using a 10×10 PBS droplet and a 10×10 glucose droplet.

We repeated experiments involving the mixing and splitting operations 100 times. Each mixing (splitting) operation takes two seconds (one second) to complete. Based on the classification method described in Section III-B, the outcome of each operation is placed in one of the three categories: *no error*, *minor error*, and *major error*.

The outcome probability of mixing/splitting/dilution operations can be calculated based on the experimental results. In this paper, the selection of threshold values (T_{m1}, T_{m2}, T_{s1}, and T_{s2}) is based on the experimental results. Fig. 5 presents the distribution of the error factor F_{mix} and F_{split} for mixing and splitting operations. As shown in the figure, there are three clusters corresponding to the three outcomes: *major error*, *minor error*, and *no error*. In Fig. 5(a), 0.70 (0.90) can separate cluster 1 and cluster 2 (cluster 2 and cluster 3). Accordingly, we set 0.70 and 0.90 as T_{m1} and T_{m2}, respectively. Similarly, we set 0.50 and 0.80 as T_{s1} and T_{s2}, respectively.

Experimentally characterized outcome probabilities are shown in Table II. Note that the outcome probabilities in Table II are obtained using a fabricated MEDA biochip. These probabilities are used to provide some level of information. Note that different MEDA biochips may result in different

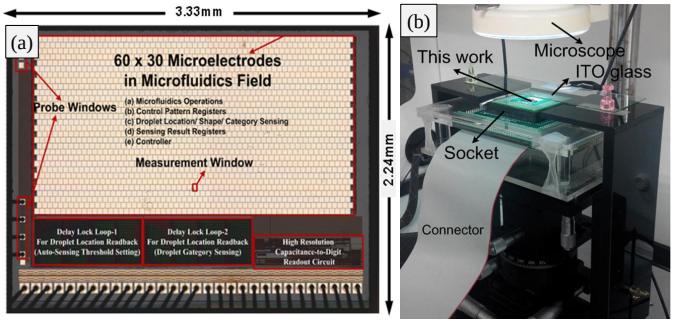


Fig. 4. (a) Chip micro-photo and (b) experimental setup.

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.2017.2729347, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

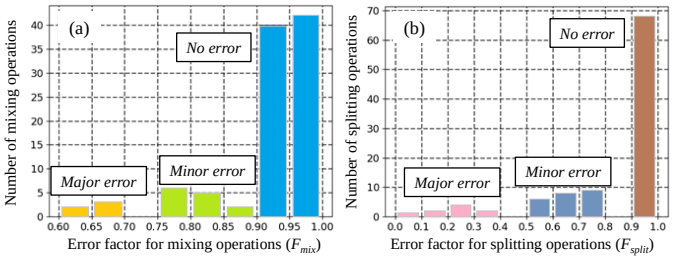


Fig. 5. Distribution of (a) F_{mix} and (b) F_{split} over the number of operations.

TABLE II
EXPERIMENTALLY CHARACTERIZED OUTCOME PROBABILITY.

Outcome	Probability		
	Mixing	Splitting	Dilution
No error	0.82	0.68	0.56
Minor error	0.13	0.23	0.30
Major error	0.05	0.09	0.14

probability values. If accurate probabilities are required for a MEDA biochip, corresponding experiments are needed to be executed on the same chip.

We observed that the outcome probability of one operation does not depend on the outcome of the previous operation. Therefore, the outcome probabilities in Table II are static. However, the outcome probabilities may be different for different types of droplets. More studies are needed to explore the relationship between the outcome probability and the type of droplet. For the sake of simplicity and due to then lack of significant experimental results to the contrary, we assume here that the outcome probability is the same for all types of droplets. According to Table II, if we detect a mixing error, the probability that the error is a minor error or a major error is calculated to be $0.13/(0.13+0.05) \approx 0.77$ or $0.05/(0.13+0.05) \approx 0.23$, respectively. Similarly, the probability that the splitting error is a minor or a major error is calculated to be 0.72 and 0.28, respectively.

D. Local Recovery Approaches

In this subsection, we first present a formal method to check for the existence of backup droplets for any operation in the given bioassay. We then present the proposed PTA-based methods for local error recovery. The method for backup-droplet checking is used in the PTA-based methods for recovery procedures for mixing errors and dilution errors.

1) Backup-Droplet Checking Method: Here we first give the definition of backup droplets. For a splitting or a dilution operation, if only one of its output droplets is used as the input for the immediate successors, the other (redundant) droplet is a backup droplet for possible error recovery. Moreover, dispensing operations can be scheduled for execution as early as possible and some extra droplets can be stored on the biochip as backup. Those unused backup droplets are sent back to the waste reservoir when the bioassay is completed.

In a given sequencing graph, each node represents an operation. We define the number of input droplets as the in-degree $IN(O_i)$ for operation O_i. Similarly, the out-degree $OUT(O_i)$ is defined as the number of output droplets. Both $IN(O_i)$ and $OUT(O_i)$ can be calculated from the sequencing graph. Note that droplets with different sizes can exist on MEDA biochips, $IN(O_i)$ and $OUT(O_i)$ are counted in the number of unit droplets. Based on $IN(O_i)$ and $OUT(O_i)$, operation O_i can be divided into three categories.

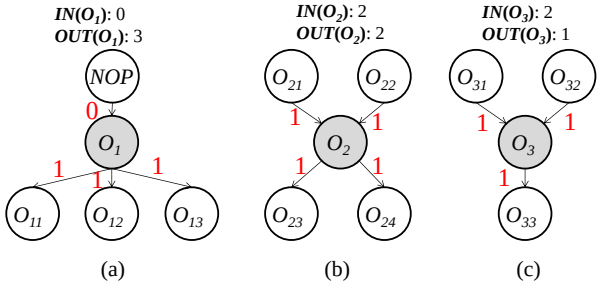


Fig. 6. Examples of operations in different categories in (a) Category I, (b) Category II, and (c) Category III. Operation NOP is a pseudo-source operation. Red numbers represent the number of unit droplets from operations to their successor operations.

Category I: $IN(O_i)$ is less than $OUT(O_i)$. If O_i is in Category I, the operation O_i is a dispensing operation, and $IN(O_i)$ is zero. Therefore, there are always backup droplets for any operation in Category I. An example is shown in Fig. 6(a). The dispensing operation O_1 is with an in-degree of zero and an out-degree of three. Dispensed droplets are used for operations O_{11}, O_{12}, and O_{13}.

Category II: $IN(O_i)$ is equal to $OUT(O_i)$. In this category, all droplets generated from the operation O_i is used for subsequent operations. Accordingly, there is no backup droplets for the operation O_i. An example is shown in Fig. 6(b). The dilution operation O_2 is with an in-degree of two and an out-degree of two. The two diluted droplets are used for subsequent operations O_{22} and O_{23}.

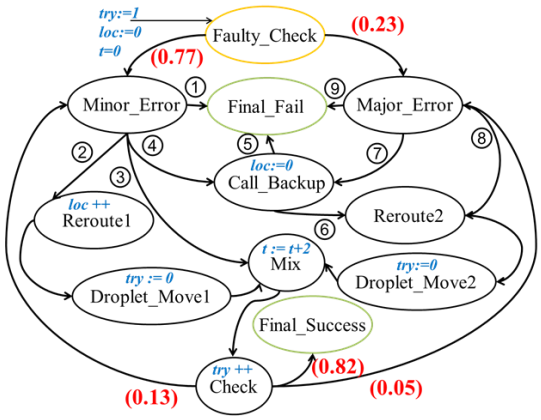
Category III: $IN(O_i)$ is larger than $OUT(O_i)$, which means not all droplets generated from O_i will be used for subsequent operations. An example is shown in Fig. 6(c). Operation O_3 is a dilution operation. However, only one diluted droplet is used for operation O_{33}. Therefore, the other diluted droplet from O_3 can be used as the backup droplet for operation O_3.

