Zipeng Li · Krishnendu Chakrabarty
Tsung-Yi Ho · Chen-Yi Lee

# Micro-Electrode-Dot-Array Digital Microfluidic Biochips

## Design Automation, Optimization, and Test Techniques

Springer

# Micro-Electrode-Dot-Array Digital Microfluidic Biochips

Zipeng Li • Krishnendu Chakrabarty
Tsung-Yi Ho • Chen-Yi Lee

# Micro-Electrode-Dot-Array Digital Microfluidic Biochips

## Design Automation, Optimization, and Test Techniques

Springer

Zipeng Li
Intel (United States)
Santa Clara, CA, USA

Tsung-Yi Ho
National Tsing Hua University
Hsinchu, Taiwan

Krishnendu Chakrabarty
Department of ECE
Duke University
Durham, NC, USA

Chen-Yi Lee
National Chiao Tung University
Hsinchu, Taiwan

ISBN 978-3-030-02963-0 ISBN 978-3-030-02964-7 (eBook)
https://doi.org/10.1007/978-3-030-02964-7

Library of Congress Control Number: 2018962868



This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

*To my beloved wife,*
*Mengjie Wei.*
*— Zipeng Li*

# Preface

Digital microfluidic biochips (DMFBs) are revolutionizing many biochemical analysis procedures, e.g., high-throughput DNA sequencing and point-of-care clinical diagnosis. However, today's DMFBs suffer from several limitations: (1) constraints on droplet size and the inability to vary droplet volume in a fine-grained manner; (2) the lack of integrated sensors for real-time detection; (3) the need for special fabrication processes and the associated reliability/yield concerns.

To overcome the above limitations, DMFBs based on a micro-electrode-dot-array (MEDA) architecture have recently been proposed. 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, i.e., thousands of microelectrodes can be integrated on a MEDA biochip. The MEDA architecture allows microelectrodes to be dynamically grouped to form a micro-component that can perform different microfluidic operations on the chip.

Design automation tools can reduce the difficulty of MEDA biochip design and help to ensure that the manufactured MEDA biochips are versatile and reliable. However, the increase in integration level introduces new challenges in the design optimization and testing of MEDA biochips, which impede their further adoption and deployment. This book is focused on enhancing the automated design and use of MEDA biochips and on developing a set of solutions to facilitate the full exploitation of design complexities that are possible with current MEDA platform. In order to fully exploit MEDA-specific advantages (e.g., real-time droplet sensing), four key research challenges are addressed in this book; these include high-level synthesis, error recovery, sample preparation optimization, and testing.

The book first presents a droplet-size-aware synthesis approach that can configure the target bioassay on a MEDA biochip. The proposed synthesis method targets reservoir placement, operation scheduling, module placement, and routing of droplets of various sizes. The book also presents an analytical model for estimating droplet velocity, and the model was experimentally validated using fabricated MEDA chips. In order to effectively exploit the advantages offered by MEDA, MEDA-specific fluidic operations, such as droplet diagonal movement and droplet shape morphing, have been considered in the proposed synthesis approach.

Next, this book presents an efficient error-recovery strategy to ensure the correctness of assays executed on MEDA biochips. By exploiting MEDA-specific advances in droplet sensing, the book presents the first probabilistic timed automata (PTA)-based error-recovery technique to dynamically reconfigure the MEDA biochip using real-time data provided by on-chip sensors. The book also presents an on-line synthesis technique and a control flow to connect local recovery procedures with global error recovery for the complete bioassay.

A potentially important application of MEDA biochips lies in sample preparation via a series of dilution steps. Sample preparation in digital microfluidics refers to the generation of droplets with target concentrations for on-chip biochemical applications. The book presents the first droplet size-aware and error-correcting sample preparation method for MEDA biochips. In contrast to previous methods, the proposed approach considers droplet sizes and incorporates various mixing models in sample preparation. Moreover, the proposed approach utilizes MEDA-enabled microfluidic operations and fully exploits the feature of real-time droplet sensing on MEDA biochips for error correction. The proposed method can significantly reduce the number of dilution steps, hence the time needed for sample preparation.

In order to ensure high confidence in the outcome of biochemical experiments, MEDA biochips must be adequately tested before they can be used for bioassay execution. The book presents efficient structural and functional test techniques for MEDA biochips. The proposed structural test techniques can effectively detect defects and identify faulty microcells, and the proposed functional test techniques address fundamental fluidic operations on MEDA biochips.

In summary, the book tackles important problems related to key stages of MEDA chip design and usage. The results emerging from this dissertation provide the first set of comprehensive design automation and test solutions for MEDA biochips. These methods are expected to not only shorten the product development cycle but also accelerate the adoption and further development of MEDA biochips by facilitating the full exploitation of design complexities that are possible with current fabrication techniques. It is anticipated that chip designers and chip users will also benefit from these methods.

Santa Clara, CA, USA — Zipeng Li
Durham, NC, USA — Krishnendu Chakrabarty
Hsinchu City, Taiwan — Tsung-Yi Ho
Hsinchu City, Taiwan — Chen-Yi Lee

# Acknowledgments

The authors acknowledge Professor Miroslav Pajic from Duke University, Durham, NC, USA, for valuable discussions and collaboration. The authors also acknowledge Dr. Kelven Yi-Tse Lai and Po-Hsien Yu from National Chiao Tung University, Hsinchu, Taiwan, for constructive suggestions.

Zipeng Li and Krishnendu Chakrabarty acknowledge the financial support received from the US National Science Foundation.

Tsung-Yi Ho and Chen-Yi Lee acknowledge the financial support received from the Taiwan Ministry of Science and Technology.

# Contents

# Chapter 1
# Introduction


## 1.1 Chapter Overview

Due to their emergence as an efficient platform for point-of-care clinical diagnostics, digital-microfluidic biochips (DMFBs) have received considerable attention in recent years. They combine electronics with biology, and they integrate various bioassay operations, such as sample preparation, analysis, separation, and detection. In this chapter, we first present an overview of digital-microfluidic biochips. We next describe emerging computer-aided design (CAD) tools for the automated synthesis and optimization of biochips from bioassay protocols. The chapter includes solutions for fluidic-operation scheduling, module placement, droplet routing, and pin-constrained chip design. We also show how recent advances in the integration of sensors into a DMFB can be exploited to provide cyberphysical system adaptation based on feedback-driven control.

Microfluidic biochips, also referred to as lab-on-a-chip, are revolutionizing many areas of biomolecular analysis, such as high-throughput DNA sequencing, point-of-care clinical diagnosis, protein crystallization, enzymatic analysis, proteomic analysis, and environmental toxicity monitoring [1–5]. Many traditional microfluidic biochips are based on the flow-based platform with microchannels, microvalves, micropumps, and addressable chambers. With the help of these microdevices, various biochemical procedures, including polymerase chain reaction (PCR), DNA purification, and protein crystallization, can be performed using flow-based microfluidic devices. For those flow-based biochips, a serious of fluidic operations, such as sample loading, separations, and mixing, occur along paths from inputs to outputs; as a result, these devices suffer from lack of reconfigurability and defect tolerance. Moreover, flow-based chips are difficult to integrate and scale down, as the scale of each component may have a decisive impact on the chip performance [6]. Therefore, flow-based microfluidic biochips are suitable for a narrow range of bioassays [7, 8].



Z. Li et al., *Micro-Electrode-Dot-Array Digital Microfluidic Biochips*,
https://doi.org/10.1007/978-3-030-02964-7_1

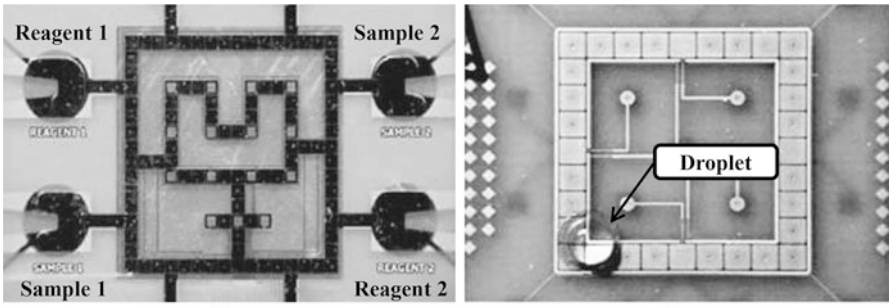



**Fig. 1.1** Fabricated digital microfluidic biochips: (**a**) glass substrate [11]; (**b**) PCB substrate [12]

An alternative category of microfluidic biochip technology, referred to as digital microfluidic biochips (DMFBs) and based on the principle of electrowetting-on-dielectric (EWOD) [3, 9], is an attractive platform to overcome the above limitations. DMFBs use discrete fluid packets as carriers to implement various fluidic operations, biochemical reactions, and detections at the micro-scale. The major advantages of digital microfluidics include: (1) simple control of multiple and parallel fluidic operations on a potentially large scale; (2) the reconfigurability of fluidic modules controlled by two-dimensional addressable control sites.