Operations in Category I and Category III are with backup droplets. For an operation O_k, if the numbers of backup droplets for O_k's immediate predecessors are all nonzero, O_k can be re-executed using backup droplets.

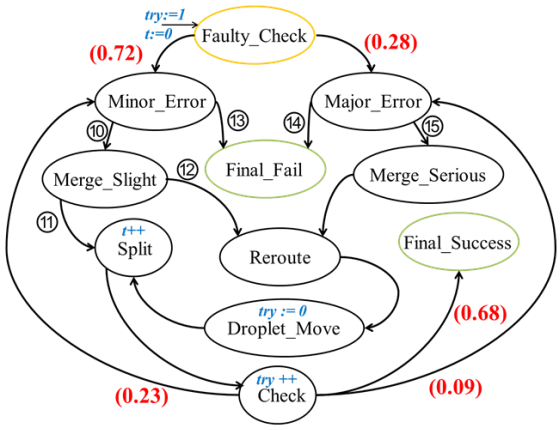
2) PTA-Based Methods for Local Recovery: We have developed PTA-based methods for local recovery in the case of mixing, splitting, and dilution errors, respectively. The state-transition diagram of the PTA for mixing errors (PTA_m) is shown in Fig. 7(a). There are two thresholds in PTA_m: the time threshold t_{th} and the location threshold l_{th}. Time threshold t_{th} limits the maximum time for local recovery and location threshold l_{th} limits the largest number of mixers that can be used for error recovery. The parameters t, try, and loc are used to record the time cost, the number of recovery operations on one mixer, and the number of mixers that have been used for local recovery, respectively.

In PTA_m, we first carry out error detection and then classify the corresponding detection outcome; an error-recovery approach is then selected based on the corresponding classification. For example, a mixing operation with a minor error can be simply recovered by redoing the mixing on the current mixer, while a mixing operation with a major error has to be

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.2017.2729347, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems



(a) PTA_m (PTA for mixing errors)



(b) PTA_s (PTA for splitting errors)

State transition conditions:

① $t \geq t_{th}$
② $t < t_{th}$ & $loc < l_{th}$ & $try > 1$
③ $t < t_{th}$ & $loc < l_{th}$ & $try = 1$
④ $t < t_{th}$ & $loc \geq l_{th}$ & $try > 1$
⑤ $backup = 0$
⑥ $backup = 1$
⑦ $t < t_{th}$ & $loc \geq l_{th}$
⑧ $t < t_{th}$ & $loc < l_{th}$
⑨ $t \geq t_{th}$
⑩ $t < t_{th}$
⑪ $try = 1$ & $t < t_{th}$
⑫ $try > 1$ & $t < t_{th}$
⑬ $t \geq t_{th}$
⑭ $t \geq t_{th}$
⑮ $t < t_{th}$

Fig. 7. Illustration of PTAs for recovery from (a) mixing errors, and (b) splitting errors. Orange (green) outline indicates that the corresponding state is a start (end) state in the state transition diagram.

routed to a nearby mixer to redo the mixing operation.

When a mixing error is detected, PTA_m enters the Faulty_Check state. Based on the classification results, the next state can either be Minor_Error or Major_Error; the corresponding transition probabilities are calculated using the experimental results discussed in Section III-C and highlighted in Fig. 7(a). If the PTA_m moves to Minor_Error, the next state can be Reroute1, Mix, Call_Backup, or Final_Fail. In the state of Reroute1, the droplet will be moved to a nearby available mixer to redo the mixing. In the Mix state, the droplet will be mixed one more time on the same mixer; In Call_Back state, PTA_m will search for backup droplets to recover from the mixing error (here we use the method in Section III-D to find backup droplets); In Final_Fail, because the time-cost exceeds t_{th}, the local recovery is deemed to have failed. Similarly, if PTA_m moves to the Major_Error state, the next state can be Reroute1, Call_Backup, or Final_Fail.

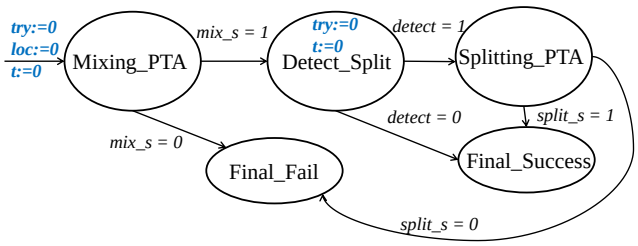


Fig. 8. Illustration of PTA_d (PTA for error recovery of dilution errors).

Whenever PTA_m transitions to Mix, the next state is always Check. Based on the results of error detection, PTA_m can move to Minor_Error, Major_Error or Final_Success. The corresponding transition probabilities are also highlighted in Fig. 7(a). The Final_Success state indicates that local recovery has been successful, while the Minor_Error and Major_Error states will continue to lead to other states until PTA_m moves either to Final_Success or Final_Fail. In PTA_m, the time cost associated with the Mix state is 2 s; other states are not associated with any time cost.

The state-transition diagram of the PTA for splitting errors (PTA_s) is shown in Fig. 7(b). This PTA is similar to PTA_m: time threshold t_{th} is used to limit the time cost for error recovery; parameters t and try are used to record the error-recovery time and the number of operations on one splitter, respectively. In contrast to PTA_m, two states, Merge_Minor and Merge_Major, are added to PTA_s. If the outcome of a splitting operation is classified as *minor error* (*major error*), the split droplets will be merged for the next splitting step in the Merge_Minor (Merge_Major) state. Another difference between PTA_s and PTA_m is that PTA_s does not have the Call_Backup state. This is because the new droplet for splitting can be simply generated by merging two split droplets, therefore there is no need for a backup droplet. In PTA_s, the time cost associated with the Split state is 1 s; other states are not associated with any time cost.

The PTA for dilution errors (PTA_d) is shown in Fig. 8. Since a dilution operation can be regarded as a mixing operation followed by a splitting, here we divide the dilution into two stages: mixing stage and splitting stage. Accordingly, the proposed PTA_d is a combination of PTA_m and PTA_s (see Fig. 8). Parameters try, loc, and t in PTA_d have the same meaning as the corresponding parameters in PTA_m and PTA_s. If an error is detected in the mixing stage, PTA_d enters the Mixing_PTA state. The parameter mix_s ($split_s$) is used to indicate whether recovery has been made from the mixing (splitting) error. The parameter $detect$ is used to indicate whether a splitting error is detected. If $mix_s = 1$, PTA_d moves to the Detect_Split state; otherwise, PTA_d transitions to the Final_Fail state. In the Detect_Split state, parameters try and t are reset to 0. If an error is detected in the splitting stage, then PTA_d moves to Splitting_PTA; otherwise, PTA_d transitions to the Final_Fail state. Note that PTA_d will finally transition to either the Final_Fail or Final_Success state based on whether recovery can be successfully completed.

In the proposed PTAs for local recovery, i.e., PTA_s, PTA_m, and PTA_d, a transition to the next state depends only on the current state; this is referred to as the Markov property [20]. Therefore, we map these PTAs to discrete-time

Markov chains (DTMCs). Here, we assume that the computation time for droplet routing and droplet-transportation time are negligible compared to microfluidic operation times in local recovery procedures [21]. However, the transition times from the initial state to the Final_Success state in PTA_m, PTA_s, or PTA_d may be different. Therefore, we divide the Final_Success state into different sub-states, referred to as Final_Success sub-states, in each of the above DTMCs. For example, it can take 2, 4, 6, 8, or 10 s to transition from the Faulty_Check (initial state) to the Final_Success state in PTA_m. We divide the Final_Success state into the following sub-states: (i) Final_Success_2, (ii) Final_Success_4, (iii) Final_Success_6, (iv) Final_Success_8, and (v) Final_Success_10 in the DTMC corresponding to PTA_m. The utilization of Final_Success sub-states eliminates the need for cumbersome time calculations, hence we can focus exclusively on the state-occupancy probability calculations.

We use the probabilistic model checker, PRISM version 4.0.3 [22], to verify and analyze the DTMCs. We first develop a model for each DTMC using PRISM and check whether local recovery will eventually enter either the Final_Fail state or the Final_Success sub-states. We then apply discrete-time simulation to calculate the probabilities that the DTMC can eventually transition to different Final_Success states. The summation of these probabilities is the probability of successful recovery.

IV. Time-Limit Selection for Local Recovery

PTAs for local recovery proposed in Section III-D are constrained by time limits on local recovery (TLLRs) because TLLRs can influence state transitions in the PTAs. Therefore, different TLLRs will result in different probabilities of success (POS) for local recoveries. The POSs for local recoveries can directly affect the POS for the complete bioassay. The method in [5] utilize a unified TLLR for all local recoveries, which is not necessary. Therefore, here we consider the optimization problem of how to determine the optimum TLLR for each operation under a pre-specified constraint on the given POS for the complete bioassay. We use integer linear programming (ILP) to solve the problem.

A. ILP Formulation

To create an ILP model for this problem, we need to define the set of variables and constraints. We first define tp_i as the type of operation O_i. Two values, $P_s(tp_i)$ and $P_f(tp_i)$, are defined as the probability of success and the probability of failure for O_i with the type tp_i. Note that $P_s(tp_i)$ and $P_f(tp_i)$ can be obtained from the experimental results described in Section III-C. For example, if operation O_1 is a mixing operation, $P_s(tp_1)$ and $P_f(tp_1)$ are 0.82 and 0.18, respectively. We then define $P_{ri}(T_i)$ as the probability that we can recover from the error for operation O_i using the proposed local recovery technique when TLLR is T_i. Once T_i is determined, $P_{ri}(T_i)$ can be calculated based on the proposed PTAs for different local recoveries. The calculation is presented in Section VI-A. Based on the above definitions, the probability that the operation O_i can be successfully executed can be calculated as follows.

$$P_i = P_s(tp_i) + P_f(tp_i) \cdot P_{ri}(T_i) \tag{3}$$

In order to calculate $P_{ri}(T_i)$, we introduce a binary variable that is defined as follows: $d_{ij} = 1$ if $T_i = j$ s, and 0 otherwise. For example, if $T_1 = 3$ s for operation O_1, d_{13} is 1. Otherwise, d_{13} is 0. Variable S_{ij} is the probability that the error on operation O_i can be recovered if the TLLR is j. As discussed in Section III-D, we can utilize the PRISM model checker to calculate the probability S_{ij} once the type of operation O_i is determined. Those probabilities are presented in Table III. Finally, $P_{ri}(T_i)$ can be calculated using (4), where TR_{max} is the predefined upper bound on TLLR for all operations.

$$P_{ri}(T_i) = \sum_{j=1}^{TR_{\max}} (S_{ij} \cdot d_{ij}) \tag{4}$$

Similarly, the TLLR T_i for operation O_i can be calculated using (5).

$$T_i = \sum_{j=1}^{TR_{\max}} (j \cdot d_{ij}) \tag{5}$$

Using above defined variables, the objective function of the ILP model is shown in (6), where N is the number of operations in the given bioassay.

$$\text{Minimize} \quad \frac{1}{N} \sum_{i=1}^{N} T_i \tag{6}$$

As the POS of the complete bioassay should be no less than a pre-specified value P_{th}, the constraint for the ILP model can be expressed in (7).

$$\prod_{i=1}^{N} P_i \geq P_{th} \tag{7}$$

It should be noted that (7) is a non-linear constraint. We linearize this by introducing a new variable $Q_m = \prod_{i=1}^{m} P_i$. The constraint (7) can be transformed to (8).

$$\begin{aligned} &POS = Q_N \\ &Q_N = Q_{N-1} \cdot P_N \\ &Q_{N-1} = Q_{N-2} \cdot P_{N-1} \\ &... \\ &Q_2 = Q_1 \cdot P_2 = P_1 \cdot P_2 \end{aligned} \tag{8}$$

Variable Q_i in (8) can be calculated using (9).

$$\begin{aligned} Q_i &= Q_{i-1} \cdot P_i \\ &= Q_{i-1} \cdot \left(P_s(tp_i) + P_f(tp_i) \cdot \sum_{j=1}^{TR_{\max}} (S_{ij} \cdot d_{ij}) \right) \\ &= P_s(tp_i) \cdot Q_{i-1} + P_f(tp_i) \cdot \sum_{j=1}^{TR_{\max}} (S_{ij} \cdot d_{ij} \cdot Q_{i-1}) \end{aligned} \tag{9}$$

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.2017.2729347, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

Objective:

$$\text{Minimize } \frac{1}{N}\sum_{i=1}^{N} T_i$$

Subject to:

$$P_i = P_s(tp_i) + P_f(tp_i)\cdot\sum_{j=1}^{TR_{\max}}\left(S_{ij}\cdot d_{ij}\right) \quad \forall i$$

$$T_i = \sum_{j=1}^{TR_{\max}}\left(j\cdot d_{ij}\right) \quad \forall i$$

$$Q_i = P_s(tp_i)\cdot Q_{i-1} + P_f(tp_i)\cdot\sum_{j=1}^{TR_{\max}} G_{ij} \quad \forall i>1$$

$$G_{ij} = S_{ij}\cdot F_{ij} \quad \forall j, \forall i>1$$

$$F_{ij} \geq 0 \quad \forall j, \forall i>1$$

$$F_{ij} \leq d_{ij} \quad \forall j, \forall i>1$$

$$Q_{i-1} - F_{ij} \geq 0 \quad \forall j, \forall i>1$$

$$Q_{i-1} - F_{ij} \leq 1 - d_{ij} \quad \forall j, \forall i>1$$

$$Q_1 = P_1$$

$$Q_N \geq P_{th}$$

Fig. 9. ILP model for the optimization of TLLR selection.

Note that the product of variables d_{ij} and Q_{i-1} is also nonlinear. We define another variable F_{ij} to represent the product of these two variables, and the variable F_{ij} can be linearized using standard linearization techniques as shown in Fig. 9. Variable G_{ij} is used to represent the product of S_{ij} and F_{ij}. Accordingly, (9) can be linearized as (10).

$$Q_i = P_s(tp_i)\cdot Q_{i-1} + P_f(tp_i)\cdot\sum_{j=1}^{TR_{\max}} G_{ij} \qquad (10)$$

The complete ILP model is shown in Fig. 9.

B. Randomized Rounding

While the ILP model presented in Section IV-A can be used to optimally solve TLLR selection problem, the ILP model does not scale well for large bioassays. Since LP problems can be solved optimally in polynomial time [23], we adopt the method of LP-relaxation, i.e., the binary variables are relaxed to real-valued variables. However, the fractional values obtained for the d_{ij} variables are inadmissible in practice; these variables must be mapped to either 0 or 1. Therefore, here we use randomized rounding to regulate the LP solution.

The randomized rounding technique for ILP problems consists of three steps. The first step is to solve the corresponding LP problem, fixing all d_{ij} variables than are assigned to 1. The second step is to randomly pick a variable from unfixed d_{ij} variables and assign it to 1 with a probability equal to the corresponding d_{ij} value. For example, the LP solution assigns 0.8 to a variable d_{ij}, we first generate a random number between 0 and 1. If the generated number is no large than 0.8, d_{ij} is assigned to 1. Otherwise, d_{ij} is assigned to 0. Note that during the randomized rounding, the constraints in the ILP model should not be violated in order to ensure the final solution is feasible. For example, if d_{ij} is supposed to be rounded to 1 using the randomized rounding, we first check whether $d_{ij} = 1$ can result in any constraint violation. If so, d_{ij} is forced to be assigned to 0. On the third step, the LP problem is solved again and the randomized rounding is repeated until all d_{ij} variables are set to either 0 or 1.