DMFBs are compatible with various substrates, e.g., silicon [10], glass [11], and PCB [12]. A range of fabrication methods have been developed for DMFBs to meet the needs of various applications. Two examples of DMFBs, which are fabricated with a glass substrate and a PCB substrate respectively, are shown in Fig. 1.1.

In this chapter, we will introduce basic concepts and background for digital microfluidics and provide motivation for the design automation and test techniques for digital microfluidics. Section 1.2 presents an overview of digital microfluidics and the underlying working principle of theses devices. Section 1.3 introduces a new digital microfluidics-based architecture, namely micro-electrode-dot-array (MEDA), and its associated advantages over conventional DMFBs. Section 1.4 presents an overview of the design automation techniques (e.g., high-level synthesis and chip-level design) and test techniques (e.g., structural test and functional test) for DMFBs and the motivation for corresponding research on MEDA biochips. Finally, an outline of this book is provided in Sect. 1.5.

## 1.2 Overview of Digital Microfluidics

A DMFB is an example of a lab-on-a-chip that is capable of carrying out biochemistry on a chip. A typical DMFB consists of a bottom layer and a top layer [13]. Today's DMFBs can be integrated with sensors and intelligent software-based control (i.e., cyberphysical DMFBs). For example, a photodetector (PD) for

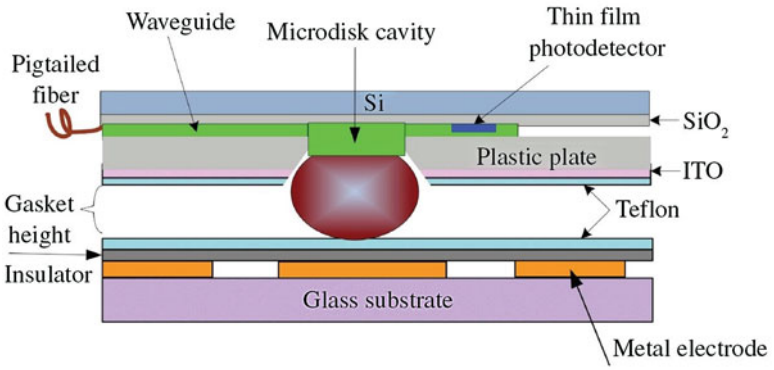



**Fig. 1.2** A sideview of a two-layer digital microfluidic biochip with an integrated photodetector (PD) [14]

optical sensing can be integrated into the top plate of the DMFB [14]; see Fig. 1.2. A dielectric layer is deposited on both surfaces of bottom and top plate to prevent any electrical conduction between the electrode and the droplet. Because droplets with biological samples are typically of nanoliter or picoliter volumes, a filler medium such as silicone oil is used between the two plates to avoid evaporation and reduce the likelihood of cross-contamination. Manipulated using electrowetting-on-dielectric (EWOD), droplets of samples and reagents can be transported along the electrodes and fluidic operations such as dispensing, transporting, mixing, dilution and splitting can be carried out on the chip.

Compared with conventional benchtop procedures, a DMFB offers advantages of lower cost, less power consumption, easier system integration, and less human error. In recent years, with the integration of different types of sensors (such as capacitive sensors [15] and optical sensors [14] in a cyberphysical platform), real-time feedback and software-based control can be realized and a DMFB becomes even more attractive.

In following subsections, we describe the principle of electrowetting-on-dielectric (EWOD), basic fluidic operations on DMFBs, and some real-life applications on different digital microfluidic platforms.

### *1.2.1 Principle of EWOD and Fluidic Operations*

In this subsection, we first introduce the principle of electrowetting-on-dielectric (EWOD) and then describe basic fluidic operations that can be achieved on a DMFB.

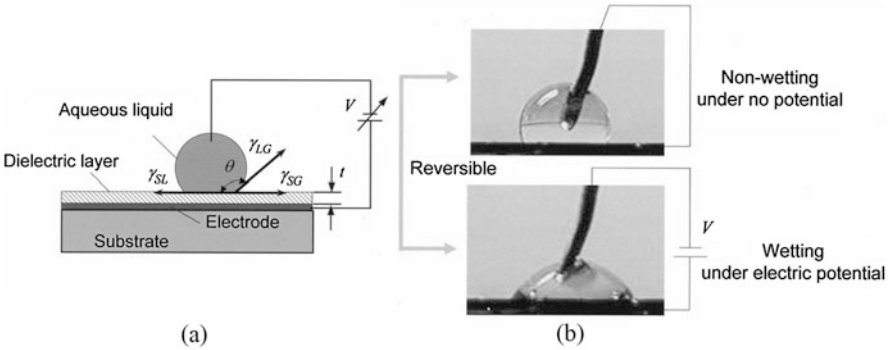



**Fig. 1.3** Principle of electrowetting on dielectric (EWOD). (**a**) Sideview of a DMFB. (**b**) EWOD demonstration [18]

#### 1.2.1.1 Principle of EWOD

EWOD is a phenomenon where an electric field can influence the wetting behavior of a conductive droplet in contact with a hydrophobic, insulated electrode. When an electric voltage is applied between the liquid droplet and the electrode, an electric field across the insulator is generated, which lowers the interfacial tension between the liquid and the insulator surface (as illustrated in Fig. 1.3) according to the Lippman-Young equation [16], see (1.1):

$$\gamma_{SL}(V) = \gamma_{SL}(V = 0) - \frac{C}{2}V^2 \tag{1.1}$$

where $\gamma_{SL}$ is the solid-liquid interfacial tension, $C(\mathrm{F}/m^2)$ is the capacitance of the dielectric layer, and $V$ is equivalent to the applied voltage. At the three-phase contact line, Lippman-Young equation describes the relationship among contact angle and interfacial tensions:

$$\cos\theta = \frac{\gamma_{SG} - \gamma_{SL}}{\gamma_{LG}} \tag{1.2}$$

where $\gamma_{SG}$ is the solid-gas interfacial tension, and $\gamma_{LG}$ is the liquid-gas interfacial tension. Substituting (1.1) into (1.2), the change of the contact angle caused by electro potential can be described by (1.3):

$$\cos\theta = \cos\theta_0 - \frac{\varepsilon_0\varepsilon}{2\gamma_{LG}t}V^2 \tag{1.3}$$

where $\theta_0$ is the equilibrium contact angle with no applied potential, $\varepsilon_0$ is the permittivity of vacuum, $\varepsilon$ is the dielectric constant of the dielectric layer, and $t$ is the thickness of the dielectric layer.

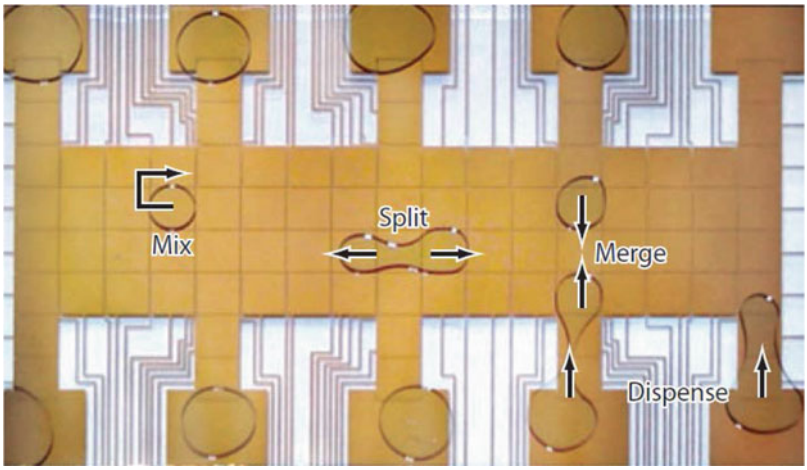



**Fig. 1.4** Droplet operations in a DMFB, including mixing, splitting, merging, and dispensing from reservoirs [17]

Since droplet movement is occurred because of capillary pressure that results from asymmetric contact angles across the droplet [17], the driving force, $F$, can be calculated using (1.4):

$$F = L\gamma(\cos\theta - \cos\theta_0) = \frac{\varepsilon_0\varepsilon_r L V^2}{2d} \tag{1.4}$$

where $L$ is the length of the contact line overlapping the actuated electrode.

#### 1.2.1.2 Fluidic Operations

According to the principle of EWOD, if we apply a series of potentials to electrodes on the chip, various fluidic operations can be achieved on the DMFB. Basic fluidic operations include: (1) transporting, (2) dispensing and storing, (3) mixing, and (4) splitting and merging (see Fig. 1.4). Advanced fluidic operations can be regarded as a group of basic fluidic operations.