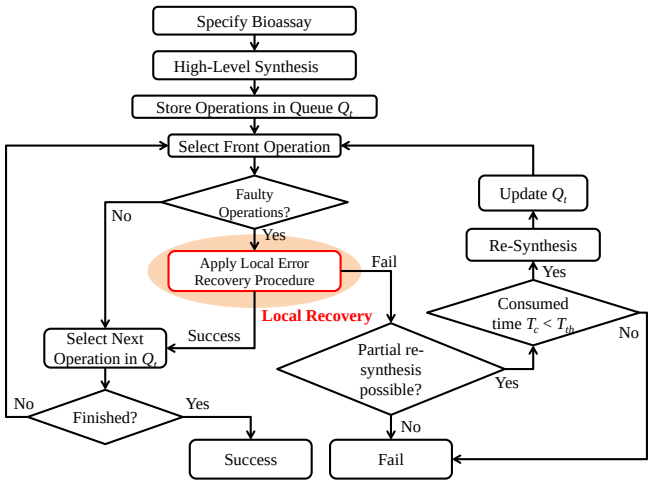


Fig. 10. Control flow for error recovery.

V. Error Recovery for the Complete Bioassay

In this section, we present the error recovery for the complete bioassay. A control flow is first proposed to connect the local recovery procedures with global error recovery for the complete bioassay. The on-line synthesis technique for MEDA biochips is then proposed.

A. Control Flow

The proposed control flow is shown in Fig. 10. When a bioassay is specified for execution on the biochip, high-level synthesis techniques are first utilized to derive the results of operation scheduling, module placement and droplet routing [21]. Then all operations are stored in the queue Q_t based on their assigned starting time in an ascending order. Operations are executed based on their order in Q_t. Once an operation is found to have an error, the local error-recovery approach is invoked. If the bioassay can recover from that error, local error-recovery returns global control to the controller and the next operation in Q_t is executed. If recovery cannot be made from the error, the controller checks whether partial re-synthesis (PRS) [11] is available. If PRS is available, new synthesis results are generated and Q_t is updated; otherwise, the error recovery fails and we need to execute the bioassay from the beginning.

For the complete bioassay, the probability of successful recovery and the completion time depend on the number of detected errors and the local recovery approaches. Let n denote the number of detected errors in the bioassay. Suppose the probability of successful local recovery for error i ($i = 1, 2, \ldots, n$) is P_i. Under the assumption that local error recoveries are mutually independent, the corresponding probability of successful recovery (P_b) can be computed as $P_b = \prod_{i=1}^{n} P_i$. An on-line synthesis technique is proposed in Section V-B to dynamically generate synthesis results when

errors are detected and to calculate the bioassay completion time.

B. On-Line Synthesis

As illustrated in Fig. 10, the recovery graph is generated and utilized to recover from the detected error when partial recovery synthesis is available. The sequencing graph is also updated to incorporate the generated recovery graph. On-line synthesis technique is used to generate results of operation scheduling, module placement, and droplet routing on-the-fly for the updated sequencing graph with a minimum impact on the time-to-response.

An off-line synthesis technique for MEDA biochips has been proposed in [4]. However, the method in [4] takes long computation time for large scale of sequencing graph, i.e., it takes 341.69 s to generate the synthesis results for protein-dilution bioassay [24]. In order to ensure efficient reconfiguration, we propose a rapid on-line synthesis technique. Moreover, compared with off-line synthesis, on-line synthesis can optimize the bioassay completion time because it can dynamically exploit the biochip configuration at the time of error occurrence. Details of the proposed on-line synthesis technique are described as follows.

1) Adaptive Scheduler: Since error-recovery response time is critical, we use the greedy list-scheduling algorithm with computational complexity $O(n)$ [21], where n is the number of operations in the sequencing graph.

Since droplet routing time is no longer negligible and it needs to be considered on MEDA biochips [4], the sequencing graph needs further update to reflect the influence of droplet routing time. Here we first set the upper bound of the droplet routing time, T_{up}, as the maximum droplet routing time in the original synthesis results. We then insert droplet-routing operation with execution time T_{up} between any two consecutive operations.

The list scheduler is invoked when final sequencing graph is generated. It receives the updated sequencing graph and generates operation scheduling results. If one droplet routing time T_r violates the pre-specified upper bound T_{up}, the sequencing graph is re-generated for unexecuted operations. At the same time, T_{up} is updated to be T_r. Adaptive scheduler can then be invoked again to generate the scheduling results.

2) Adaptive Placer: A forbidden set (FS)-based module placement method is proposed in [4]. The method first identifies the FS for each new module M_i (a set of locations where M_i cannot be placed) is always maintained). Then, all the potential sites where M_i can be placed are obtained and stored in the possible set (PS). A scanning approach is finally used to examine each possible placed region, compute the placement cost for each position (based on droplet routing time), and finally select the optimal location. The placement cost for a new module M_i is calculated using $Cost(M_i) = \max_{o_j \in parent(o_i)} \{RT_{ij}(M_i, M_j)\}$, where $RT_{ij}(M_i, M_j)$ is the droplet routing time between modules M_i and M_j, $parent(o_i)$ is the set of o_i's parent operations and M_j is the bound fluidic module for o_j. However, the scanning approach needs to examine each possible location site, which consumes large computation time.

```
1: Start: input start location S and end location E;
2: frontier = PriorityQueue()
3: frontier.put(S, 0)
4: came_from = { }
5: came_from[S] = None
6: while not frontier.empty( ) do
7:    current = frontier.get( )
8:    if current == T then
9:       output droplet route
10:      break
11:   end if
12:   for next in graph.neighbors(current) do
13:      if next not in came_from & not in obstacles then
14:         priority = Manhattan(T, next)
15:         frontier.put(next, priority)
16:         came_from[next] = current
17:      end if
18:   end for
19: end while
20: End: output derived droplet route.
```

Fig. 11. Pseudocode for the adaptive router.

The proposed adaptive placer utilizes the method in [4] to identify all possible placed sites for the new module M_i. In contrast to the scanning site-selection method, we avoid the use of a scanning method to select the optimal location for M_i. We formulate the routing cost from module M_i to module M_j as $((x_j - x_i)^2 + (y_j - y_i)^2) \times 1/V_{ij}$, where (x_i, y_i) and (x_j, y_j) are locations for M_i and M_j, respectively, and V_{ij} is the droplet velocity, which can be calculated using the velocity model proposed in [4]. Similarly, the placement cost for a new module M_i is calculated using $\sum_{i=1}^{n} ((x_n - x_i)^2 + (y_n - y_i)^2)/V_{in}$. The objective is to find an optimal location such that the placement cost for M_i is minimized.

Based on the placement cost function, the optimal x coordinate (x_{oi}) and y coordinate (y_{oi}) for M_i will be $\sum_{i=1}^{n} (x_i \times 1/v_{in}) / \sum_{i=1}^{n} 1/v_{in}$ and $\sum_{i=1}^{n} (y_i \times 1/v_{in}) / \sum_{i=1}^{n} 1/v_{in}$, respectively. However, x_{oi} and y_{oi} may not always be integers, and they need to be rounded to closest integers x'_{oi} and y'_{oi}. If the rounded location is in the possible set of M_i, it is selected as the final location; Otherwise, the closest location to the rounded location in the possible set is selected. Using the proposed site-selection method, the computation complexity of the placer is reduced from $O(nS)$ to $O(n)$, where n and S represent the number of operations and the size of the MEDA biochip, respectively.