*Transporting* A liquid droplet on the DMFB can be transported by asymmetrically changing the interfacial tension, which can be achieved by applying an electrical potential. Droplet transportation is performed over the electrodes to connect different fluidic modules on the chip. Details of droplet transportation can be found in [19]. Droplet transportation velocities of up to 100 mm/s at 60 V have been achieved for droplets with volumes in the range 100–1000 nl [3].

*Dispensing and Storing* Dispensing (droplet generation) is a key fluidic operation for a DMFB, because it provides fluidic I/O and the world-to-chip interface.

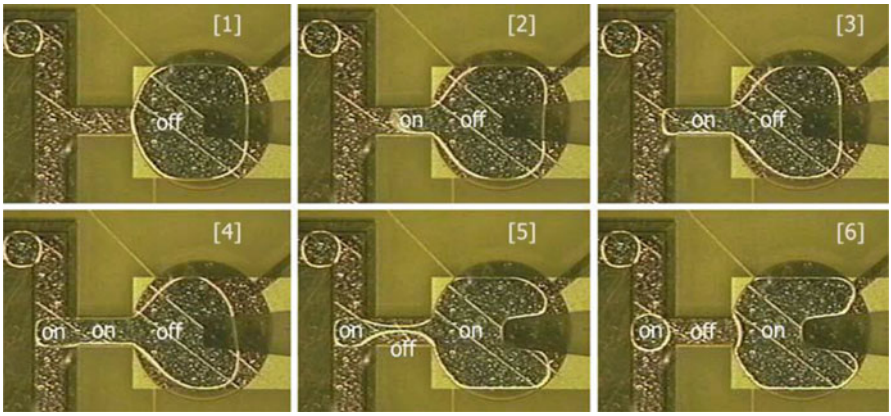



**Fig. 1.5** Droplet dispensing from an on-chip reservoir [11]

Generally, droplet dispensing refers to the process of generating small-volume liquid droplets on the unit electrode for manipulation on the DMFB. Droplets can also be stored in the on-chip reservoir.

In order to dispense droplets from an on-chip reservoir, liquid needs to be first pulled out of the reservoir and then be separated from the reservoir: this progression can be seen in Fig. 1.5.

*Splitting and Merging* The splitting and merging operations both use three consecutive electrodes. In splitting, the outer two electrodes are turned on, and the inner electrode is turned off. In general, the hydrophilic forces induced by the two outer electrodes stretch the droplet while the hydrophobic forces in the center pinch off the liquid into two small droplets [20]. For the merging operation, the inner electrode is turned on, and the outer two electrodes are turned off. The progression of droplet splitting and merging can be seen in Fig. 1.6.

*Mixing* Mixing on a DMFB can either be used for pre-processing, sample dilution, or for reactions between samples of particular concentration ratios. If we define the consecutive on and off of one electrode as one cycle, mixing operations usually takes around 1000 cycles [3]. Creating turbulent flow in mixing regions or creating multilaminates are two effective ways to speed up mixing.

### *1.2.2 Applications of Digital Microfluidics*

Because the DMFB is a versatile fluid handling platform with many unique advantages, it has been applied to a large number of applications in biochemistry, such as enzymatic reactions, immunoassays, DNA-based applications and clinical diagnosis [3, 13, 17, 21–23]. Several such examples are described below.

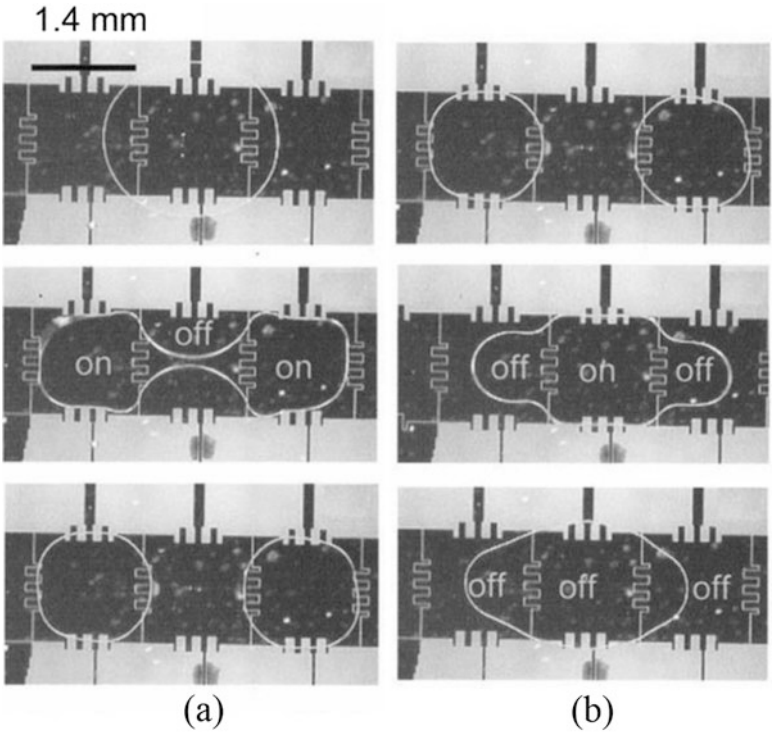



**Fig. 1.6** Sequential images of successful (**a**) splitting and (**b**) merging of droplets at 25 V (gap size d = 70 μm, electrode is 1.4 mm × 1.4 mm, volume is 0.2 μl) [18]

#### 1.2.2.1 Clinical Diagnostics

A typical clinical diagnostic procedure can usually be divided into four sequential stages: (1) sample collection, (2) sample preparation, (3) analytical processing, and (4) detection. DMFBs have been engaged in dealing with stages (2) and (3). In one example, a DMFB has been developed to carry out real-time PCR, as shown in [21]. In [21], results generated by DMFBs have been shown to match with results generated by a reference approach, while the DMFB-based approach is three times faster. An system view of a commercial DMFB and the corresponding board-level schematic are shown in Fig. 1.7.

#### 1.2.2.2 Proteomics

Proteomics experiments usually require a series of steps involving sample processing before analysis (e.g., using spectrometers) can be performed. Sample processing steps are usually tedious and time-consuming. The DMFB platform

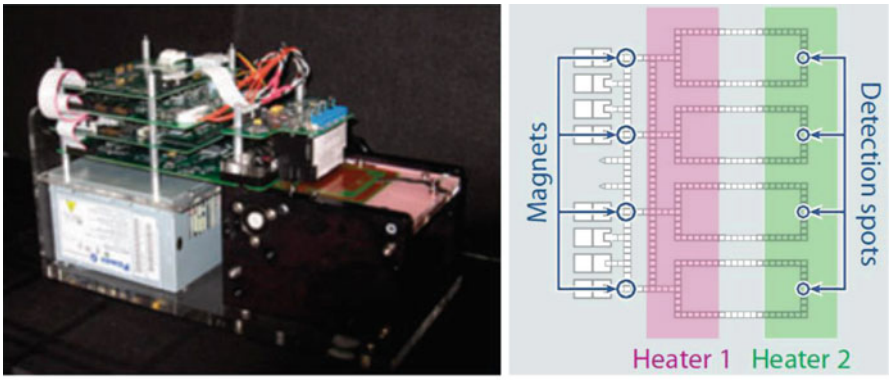



**Fig. 1.7** (**a**) A self-contained DMFB system for multiplexed real-time polymerase chain reaction (PCR) and (**b**) corresponding printed circuit board schematic for PCR [21]

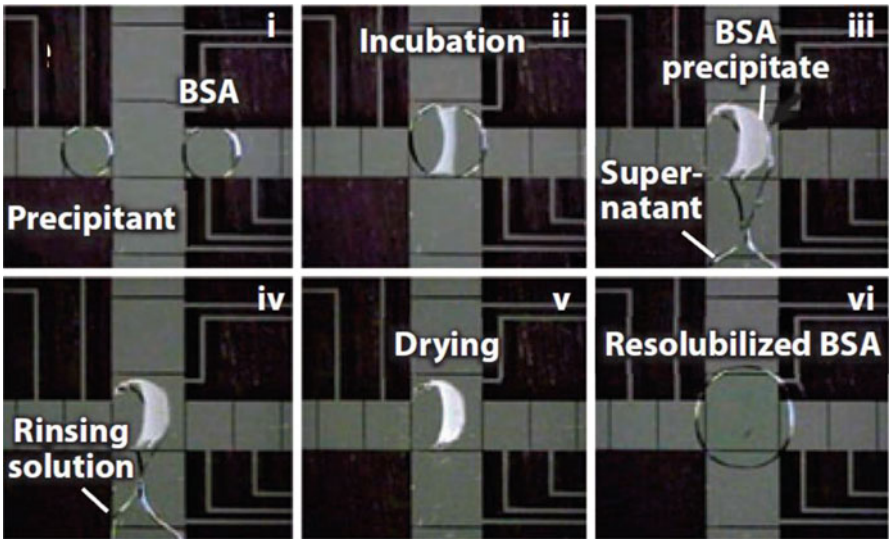



**Fig. 1.8** Video screenshots depicting the extraction and purification of proteins by precipitation [22]

is suitable for proteomics because of its ability to individually address many reagents simultaneously. A DMFB-based protocol has been developed to extract and purify proteins from complex biological mixtures [22]. This method has a protein recovery efficiency of nearly 80% and it requires much less time when compared with conventional approaches. The video screenshots depicting the extraction and purification of proteins by precipitation are shown in Fig. 1.8.

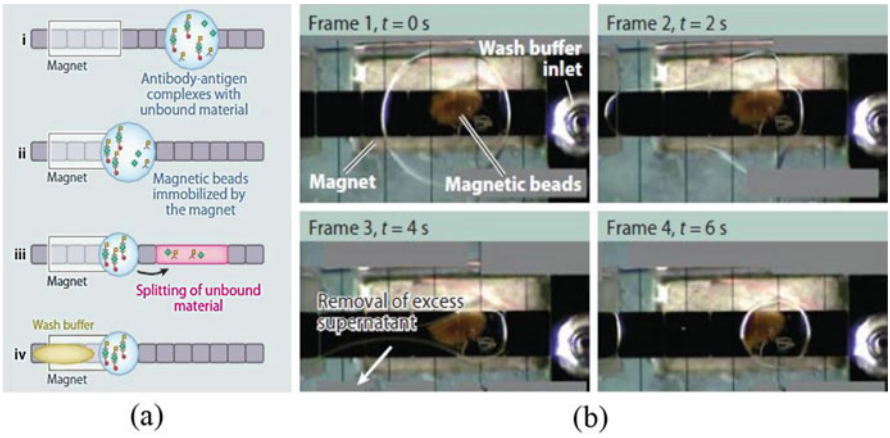



**Fig. 1.9** (**a**) Protocol for heterogeneous magnetic bead-based immunoassay on a DMFB and (**b**) video sequence of magnetic bead washing performed by removing the excess supernatant on the chip [23]

#### 1.2.2.3 Immunoassays

An immunoassay is a technique that exploits specific antibody-antigen interactions for the detection of relevant analytes. In these assays, antibodies aims to bind to the target analyte, which can also be accomplished by other binding molecules (e.g., oligonucleic acids and peptide-based aptamers). The work in [23] used an oil-filled DMFB to detect insulin and interleukin-6 by using droplets carrying paramagnetic beads modified with antibodies. An illustration of the protocol and a video sequence of magnetic bead washing are shown in Fig. 1.9.

## 1.3 Background of MEDA Biochips

A digital microfluidic biochip (DMFB) is an attractive technology platform for automating laboratory procedures in biochemistry. However, today's DMFBs suffer from several limitations: (1) constraints on droplet size and the inability to vary droplet volume in a fine-grained manner; (2) the lack of integrated sensors for real-time detection; (3) the need for special fabrication processes and reliability/yield concerns [15]. To overcome the above problems, DMFBs based on a micro-electrode-dot-array (MEDA) architecture have recently been demonstrated [24, 25]. We first present and introduce the basic architecture and functionalities of MEDA biochips and we then describe several novel characteristics specific to MEDA biochips. Those specific characteristics cannot be easily implemented on conventional DMFBs, and they can help with improving the chip performance. Therefore, those characteristics need to be carefully considered when applying applications on a MEDA biochip.

### 1.3.1 MEDA Architecture

To overcome the above limitations, DMFBs based on a micro-electrode-dot-array (MEDA) architecture have been proposed [24, 26, 27]. 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* [24]. 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 biochips have been fabricated using TSMC 0.35 μm CMOS technology [27], and these devices can use a power-supply voltage of only 3.3 V [26]. Therefore, MEDA biochips overcome some of the key limitations of conventional digital microfluidic biochips, e.g., the need for higher voltages (tens of volts) and a custom fabrication process.

A system illustration of MEDA architecture is shown in Fig. 1.10. MEDA architecture integrates the microfluidic interface and microelectronic interface together; a shielding layer is used to separate these two layers [26]. The size of the microelectrodes can be over 20 times smaller (e.g., 37 μm [27] in length) than conventional electrodes. Each microelectrode cell (MC) consists of a microelectrode, an activation circuit, and a sensing circuit. Three important functions of an MC are: (1) droplet actuation, (2) droplet maintenance and (3) droplet sensing. High voltage (HV) is applied on the top reference electrode in a MEDA biochip. For droplet actuation, the voltage on the microelectrode is set to be ground and droplet can be driven to the actuated microelectrodes; For droplet maintenance, the voltage on the microelectrode is set to be HV and droplet is maintained at the current location. Each MC can also sense the capacitance difference and store the sensing data into 1-bit register to construct 2D location map.

MEDA biochips allow the dynamic grouping of microelectrodes to form different shapes and fluidic modules, such as mixers and diluters (see Fig. 1.10). A dielectric layer and a hydrophobic layer are deposited above the microelectrodes, and droplets are sandwiched between two plates. A side-view of MEDA biochips is shown in Fig. 1.11. Once the electrodes are actuated, the EWOD force induces droplet motion. In this way, fluidic operations (e.g., cutting and mixing) on droplets of different sizes can be achieved on the chip.

### 1.3.2 Major Functions for Each Microelectrode Cell

There are three major functions implemented by an MC: (1) droplet actuation, (2) droplet maintenance, and (3) droplet sensing. The block diagram of the actuation/sensing circuit in each MC is shown in Fig. 1.12. Details of three MC functions are described below.

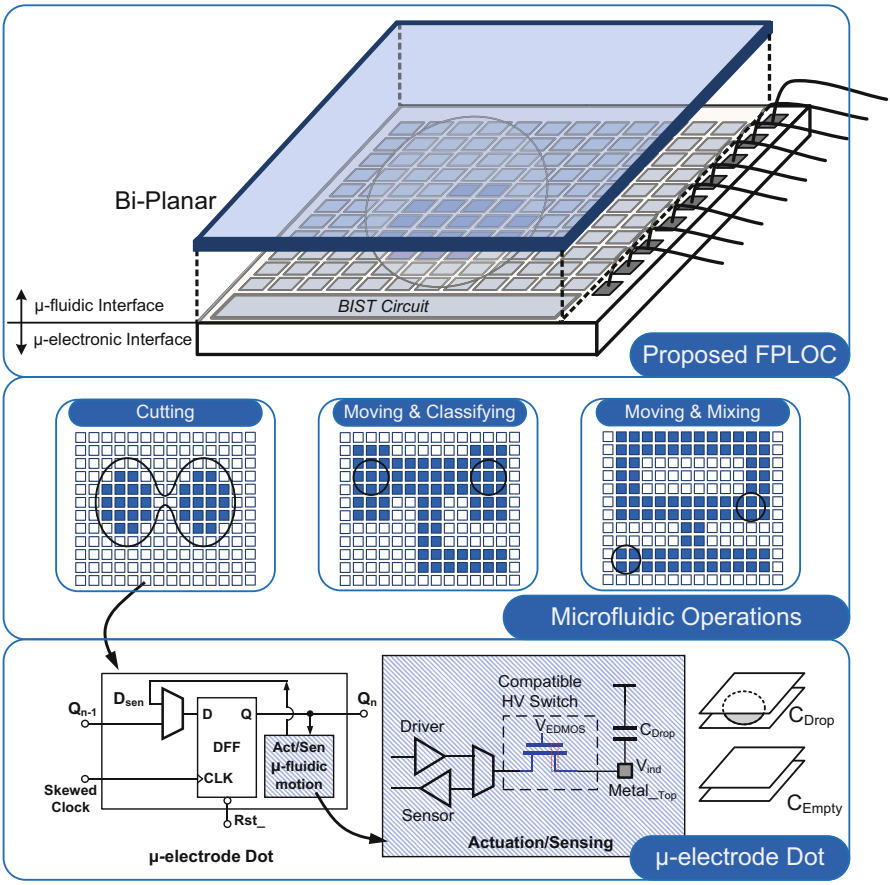



**Fig. 1.10** System illustration of the MEDA architecture [25]

- **Droplet actuation:** When $CONT = 0, ACT = 1, IN = 1$, and HV on the top plate is enabled, transistors M3 and M4 (M1 and M2) are "on" ("off"). Accordingly, the microelectrode is driven to the ground voltage and the droplet can therefore be actuated because there is a potential difference between the top plate and the bottom microelectrode.
- **Droplet maintenance:** When $CONT = 0, ACT = 1, IN = 0$, and HV on the top plate is enabled, transistors M1 and M2 (M3 and M4) are "off" ("on"). Therefore, the voltage on the microelectrode follows the voltage on the top plate, and the droplet location does not change. Because the drain of transistor M4 is directly connected to HV microelectrode, M4 is a specially developed extended-drain MOSFET (EDMOS), which significantly increases avalanche breakdown voltage and reduces leakage current [25]. Hence, EDMOS helps to improve the actuation capability of MEDA biochips.