3) Adaptive Router: Droplet routing problem has been proved be NP-hard [25]. Several heuristic droplet-routing algorithms, including modified Lee algorithm [26], integer linear programming (ILP)-based algorithm [27], and soukup-based algorithm [28], have been proposed for conventional DMFBs. An A*-based algorithm is proposed for MEDA biochips [29]. However, all these algorithms take significant amount of time

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.2017.2729347, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

TABLE III
PROBABILITY OF SUCCESS FOR DIFFERENT LOCAL STRATEGIES*.

Local Recovery	Probability of Success with Different Time Cost Constraint t									
	t = 1 s	t = 2 s	t = 3 s	t = 4 s	t = 5 s	t = 6 s	t = 7 s	t = 8 s	t = 9 s	t = 10 s
Mixing (without backup droplets)	N/A	0.820	0.820	0.968	0.968	0.994	0.994	0.997	0.997	0.998
Mixing (with backup droplets)	N/A	0.820	0.820	0.968	0.968	0.994	0.994	0.998	0.998	0.999
Splitting	0.680	0.880	0.966	0.985	0.992	0.992	0.992	0.992	0.992	0.992
Dilution (without backup droplets)	N/A	N/A	0.557	0.721	0.792	0.851	0.934	0.952	0.960	0.978
Dilution (with backup droplets)	N/A	N/A	0.557	0.721	0.792	0.851	0.934	0.952	0.960	0.978

∗N/A denotes the fact that the local recovery cannot be completed with the corresponding time constraint.

to determine the shortest droplet route. Therefore, they are not suitable for on-line synthesis. Instead, we use a greedy best first search (BFS) algorithm for the adaptive router. The pseudocode for the greedy BFS algorithm is shown in Fig. 11. Queue **frontier** is a priority queue, in which elements are sorted in ascending order of their priorities (line 2). The priorities are calculated based on the Manhattan distance to the target location T (line 14). The start location S is assigned with a priority of 0 (line 3). Function "came_from" record the droplet route (lines 4-5). While queue **frontier** is not empty, the first element $current$ with the lowest priority in **frontier** is selected (lines 6-7). If $current$ is the target location T, the droplet route is obtained and the process stops (lines 8-11). Since diagonal droplet movement can be achieved on MEDA biochips [4], [13], the neighbors of one location L_i include all locations vertically, horizontally, and diagonally adjacent to L_i. An example is shown in Fig. 11. The priorities of the neighbor locations of $current$ is calculated as the Manhattan distance between locations $current$ and T. The neighbors of $current$ are sorted and stored in **frontier** if they have not been recorded and they are not in the obstacle region (lines 12-18). Note that the obstacle region is defined as the region of executing fluidic modules.

The obtained droplet route needs to meet both (i) fluidic constraint and (ii) timing constraint. The timing constraint is introduced in Section V-B. Once the droplet routing time is larger than the pre-specified upper bound T_{up}, the sequencing graph is updated and the on-line synthesizer needs to regenerate the synthesis results. For multiple droplet routes that may intersect or overlap with each other, fluidic constraint rules is imposed to avoid undesirable behavior. Details of the fluidic constraint for MEDA biochips is described in [30]. If the adaptive router fails to route the droplet from the start location to the end location, the adaptive placer is invoked to regenerate the module-placement results to improve the droplet routability.

Note that even though the greedy BFS method may not obtain the exact shortest droplet route, the algorithm is very fast and is suitable for on-line synthesis.

VI. EXPERIMENTAL RESULTS

We first examine the relationship between the probability of success for error recovery and the error-recovery time for each type of local error. The evaluation is carried out for both scenarios—with the presence and absence of backup droplets. We then present simulation results for three real-life benchmarks to evaluate the proposed error-recovery strategy.

A. Results for Local Faults

Probability computation in PTAs can be carried using a recursive procedure [31]; therefore, as discussed in Section IV, we utilize the PRISM model checker to calculate the probability of success (POS) for each type of local error.

Some types of samples, e.g., fibronectin, are known to degrade within 10 s [32]. Therefore, the time threshold t_{th} is set to be 10, 5, and 10 s in the PTAs for mixing, splitting, and dilution errors, respectively. The location threshold l_{th} is set to be 2 in both PTAs for mixing errors and dilution errors. This is because if recovery cannot be made from the error after using two mixers (diluters), the error-recovery time has already exceeded the time threshold.

Lines 1-2 in Table III present the relationship between the POS (P_{mix}) and the time cost (T_{mix}) for mixing errors. We note that, as expected, the more the time spent on error recovery, the higher is the value of P_{mix}. When T_{mix} is larger than 8 s, P_{mix} for the scenario of "with backup droplets" is larger than the corresponding P_{mix} for the scenario of "without backup droplets". However, the difference is almost negligible; thus the backup droplets do not significantly influence P_{mix} for mixing errors.

The relationship between the POS (P_{split}) and the time cost (T_{split}) for splitting errors is shown in line 3 in Table III. As explained in Section V, no backup droplet is utilized by the PTA for splitting errors. Therefore, here we only consider the scenario of "without backup droplet". Note that P_{split} also increases with the increase in T_{split}.

Lines 4-5 in Table III present the relationship between the POS (P_{dilute}) and the time cost T_{dilute} for dilution errors. Similar to local recovery for mixing and splitting errors, P_{dilute} increases with the increase in T_{dilute}. Based on these simulation results, we conclude that P_{dilute} for the scenario of "with backup droplet" is larger than the corresponding P_{dilute} for the scenario "without backup droplet" when T_{dilute} is larger than 15 s. However, since we have set t_{th} to be 10 s for dilution errors, there is no difference between the values of P_{dilute} in lines 4-5 in Table III.

B. Results for Bioassays

We next present simulation results for three real-life benchmarks, namely PCR [33], *in-vitro* diagnostics [25], and protein dilution [24], to evaluate the proposed method. All simulations are carried out on an Intel Core i7 platform with a 2.67 GHz CPU and 8 GB of RAM. The experimentally characterized module library for MEDA is shown in Table IV. Without any loss of generality, we first set the size of one electrode in conventional DMFBs to be equal to a 4×4 microelectrode

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.2017.2729347, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

TABLE IV
EXPERIMENTALLY CHARACTERIZED MEDA MODULE LIBRARY FOR SYNTHESIS.

Operation	Resource	Time
Dispensing	Reservoir	3 s
Mixing	2×2-array mixer	2 s
Diluting	2×2-array diluter	3 s
Sensing	sensing circuit	0.01 s

array in MEDA-based biochips; therefore, a 2×2 array in Table IV actually represents an 8×8 microelectrode array. We then set the chip size to be 10×10.

The error-recovery capability of MEDA-based biochips can be evaluated on the basis of the bioassay completion time and the POS when errors are detected. Here we use P_{pcr}, $P_{invitro}$, and $P_{protein}$ to present the POS for bioassays of PCR, *in-vitro* diagnostics, and protein dilution, respectively. Likewise, T_{pcr}, $T_{invitro}$, and $T_{protein}$ refer to the completion time for bioassays of PCR, *in-vitro* diagnostics, and protein dilution, respectively.

In our simulation, we randomly inject up to four errors into each benchmark 30 times, and then calculate the corresponding completion time and POS for different time limits on local recovery (TLLR). Similar to t_{th} in Section III-D, TLLR is defined as the maximum time that is allowed for local recovery. Here we utilize a unified value of TLLR for all local errors. For example, if TLLR is set to 5 s, the time limits for all types of errors are also 5 s. The comparison between the unified TLLR selection and the unique TLLR selection proposed in Section IV is presented in Section VI-C. Based on the simulation results, the variance in the completion time and the POS is negligible for all benchmarks; therefore, we only present the mean values in Fig. 12 to Fig. 14. The CPU time for the proposed on-line synthesis method ranges from 0.03 s to 0.18 s, which meets the requirement of real-time computation and further demonstrates that the computation time for on-line synthesis will not influence the completion time for the bioassay.