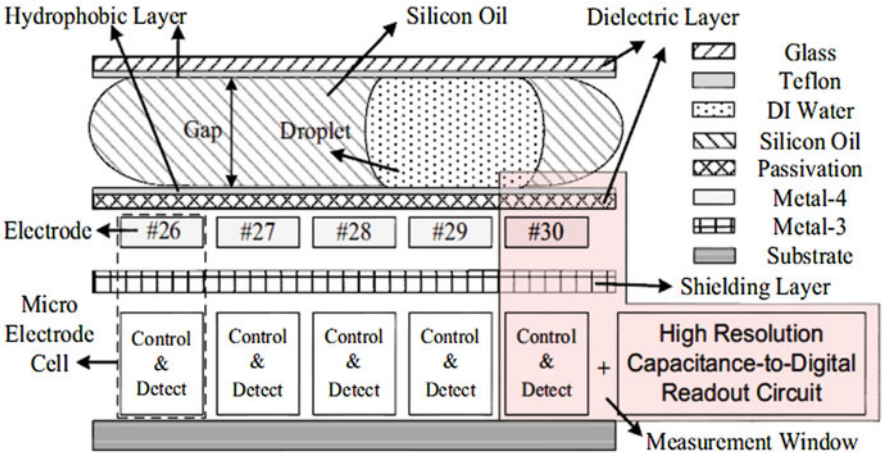



**Fig. 1.11** A side-view of a MEDA biochip [27]

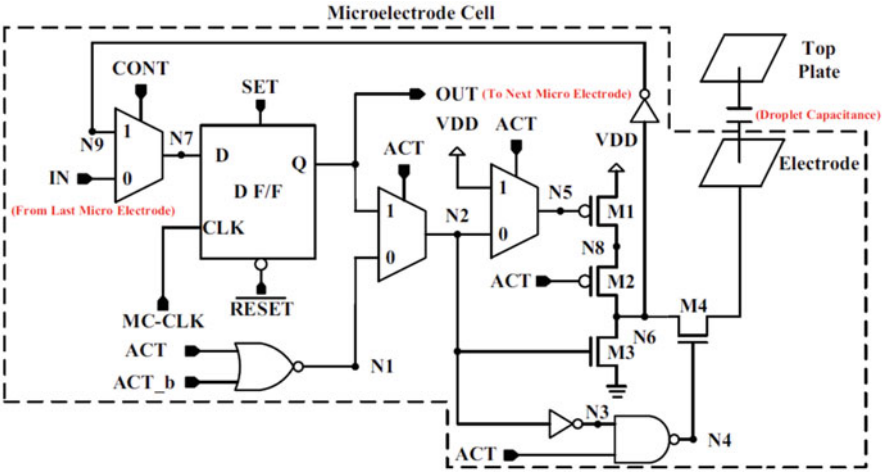



**Fig. 1.12** Illustration of the actuation/sensing circuit in each MC [27]

- **Droplet sensing:** For the droplet-sensing function, the top plate is switched to the ground voltage. Transistors M3 and M4 are first turned on to discharge the parasitic electrical carriers from the bottom microelectrode. After the auto-zeroing procedure, transistors M1 and M2 inject a $\mu$A-level charging current into the microelectrode. Because the charging time depends on the effective capacitance and the droplet appearance can result in a change of the capacitance of a few fFs, we are able to detect the existence of the droplet based on the value of the charging time [25].

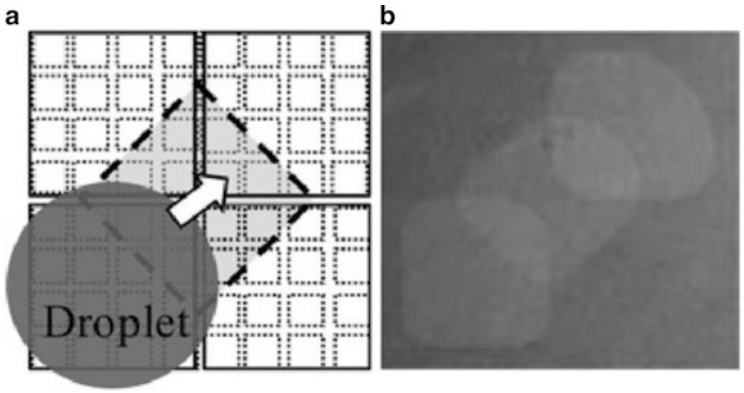



**Fig. 1.13** (**a**) Illustration of the diagonal droplet movement using an interim electrode and (**b**) experimental demonstration [24]

### *1.3.3 Novel Characteristics for MEDA*

Because of the special architecture of MEDA biochips, there are several novel characteristics that are specific to MEDA biochips. Those characteristics include but not limited to droplet shape morphing [28], diagonal droplet movement [24], diagonal droplet splitting [24], lamination mixing [24], and real-time droplet sensing [29]. These characteristics extend the flexibility of conventional DMFBs and help to better control droplets on MEDA biochips [24, 26]. Here we briefly introduce two MEDA characteristics: (1) diagonal droplet movement, and (2) lamination mixing.

#### 1.3.3.1 Diagonal Droplet Movement

Since microelectrodes can be dynamically grouped to form a functional electrode, diagonal droplet transportation can be achieved on MEDA biochips. Diagonal droplet movement cannot be achieved on conventional DMFBs because there is very small contact area between the droplet and the targeted electrode. On MEDA biochips, an interim electrode (see dotted square in Fig. 1.13) can be formed to facilitate diagonal droplet movement. This feature is presented in Fig. 1.13a and experimental demonstration is shown in Fig. 1.13b. Diagonal droplet movement can note only increase the flexibility of droplet movement but also significantly reduce the routing time, as reported in [30].

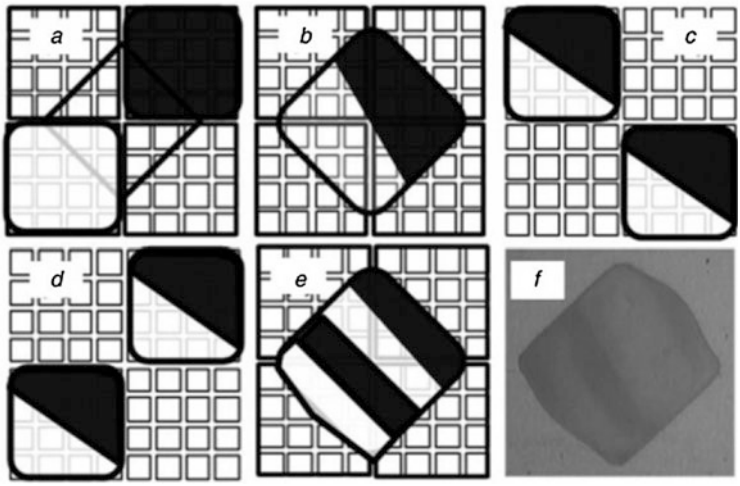



**Fig. 1.14** Illustration of SAR lamination mixing on the MEDA biochip [24]. (**a**) step1: two droplets are placed diagonally. (**b**) step2: two droplets are merged. (**c**) step 3: the merged droplet is split into two droplets. (**d**) step 4: two split droplets are transported to adjacent location. (**e**) step 5: two droplets are merged again. (**f**) experimental demonstration of a mixed droplet

#### 1.3.3.2 Lamination Mixing

Droplet mixing can be achieved by transporting two droplets across an electrode array into an electrode. A split-and-recombination (SAR) lamination is proposed for MEDA biochips to achieve fast mixing using as small an area as possible. SAR lamination mixing can effectively exploit the advantages of low-Reynolds-number fluidics [24].

There are three basic steps for SAR lamination mixing: (1) fluid-element splitting, (2) recombination, and (3) rearrangement. After one SAR cycle, two lamellae can be split into four lamellae. Figure 1.14 illustrates a $4 \times 4$ lamination mixer on the MEDA biochip. Droplets are merged and splitted multiple cycles to achieve SAR lamination mixing.

## 1.4 Design Automation and Test Techniques for MEDA Biochips

In this section, we describe the computer-aided design (CAD) flow and test techniques for conventional DMFBs, and provide the motivation for developing design automation and test techniques for MEDA biochips.