As shown in Fig. 12-14, larger TLLR and a smaller number of inserted errors result in higher POS for all three bioassays. However, larger TLLR leads a significant increase in the completion time. Therefore, there is a trade-off between the POS and the completion time.

In order to compare the proposed method with prior works [8] and [11], we randomly injected errors into the three benchmarks 30 times and calculated the mean value of both the POS and the completion time. For both methods, all operations that did not have injected errors were assumed to complete successfully with probability of one.

For the proposed method, when an error is detected, the corresponding local recovery procedure is invoked. If local recovery fails, no additional recovery approaches are utilized. For prior methods for comparison, when an error is detected, we must repeat all necessary operations since there is no specific local recovery procedure. We used the same probabilities that were presented in Section III for the computation. The results of this comparison are presented in Fig. 15. As the number of errors increases, the POS of the proposed

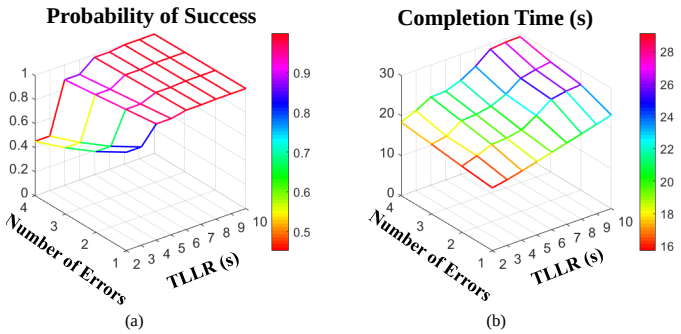


Fig. 12. Illustration of (a) the probability of success and (b) the completion time as the TLLR and the number of inserted errors are varied for the PCR benchmark.

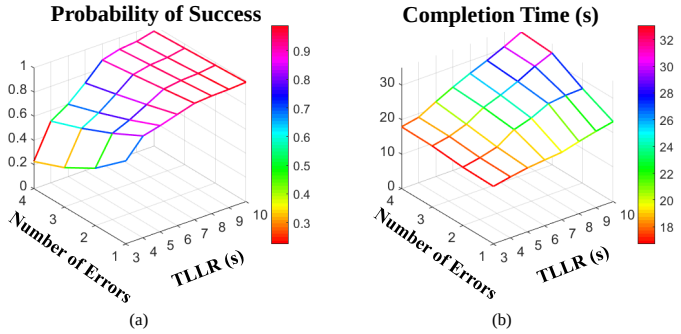


Fig. 13. Illustration of (a) the probability of success and (b) the completion time as the TLLR and the number of inserted errors are varied for the *in-vitro* diagnostic benchmark.

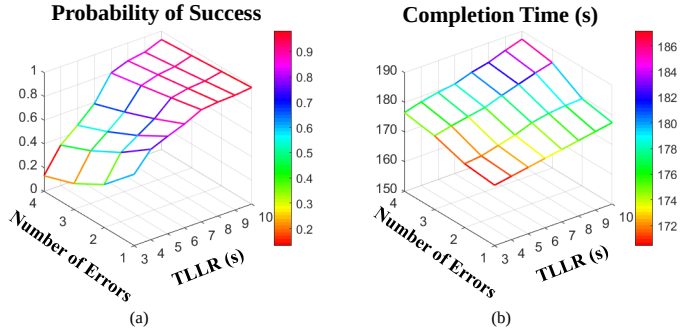


Fig. 14. Illustration of (a) the probability of success and (b) the completion time as the TLLR and the number of inserted errors are varied for the protein dilution benchmark.

method falls slowly while the POS for [8] falls rapidly for all benchmarks. Moreover, the proposed method also takes less completion time for most cases compared to [8].

C. Results for TLLR Selection

We finally compare the simulation results between the unified TLLR selection method (TLLR values are the same for all operations) and the TLLR selection method proposed in Section IV (TLLR values can be different for different operations). The comparison results for PCR benchmark, *in-vitro* diagnostic benchmark, and protein dilution benchmark are presented in Table V, Table VI, and Table VII, respectively.

In these three tables, ILP and UNI represent the ILP-based TLLR selection method and the unified TLLR selection method, respectively. Parameter P_{th} is the threshold POS, and parameter N represents the number of operations in the bioassay. Metrics include the minimum value, maximum value, and average value of TLLRs for all operations in

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.2017.2729347, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

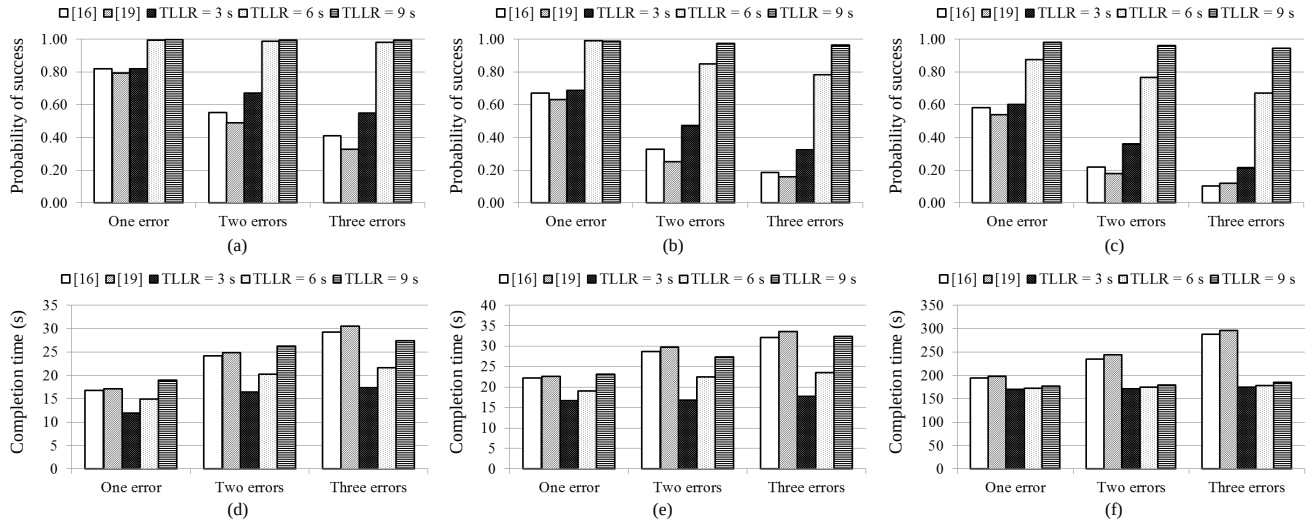


Fig. 15. Comparison between the proposed method with previous methods [8] and [11] on (a) the POS for PCR benchmark, (b) the POS for *in-vitro* diagnostic benchmark, (c) the POS for protein dilution benchmark, (d) the completion time for PCR benchmark, (e) the completion time for *in-vitro* diagnostic benchmark and (f) the completion time protein dilution benchmark.

TABLE V
COMPARISON BETWEEN THE ILP-BASED AND THE UNIFIED TLLR SELECTION METHOD FOR PCR BENCHMARK.

Parameters	$P_{th} = 0.95^N$		$P_{th} = 0.96^N$		$P_{th} = 0.97^N$		$P_{th} = 0.98^N$		$P_{th} = 0.99^N$	
	ILP	UNI	ILP	UNI	ILP	UNI	ILP	UNI	ILP	UNI
Minimum value	2.00 s	2.00 s	2.00 s	2.00 s	2.00 s	4.00 s	2.00 s	4.00 s	2.00 s	4.00 s
Maximum value	2.00 s	2.00 s	2.00 s	2.00 s	4.00 s	4.00 s	4.00 s	4.00 s	4.00 s	4.00 s
Average value	2.00 s	2.00 s	2.00 s	2.00 s	2.29 s	4.00 s	3.14 s	4.00 s	3.71 s	4.00 s

TABLE VI
COMPARISON BETWEEN THE ILP-BASED AND THE UNIFIED TLLR SELECTION METHOD FOR *IN-VITRO* DIAGNOSTIC BENCHMARK.