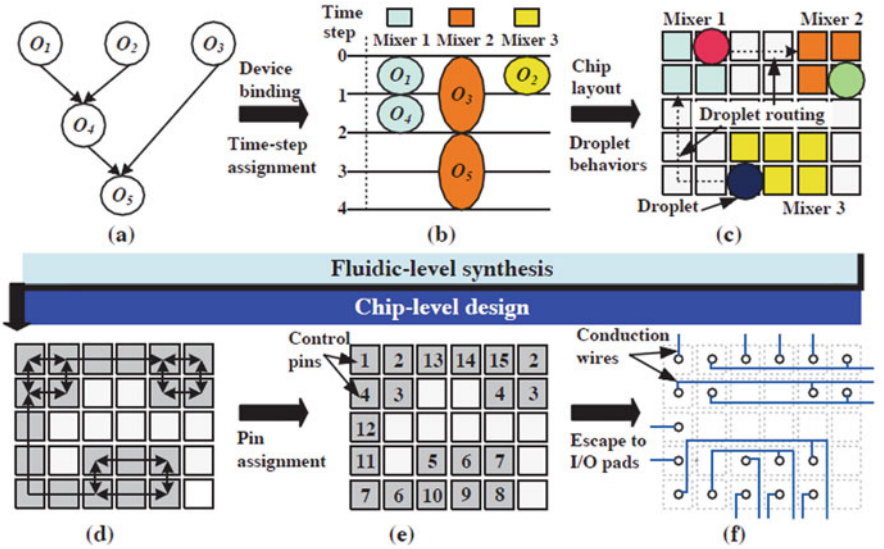



**Fig. 1.15** Illustration of state-of-the-art CAD flow for DMFBs [31]. (**a**) Sequencing graph. (**b**) Scheduled result. (**c**) Placement and routing. (**d**) Used electrodes. (**e**) Electrode addressing. (**f**) Wiring solution

### *1.4.1 CAD Flow and Test Techniques*

Design-automation tools can significantly reduce the difficulty of biochip design and help to ensure that manufactured biochips are versatile and reliable. With the help of these CAD tools, biochip users, e.g., analytical chemists, nurses, doctors and clinicians, will be able to better exploit biochip benefits, and biochip designers can be freed from cumbersome and labor-intensive tasks.

An illustration of state-of-the-art automated DMFB design flow is shown in Fig. 1.15. As shown in Fig. 1.15, a regular CAD flow of DMFB consists of two stages: *fluidic-level synthesis* and *chip-level design*. Fluidic-level synthesis consists of two major phases, namely architectural-level synthesis (i.e., high-level synthesis) and geometry-level synthesis (i.e., physical design) [32, 33]. Fluidic-level synthesis helps to generate an automated scheduling of bioassays and a generation of a mapping of bioassay operations to a DMFB. On the other hand, chip-level design determines the control-signal plan and electrical connections for the electrodes to execute the synthesized result. Chip-level design consists of the steps of electrode addressing followed by wire routing [31, 34]. Electrode-addressing methods allow a single pin to control multiple electrodes, i.e., the control of microfluidic arrays can be achieved using a small number of pins [35, 36]. Wire routing routes conduction wires from the topside electrode array to the surrounding pads.

Recent advances in the testing of DMFBs include fault modeling, structural test, and functional test [37]. In order to facilitate the detection of defects, fault

models are developed to efficiently represent the effect of physical defects at some level of abstraction. Structural test methods use test droplets to traverse the target microfluidic array to detect and locate defect sites by tracking droplet motion [38–42]. On the other hand, functional testing involves test procedures to check whether fluidic modules can be used to correctly perform certain operations, e.g., droplet mixing and splitting [43–45].

### *1.4.2 Motivation for MEDA-Specific Research*

As mentioned above, there has been a considerable amount of work on design automation and test techniques for DMFBs [31, 34, 40, 46–54]. However, due to the inherent differences between today's DMFBs and MEDA architecture, most of existing design automation and test techniques cannot be directly utilized for MEDA biochips or cannot fully exploit MEDA benefits. Here we list some of these differences.

- Compared to conventional DMFBs, fluidic operations on a MEDA biochip, e.g., lamination mixing [24], can be completed in much less time. Hence, in contrast to assumptions made in prior work [46, 55], droplet routing time is no longer negligible and it needs to be considered.
- MEDA introduces the diagonal movement of droplets, which provides greater degree of freedom compared to traditional DMFBs where droplet movement is limited to the "x" and "y" directions. This degree of freedom can be exploited for more efficient bioassay execution.
- Due to the integration of active CMOS logic, sensing can be carried out anywhere on a MEDA biochip [26], and the response time for sensing (10 ms [27]) is much smaller compared to that needed on a conventional DMFB (e.g., 30 s, as reported in [55]). A limitation of today's DMFBs is that sensors can only be integrated in specific areas of the biochip and there can only be a few such sensors. MEDA allows far more flexibility and a sensor can be integrated below each electrode.
- MEDA permits precise and flexible control of both droplet size and shape. Hence, unlike in traditional methods, droplet size-aware synthesis is necessary for MEDA biochips.

In order to target MEDA designs, and in contrast to previous design-automation and test techniques, the design automation techniques for MEDA biochips should incorporate specific characteristics of MEDA biochips, e.g., manipulating droplets of various sizes, diagonal movement of droplets in a two-dimensional array, and the impact of droplet-transportation time on synthesis. Similarly, test techniques should also address MEDA-specific operations, such as diagonal splitting and real-time droplet sensing, which cannot be tested using previous test methods for conventional DMFBs. Therefore, special design, optimization, and test techniques must be developed to fully exploit the advantages offered by the MEDA architecture.

## 1.5 Outline of the Book

This book addresses the design automation and test techniques for MEDA biochips and it is motivated by the MEDA architecture and MEDA-specific characteristics. The book is organized in six chapters. An overview of these chapters is as follows:

Chapter 2 presents a high-level synthesis technique for MEDA biochips. The proposed technique performs scheduling, placement, and droplet size-aware routing in a unified manner. The synthesis approach is able to route sample droplets of different sizes and it incorporates new MEDA-specific fluidic operations such as diagonal droplet motion. An analytical model has also been presented for droplet velocity and the model is experimentally validated using a fabricated MEDA biochip.

Chapter 3 presents the first error-recovery strategy specific for MEDA biochips. The chapter first presents 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. Different probabilistic timed automata (PTA)-based error-recovery strategies for various types of local errors are then presented. An on-line synthesis technique and a control flow are finally proposed to connect local-recovery procedures with global error recovery for the complete bioassay.

Chapter 4 focuses on the development of test techniques for MEDA biochips. Efficient structural test techniques are first proposed to effectively detect defects in a MEDA biochip and identify faulty microcells. Functional test techniques are then proposed to address fundamental MEDA operations, such as droplet dispensing, transportation, mixing, and splitting. Simulation results and experiments using fabricated MEDA biochips highlight the effectiveness of the proposed test techniques.

An sample preparation method for MEDA biochips, referred to as weighted sample preparation method (WSPM), is presented in Chap. 5. Two new MEDA-enabled operations for sample preparation have been proposed and experimentally demonstrated on a fabricated MEDA biochip. In order to tolerate the inherent uncertainty of fluidic operations, we have also proposed a hybrid error-recovery technique integrated with WSPM by exploiting MEDA-specific advances in real-time droplet sensing. Experimental demonstration using a fabricated MEDA biochip and simulation results highlight the effectiveness of the proposed sample-preparation method.

Finally, Chap. 6 summarizes the contributions of the book and identifies directions for future work.

## References

1. Schulte, T. H., Bardell, R. L., & Weigl, B. H. (2002). Microfluidic technologies in clinical diagnostics. *Clinica Chimica Acta, 321*(1), 1–10.
2. Guiseppi-Elie, A., Brahim, S., Slaughter, G., & Ward, K. R. (2005). Design of a subcutaneous implantable biochip for monitoring of glucose and lactate. *IEEE Sensors Journal, 5*(3), 345–355.

3. Fair, R. B. (2007). Digital microfluidics: Is a true lab-on-a-chip possible? *Microfluidics and Nanofluidics, 3*(3), 245–281.
4. Samiei, E., Tabrizian, M., & Hoorfar, M. (2016). A review of digital microfluidics as portable platforms for lab-on a-chip applications. *Lab on a Chip, 16*(13), 2376–2396.
5. Millington, D., Norton, S., Singh, R., Sista, R., Srinivasan, V., & Pamula, V. (2018). Digital microfluidics comes of age: High-throughput screening to bedside diagnostic testing for genetic disorders in newborns. *Expert Review of Molecular Diagnostics, 18*, 701–712.
6. Luo, Y., Chakrabarty, K., & Ho, T.-Y. (2015). *Hardware/Software co-design and optimization for cyberphysical integration in digital microfluidic biochips*. Cham: Springer.
7. Zhao, Y., & Chakrabarty, K. (2012). *Design and testing of digital microfluidic biochips*. New York: Springer Science and Business Media.
8. Chakrabarty, K. (2010). Design automation and test solutions for digital microfluidic biochips. *IEEE Transactions on Circuits and Systems I: Regular Papers, 57*(1), 4–17.
9. Berthier, J. (2012). *Micro-Drops and Digital Microfluidics*. Waltham: William Andrew.
10. Hu, K., Ibrahim, M., Chen, L., Li, Z., Chakrabarty, K., & Fair, R. (2015). Experimental demonstration of error recovery in an integrated cyberphysical digital-microfluidic platform. In *IEEE Biomedical Circuits and Systems Conference*, pp. 1–4.
11. Srinivasan, V., Pamula, V. K., & Fair, R. B. (2004). Droplet-based microfluidic lab-on-a-chip for glucose detection. *Analytica Chimica Acta, 507*(1), 145–150.
12. Chen, X., Cui, D., Liu, C., Li, H., & Chen, J. (2007). Continuous flow microfluidic device for cell separation, cell lysis and DNA purification. *Analytica Chimica Acta, 584*(2), 237–243.
13. Chang, Y.-H., Lee, G.-B., Huang, F.-C., Chen, Y.-Y., Lin, J.-L. (2006). Integrated polymerase chain reaction chips utilizing digital microfluidics. *Biomedical Microdevices, 8*(3), 215–225.
14. Luan, L., Evans, R. D., Jokerst, N. M., & Fair, R. B. (2008). Integrated optical sensor in a digital microfluidic platform. *IEEE Sensors Journal, 8*(5), 628–635.
15. Hu, K., Hsu, B.-N., Madison, A., Chakrabarty, K., & Fair, R. (2013). Fault detection, real-time error recovery, and experimental demonstration for digital microfluidic biochips. In *Proceedings of IEEE/ACM Design, Automation and Test Conference in Europe*, pp. 559–564.
16. Pollack, M., Shenderov, A., & Fair, R. (2002). Electrowetting-based actuation of droplets for integrated microfluidics. *Lab on a Chip, 2*(2), 96–101.
17. Choi, K., Ng, A. H., Fobel, R., & Wheeler, A. R. (2012). Digital microfluidics. *Annual Review of Analytical Chemistry, 5*, 413–440.
18. Cho, S. K., Moon, H., & Kim, C.-J. (2003). Creating, transporting, cutting, and merging liquid droplets by electrowetting-based actuation for digital microfluidic circuits. *Journal of Microelectromechanical Systems, 12*(1), 70–80.
19. Mugele, F., & Baret, J.-C. (2005). Electrowetting: From basics to applications. *Journal of Physics: Condensed Matter, 17*(28), R705.
20. Berthier, J., Clementz, P., Raccurt, O., Jary, D., Claustre, P., Peponnet, C., et al. (2006). Computer aided design of an EWOD microdevice. *Sensors and Actuators A: Physical, 127*(2), 283–294.
21. Wulff-Burchfield, E., Schell, W. A., Eckhardt, A. E., Pollack, M. G., Hua, Z., Rouse, J. L., et al. (2010). Microfluidic platform versus conventional real-time polymerase chain reaction for the detection of Mycoplasma pneumoniae in respiratory specimens. *Diagnostic Microbiology and Infectious Disease, 67*(1), 22–29.
22. Jebrail, M. J., & Wheeler, A. R. (2008). Digital microfluidic method for protein extraction by precipitation. *Analytical Chemistry, 81*(1), 330–335.
23. Sista, R. S., Eckhardt, A. E., Srinivasan, V., Pollack, M. G., Palanki, S., & Pamula, V. K. (2008). Heterogeneous immunoassays using magnetic beads on a digital microfluidic platform. *Lab on a Chip, 8*(12), 2188–2196.
24. Wang, G., Teng, D., & Fan, S.-K. (2011). Digital microfluidic operations on micro-electrode dot array architecture. *IET Nanobiotechnology, 5*(4), 152–160.
25. Lai, K. Y.-T., Shiu, M.-F., Lu, Y.-W., Ho, Y.-C., Kao, Y.-C., Yang, Y.-T., et al. (2015). A field-programmable lab-on-a-chip with built-in self-test circuit and low-power sensor-fusion solution in 0.35 $\mu$m standard CMOS process. In *Proceedings of IEEE Asian Solid-State Circuits Conference*, pp. 1–4.

26. Wang, G., Teng, D., Lai, Y.-T., Lu, Y.-W., Ho, Y., & Lee, C.-Y. (2013). Field-programmable lab-on-a-chip based on microelectrode dot array architecture. *IET Nanobiotechnology, 8*(3), 163–171.
27. Lai, K. Y.-T., Yang, Y.-T., & Lee, C.-Y. (2015). An intelligent digital microfluidic processor for biomedical detection. *Journal of Signal Processing Systems, 78*(1), 85–93.
28. Li, Z., Lai, K. Y.-T., Yu, P.-H., Chakrabarty, K., Ho, T.-Y., & Lee, C.-Y. (2017). Droplet size-aware high-level synthesis for micro-electrode-dot-array digital microfluidic biochips. *IEEE Transactions on Biomedical Circuits and Systems, 11*(3), 612–626.
29. Li, Z., Lai, K. Y.-T., McCrone, J., Yu, P.-H., Chakrabarty, K., Pajic, M., et al. (2018). Efficient and adaptive error recovery in a micro-electrode-dot-array digital microfluidic biochip. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 37*(3), 601–614.
30. Chen, Z., Teng, D. H.-Y., Wang, G. C.-J., & Fan, S.-K. (2011). Droplet routing in high-level synthesis of configurable digital microfluidic biochips based on microelectrode dot array architecture. *Biochip Journal, 5*(4), 343–352.
31. Ho, T.-Y., Zeng, J., & Chakrabarty, K. (2010). Digital microfluidic biochips: A vision for functional diversity and more than moore. In *Proceedings of IEEE/ACM International Conference on Computer-Aided Design*, pp. 578–585.
32. Su, F., & Chakrabarty, K. (2004). Architectural-level synthesis of digital microfluidics-based biochips. In *Proceedings of IEEE/ACM International Conference on Computer-Aided Design*, pp. 223–228.
33. Su, F., Chakrabarty, K., & Fair, R. B. (2006). Microfluidics-based biochips: Technology issues, implementation platforms, and design-automation challenges. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 25*(2), 211–223.
34. Huang, T.-W., Yeh, S.-Y., & Ho, T.-Y. (2011). A network-flow based pin-count aware routing algorithm for broadcast-addressing EWOD chips. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 30*(12), 1786–1799.
35. Zhao, Y., Xu, T., & Chakrabarty, K. (2011). Broadcast electrode-addressing and scheduling methods for pin-constrained digital microfluidic biochips. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 30*(7), 986–999.
36. Xu, T., & Chakrabarty, K. (2008). Broadcast electrode-addressing for pin-constrained multi-functional digital microfluidic biochips. In *Proceedings of ACM/IEEE Design Automation Conference*, pp. 173–178.
37. Shukla, V., Hussin, F. A., Hamid, N. H., & Zain Ali, N. B. (2017). Advances in testing techniques for digital microfluidic biochips. *Sensors, 17*(8), 1719.
38. Mitra, D., Ghoshal, S., Rahaman, H., Chakrabarty, K., & Bhattacharya, B. B. (2011). Test planning in digital microfluidic biochips using efficient eulerization techniques. *Journal of Electronic Testing Theory and Application, 27*, 657–671.
39. Li, Z., Dinh, T. A., Ho, T.-Y., & Chakrabarty, K. (2014). Reliability-driven pipelined scan-like testing of digital microfluidic biochips. In *Proceedings of IEEE Asian Test Symposium*, pp. 57–62.
40. Dinh, T. A., Yamashita, S., Ho, T.-Y., & Chakrabarty, K. (2015). A general testing method for digital microfluidic biochips under physical constraints. In *Proceedings of IEEE International Test Conference*, pp. 1–8.
41. Mukherjee, S., & Samanta, T. (2015). Distributed scan like fault detection and test optimization for digital microfluidic biochips. *Journal of Electronic Testing, 31*(3), 311–319.
42. Bhattacharjee, S., Mitra, D., & Bhattacharya, B. B. (2017). Robust in-field testing of digital microfluidic biochips. *ACM Journal on Emerging Technologies in Computing Systems, 14*(1), 5.
43. Xu, T., & Chakrabarty, K. (2007). Functional testing of digital microfluidic biochips. In *Proceedings of IEEE International Test Conference*, pp. 1–10.
44. Mitra, D., Ghoshal, S., Rahaman, H., Bhattacharya, B. B., Majumder, D. D., & Chakrabarty, K. (2008). Accelerated functional testing of digital microfluidic biochips. In *IEEE Asian Test Symposium*, pp. 295–300.