Parameters	$P_{th} = 0.95^N$		$P_{th} = 0.96^N$		$P_{th} = 0.97^N$		$P_{th} = 0.98^N$		$P_{th} = 0.99^N$	
	ILP	UNI	ILP	UNI	ILP	UNI	ILP	UNI	ILP	UNI
Minimum value	2.00 s	4.00 s	2.00 s	5.00 s	2.00 s	6.00 s	2.00 s	7.00 s	4.00 s	8.00 s
Maximum value	7.00 s	4.00 s	7.00 s	5.00 s	8.00 s	6.00 s	7.00 s	7.00 s	10.00 s	8.00 s
Average value	3.11 s	4.00 s	3.44 s	5.00 s	3.89 s	6.00 s	4.55 s	7.00 s	5.56 s	8.00 s

TABLE VII
COMPARISON BETWEEN THE ILP-BASED AND THE UNIFIED TLLR SELECTION METHOD FOR PROTEIN DILUTION BENCHMARK.

Parameters	$P_{th} = 0.95^N$		$P_{th} = 0.96^N$		$P_{th} = 0.97^N$		$P_{th} = 0.98^N$		$P_{th} = 0.99^N$	
	ILP	UNI	ILP	UNI	ILP	UNI	ILP	UNI	ILP	UNI
Minimum value	2.00 s	7.00 s	2.00 s	7.00 s	2.00 s	7.00 s	2.00 s	8.00 s	2.00 s	10.00 s
Maximum value	7.00 s	7.00 s	7.00 s	7.00 s	7.00 s	7.00 s	8.00 s	8.00 s	10.00 s	10.00 s
Average value	5.63 s	7.00 s	5.89 s	7.00 s	6.19 s	7.00 s	7.19 s	8.00 s	8.93 s	10.00 s

the corresponding benchmark. Based on our experiments, we found that the threshold POS has limited influence on the CPU time for both ILP and UNI methods. For UNI, the CPU time is $\sim$0 s for all three benchmarks. For ILP, the CPU time for PCR, *in-vitro* diagnostic, and protein dilution benchmark is 56 s, 492 s, and 6215 s, respectively.

As shown in Table V to Table VII, the minimum value, maximum value, and average value of TLLRs are always the same for the unified TLLR selection method. Compared with the unified TLLR selection method, the proposed ILP-based method results the difference between the minimum value, maximum value, and average value of TLLRs. As shown in the tables, the ILP method can effectively reduce the average value of TLLRs, which helps to reduce the time spent on error recovery. The saved time on error recovery can also help to reduce the completion time for the bioassay, which helps to improve the efficiency. For small benchmarks, UNI and ILP method are similar to each other. The ILP method significant outperforms UNI method for large bioassays.

VII. CONCLUSION

We have presented the first error-recovery strategy for MEDA biochips. We first described a classification of the outcomes of operations into different categories. Laboratory experiments using a fabricated MEDA biochip were used to estimate outcome probabilities for various operations. We then presented different probabilistic-timed-automata (PTA)-based error-recovery strategies for various types of local errors. We have also proposed an optimal time-limit selection method for local error recovery. An on-line synthesis technique and a control flow were proposed to connect the local recovery procedures with global error recovery for the complete bioassay. Simulation results for three benchmarks and comparison with prior methods highlight the effectiveness of the proposed error-recovery strategy.

REFERENCES

[1] M. Ibrahim et al., "Advances in design automation techniques for digital-microfluidic biochips," in *Formal Modeling and Verification of Cyber-Physical Systems*, pp. 190–223, 2015.

[2] K. Lai et al., "An intelligent digital microfluidic processor for biomedical detection," *Journal of Signal Processing Systems*, vol. 78, pp. 85–93, 2015.

[3] Y.-C. Ho et al., "Design of a micro-electrode cell for programmable lab-on-cmos platform," in *Proc. IEEE ISCAS*, pp. 2871–2874, 2016.

[4] Z. Li et al., "High-level synthesis for micro-electrode-dot-array digital microfluidic biochips," in *Proc. ACM/IEEE DAC*, pp. 146:1–6, 2016.

[5] Z. Li et al., "Error recovery in a micro-electrode-dot-array digital microfluidic biochip," in *Proc. IEEE/ACM ICCAD*, pp. 105:1–8, 2016.

[6] H. Verheijen et al., "Reversible electrowetting and trapping of charge: model and experiments," *Langmuir*, vol. 15, pp. 6616–6620, 1999.

[7] J.-Y. Yoon et al., "Preventing biomolecular adsorption in electrowetting-based biofluidic chips," *Analytical Chemistry*, vol. 75, pp. 5097–5102, 2003.

[8] Y. Luo et al., "Error recovery in cyberphysical digital microfluidic biochips," *IEEE TCAD*, vol. 32, pp. 59–72, 2013.

[9] M. Alistar et al., "Redundancy optimization for error recovery in digital microfluidic biochips," *Design Automation for Embedded Systems*, vol. 19, pp. 129–159, 2015.

[10] C. Jaress et al., "Rapid online fault recovery for cyber-physical digital microfluidic biochips," in *Proc. IEEE VTS*, pp. 1–6, 2015.

[11] Y. Zhao et al., "Integrated control-path design and error recovery in the synthesis of digital microfluidic lab-on-chip," *ACM JETC*, vol. 6, pp. 1–11, 2010.

[12] T. Xu et al., "A droplet-manipulation method for achieving high-throughput in cross-referencing-based digital microfluidic biochips," *IEEE TCAD*, vol. 27, pp. 1905–1917, 2008.

[13] G. Wang et al., "Digital microfluidic operations on micro-electrode dot array architecture," *IET Nanobiotechnology*, vol. 5, pp. 152–160, 2011.

[14] G. Norman et al., "Model checking for probabilistic timed automata," *Formal Methods in System Design*, vol. 43, pp. 164–190, 2013.

[15] F. Su et al., "Testing of droplet-based microelectrofluidic systems," in *Proc. IEEE ITC*, pp. 1192–1200, 2003.

[16] T. Xu et al., "Functional testing of digital microfluidic biochips," in *Proc. of IEEE ITC*, pp. 1–10, 2007.

[17] R. Fair et al., "Chemical and biological applications of digital-microfluidic devices," *IEEE Design & Test of Computers*, vol. 24, pp. 10–24, 2007.

[18] Y. Zhao et al., "On-line testing of lab-on-chip using reconfigurable digital-microfluidic compactors," *International Journal of Parallel Programming*, vol. 37, pp. 370–388, 2009.

[19] K. Lai et al., "A field-programmable lab-on-a-chip with built-in self-test circuit and low-power sensor-fusion solution in 0.35μm standard CMOS process," in *Proc. IEEE A-SSCC*, pp. 1–4, 2015.

[20] P. Brémaud, *Markov Chains: Gibbs Fields, Monte Carlo Simulation, and Queues*. Springer Science & Business Media, 2013.

[21] F. Su et al., "High-level synthesis of digital microfluidic biochips," *ACM JETC*, vol. 3, p. 1, 2008.

[22] M. Kwiatkowska et al., "PRISM 4.0: Verification of probabilistic real-time systems," in *Proc. CAV*, pp. 585–591, 2011.

[23] D. Bertsimas and J. Tsitsiklis, *Introduction to linear optimization*, vol. 6. Athena Scientific Belmont, MA, 1997.

[24] F. Su et al., "Unified high-level synthesis and module placement for defect-tolerant microfluidic biochips," in *Proc. of ACM/IEEE DAC*, pp. 825–830, 2005.

[25] F. Su et al., "Droplet routing in the synthesis of digital microfluidic biochips," in *Proc. of IEEE/ACM DATE*, pp. 1–6, 2006.

[26] Y. Zhao and K. Chakrabarty, "Cross-contamination avoidance for droplet routing in digital microfluidic biochips," *IEEE TCAD*, vol. 31, pp. 817–830, 2012.

[27] T.-W. Huang and T.-Y. Ho, "A two-stage integer linear programming-based droplet routing algorithm for pin-constrained digital microfluidic biochips," *IEEE TCAD*, vol. 30, pp. 215–228, 2011.

[28] P. R. et al., "A novel droplet routing algorithm for digital microfluidic biochips," in *Proc. ACM GLSVLSI*, pp. 441–446, ACM.

[29] Z. Chen et al., "Droplet routing in high-level synthesis of configurable digital microfluidic biochips based on microelectrode dot array architecture," *BioChip Journal*, vol. 5, pp. 343–352, 2011.

[30] Z. Li et al., "Droplet size-aware high-level synthesis for micro-electrode-dot-array digital microfluidic biochips," *IEEE TBioCAS*, 2017.

[31] K. S. Trivedi, *Probability and Statistics with Reliability, Queueing, and Computer Science Applications*. John Wiley & Sons, 2008.

[32] A. Gefen, *Bioengineering Research of Chronic Wounds: A Multidisciplinary Study Approach*. Springer Science & Business Media, 2009.

[33] F. Su et al., "Module placement for fault-tolerant microfluidics-based biochips," vol. 11, pp. 682–710, 2004.

Zipeng Li (S'14) received the bachelor's degree in electrical engineering from the Harbin Institute of Technology, Harbin, China, in 2013, and the master's and Ph.D. degree in computer engineering from Duke University, Durham, NC, USA, in 2015 and 2017, respectively.

He is currently a Senior DFT Engineer with Intel Corporation, Santa Clara, USA. He was a DFT intern with Apple Inc., Cupertino, CA, USA, NVIDIA Inc., Santa Clara, CA, USA, GlobalFoundries, Malta, NY, USA, and Taiwan Semiconductor Manufacturing Company (TSMC), San Jose, CA, USA, and a Visiting Scholar with the National Tsing Hua University, Hsinchu, Taiwan. His current research interests include design automation for biochips, design-for-test, 2.5-D/3-D ICs, and mixed-technology microsystems.

Kelvin Yi-Tse Lai received the B.S. and M.S. degrees from Feng Chia University, Taichung, Taiwan, in 2004 and National Yunlin University of Science and Technology, Yunlin, Taiwan, in 2006, respectively. He received the Ph.D. degree at National Chiao Tung University, Hsinchu, Taiwan, in 2015.

His research interests include the environmental, physiological and biological sensors design, wireless sensing system development, low-power and event-driven readout circuit, micro-fluidics lab-chips (LoC), wireless channel coding, and machine-learning-based feature extraction algorithm/architecture design for big data application.

John McCrone received a Master of Science in Electrical and Computer Engineering from Duke University in 2015. He also previously earned a Bachelor of Science in Physics from the United States Military Academy at West Point in 2009 and served five years in the United States Army as a Military Intelligence Officer.

He currently works for International Business Machines Corporation (IBM) as a Software Engineer in Research Triangle Park, NC in the general area of Cloud Computing. His current research interests include Computer Networks, Distributed Systems, and Cryptography.

Po-Hsien Yu received the B.E. degree in mechanical engineering from National Sun Yat-sen University, Kaohsiung, Taiwan, in 2008, and the M.S. degree in electronic engineering from National Chiao Tung University, Hsinchu, Taiwan, in 2016.

He is currently a System Application Engineer with HIMAX Corporation in Taiwan. His research interests include digital integrated circuit, microfluidic system, embedded system, and VLSI design.

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 at Duke University. Prof. Chakrabarty is a recipient of the National Science Foundation 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), the ACM Transactions on Design Automation of Electronic Systems Best Paper Award (2017), and over a dozen best paper awards at major IEEE conferences. He is also a recipient of the IEEE Computer Society Technical Achievement Award (2015), the IEEE Circuits and Systems Society Charles A. Desoer Technical Achievement Award (2017), and the Distinguished Alumnus Award from the Indian Institute of Technology, Kharagpur (2014). He is a Research Ambassador of the University of Bremen (Germany) and a Hans Fischer Senior Fellow at the Institute for Advanced Study, Technical University of Munich, Germany. Prof. Chakrabarty's current research interests include: testing and design-for-testability of of integrated circuits and systems; microfluidic biochips; data analytics for fault diagnosis, failure prediction, anomaly detection, and hardware security; cybermanufacturing. 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*.

Miroslav Pajic (S'06-M'13) received the Dipl. Ing. and M.S. degrees in electrical engineering from the University of Belgrade, Serbia, in 2003 and 2007, respectively, and the M.S. and Ph.D. degrees in electrical engineering from the University of Pennsylvania, Philadelphia, in 2010 and 2012, respectively.

He is currently an Assistant Professor in the Department of Electrical and Computer Engineering at Duke University. He also holds a secondary appointment in the Computer Science Department. Prior to joining Duke, Dr. Pajic was a Postdoctoral Researcher in the PRECISE Center, University of Pennsylvania. His research interests focus on the design and analysis of cyber-physical systems and in particular real-time and embedded systems, distributed/networked control systems, and high-confidence medical devices and systems.

Dr. Pajic received various awards including the National Science Foundation (NSF) CAREER Award, Office of Naval Research (ONR) Young Investigator Program Award, 2011 ACM SIGBED Frank Anger Memorial Award, the Joseph and Rosaline Wolf Best Dissertation Award from Penn Engineering, the Best Paper Award at the 2014 ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS), and the Best Student Paper award at the 2012 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS).

Tsung-Yi Ho (SM'10) 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 presented 10 tutorials and contributed 11 special sessions in ACM/IEEE conferences, all in design automation for microfluidic biochips. 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*, *ACM Transactions on Embedded Computing 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, including DAC, ICCAD, DATE, ASP-DAC, ISPD, ICCD, etc.

Chen-Yi Lee received the B.S. degree from National Chiao Tung University (NCTU), Hsinchu, Taiwan, in 1982, and the M.S. and Ph.D. degrees from Katholieke University Leuven (KUL), Leuven, Belgium, in 1986 and 1990, respectively, all in electrical engineering. From 1986 to 1990, he was with IMEC/VSDM, Belgium, working in the area of architecture synthesis for DSP.

In February 1991, he joined the department of electronics engineering and from 2003-2006, he was the chairman. During 2000-2003, he was the director of National CHIP Implementation Center (CIC) and the coordinator of Microelectronics Program of Engineering Division, NSC, Taiwan. He was the dean of office of research and development in 2007-2010, National Chiao Tung University (NCTU). Currently, he serves as Co-Program Director of National Program of Intelligent Electronics (NPIE) and professor of the department of electronics engineering, NCTU. His research interests mainly include VLSI algorithms and architectures for high-throughput DSP applications. He is also active in various aspects of micro sensing, low-power system-on-chip, and deep learning.

Dr. Lee served as the program committee member of IEEE ISSCC in 20042006, DATE TPC member in 2006-2007, IEEE ASSCC TPC member in 2006-2014, IEEE VLSI Symposium JFE Program Committee member in 2010-2014, IEEE TCAS-II Associate Editor in 2010-2011, and the Past-Chair of Taipei Chapter of IEEE Circuits and Systems (CAS) Society. He received the Outstanding Technology Licensing Award from National Science Council in 2007-2008, and Industry Contribution Award from Ministry of Economic Affairs and Outstanding Research Award from National Science Council in 2009.