45. Xu, T., & Chakrabarty, K. (2009). Fault modeling and functional test methods for digital microfluidic biochips. *IEEE Transactions on Biomedical Circuits and Systems, 3*, 241–253.
46. Maftei, E., Pop, P., & Madsen, J. (2009). Tabu search-based synthesis of dynamically reconfigurable digital microfluidic biochips. In *Proceedings of International Conference on Compilers, Architecture, and Synthesis for Embedded Systems*, pp. 195–204.
47. Grissom, D., & Brisk, P. (2012). Path scheduling on digital microfluidic biochips. In *Proceedings of ACM/IEEE Design Automation Conference*, pp. 26–35.
48. Zhao, Y., & Chakrabarty, K. (2012). Cross-contamination avoidance for droplet routing in digital microfluidic biochips. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 31*(6), 817–830.
49. Grissom, D., & Brisk, P. (2012). Fast online synthesis of generally programmable digital microfluidic biochips. In *Proceedings of IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis*, pp. 413–422.
50. Roy, S., Chakrabarti, P. P., Kumar, S., Chakrabarty, K., & Bhattacharya, B. B. (2015). Layout-aware mixture preparation of biochemical fluids on application-specific digital microfluidic biochips. *ACM Transactions on Design Automation of Electronic Systems, 20*(3), 45.
51. Yu, S.-T., Yeh, S.-H., & Ho, T.-Y. (2015). Reliability-driven chip-level design for high-frequency digital microfluidic biochips. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 34*(4), pp. 529–539.
52. Keszocze, O., Wille, R., Ho, T.-Y., & Drechsler, R. (2014). Exact one-pass synthesis of digital microfluidic biochips. In *Proceedings of ACM/IEEE Design Automation Conference*, pp. 1–6.
53. Grissom, D., Curtis, C., Windh, S., Phung, C., Kumar, N., Zimmerman, Z., et al. (2015). An open-source compiler and PCB synthesis tool for digital microfluidic biochips. *Integration, the VLSI Journal, 51*, 169–193.
54. Jaress, C., Brisk, P., & Grissom, D. (2015). Rapid online fault recovery for cyber-physical digital microfluidic biochips. In *Proceedings of IEEE VLSI Test Symposium*, pp. 1–6.
55. Xu, T., & Chakrabarty, K. (2007). Integrated droplet routing in the synthesis of microfluidic biochips. In *ACM/IEEE Design Automation Conference*, pp. 948–953.

# Chapter 2
# Droplet Size-Aware High-Level Synthesis


## 2.1 Chapter Overview

High-level synthesis for digital microfluidic biochips (DMFBs) can be viewed as the problem of scheduling assay operations, binding operations to a given number of resources, configuring operations on the chip, and routing droplets to their destinations. The objective of high-level synthesis is to maximize parallelism, thereby decreasing completion time of bioassays.

In this chapter, we present the first high-level synthesis approach that can be used for MEDA biochips. Different with previous methods, droplet routing is an important consideration for biochip synthesis. In order to obtain synthesis results that accurately predict bioassay operation times, droplet routing needs to be seamlessly integrated in the synthesis flow. Compared to conventional DMFBs, routing complexity increases with an increase in the complexity of a MEDA biochip. The concurrent manipulation of multiple droplets with different sizes and advanced fluidic operations specific to the MEDA architecture (e.g., shape morphing and diagonal movement) further increases routing complexity. Therefore, in order to target MEDA designs, and in contrast to previous synthesis techniques, the proposed method incorporates the routing of droplets of various sizes, droplet shape morphing, diagonal movement of droplets in a two-dimensional array, and the impact of droplet-transportation time on synthesis.

The remainder of the chapter is organized as follows. Section 2.2 presents the motivation of high-level synthesis for MEDA biochips and a formal statement of the optimization problem. Section 2.3 presents an experimentally validated velocity model for droplets of different sizes. Section 2.4 describes the synthesis flow in more detail. Section 2.5 presents simulation results for three real-life laboratory protocols and experimental results for a fabricated chip. Finally, Sect. 2.6 concludes the chapter.



Z. Li et al., *Micro-Electrode-Dot-Array Digital Microfluidic Biochips*,
https://doi.org/10.1007/978-3-030-02964-7_2

## 2.2 Motivation and Problem Formulation

In this section, the motivation of high-level synthesis specific for MEDA biochips is first discussed. A formal problem formulation is then presented for the MEDA synthesis problem.

### *2.2.1 Motivation of High-Level Synthesis for MEDA*

The problem of biochip synthesis can be viewed as the problem of scheduling assay functions and binding them to a given number of resources so as to maximize parallelism, thereby decreasing the response time [1].

There have been a considerable amount of work on high-level synthesis for conventional DMFBs [2–9]. The bioassay as the input to biochip synthesis tool is usually modeled by a sequencing graph [10], which denotes the dependencies between fluidic operations. An example of a sequencing graph for the bioassay of multiplexed in-vitro diagnostics is shown in Fig. 2.1. A synthesis tool maps a given bioassay to the patterned electrode array of digital microfluidics, binds assay operation to on-chip resources, generates an optimized schedule of fluidic operations, and computes droplet transportation routes. Goals for biochip synthesis usually include (1) the minimization of bioassay completion time, (2) the minimization of chip area, and (3) the higher defect tolerance.

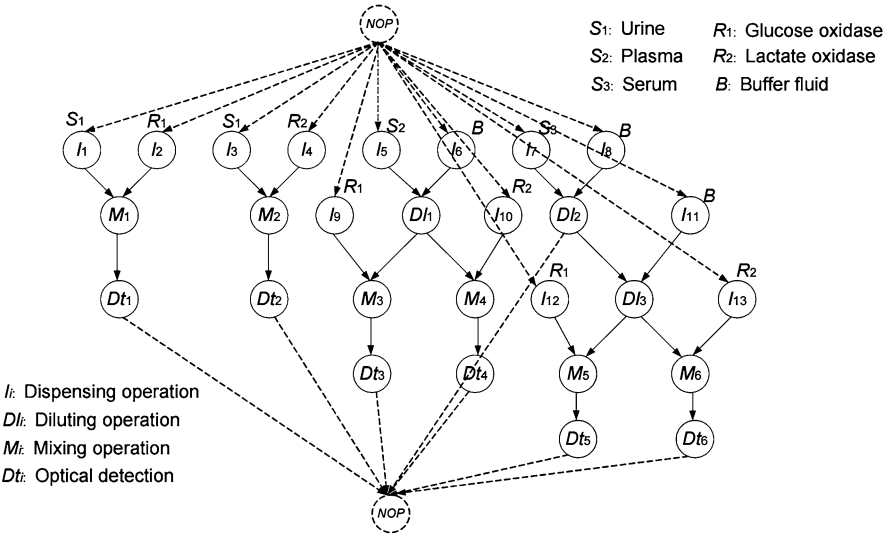



**Fig. 2.1** Sequencing graph for multiplexed in-vitro diagnostics [2]

More specifically, there are four major tasks in biochip synthesis: (1) resource binding, (2) operation scheduling, (3) module placement, and (4) droplet routing [11]. The first task, resource binding, refers to the mapping from bioassay operations to available fluidic modules. Note that there might be several modules available for a given operation. For example, either a 2×2-array mixer or a 2×3-array mixer can be utilized for a droplet mixing operation. However, different modules for a given operation may result in different operation completion times. Similarly, a fluidic module can also be associated with multiple assay operations, which necessitates resource sharing. Once the resource binding is completed, operation scheduling can be performed to determine the start and stop times of all fluidic operations, subject to both the precedence and resource-sharing constraints [12]. There are four major operation scheduling algorithms that have been proposed for biochips: modified list scheduling (MLS) [10], an optimal scheduler based on integer linear programming (ILP) [10], and two genetic algorithms [13]. More recently, a resource-constrained scheduling algorithm has also been presented based on force-directed list scheduling [9].

After the operation scheduling is completed, the key problem is the placement of fluidic modules (e.g., different types of mixers, diluters, and splitters). Since DMFBs enable dynamic reconfiguration of fluidic modules during run-time, different modules can be placed on the same location during different time intervals. Module placement problem on a DMFB has been proved to be a NP-complete problem [3]. A simulated annealing (SA)-based heuristic approach is first proposed to solve the problem in a computationally efficient manner [3]. The work in [2] then proposed a unified high-level synthesis technique for DMFBs based on parallel recombinative simulated annealing (PRSA), which offers a link between operation scheduling and module placement. The work in [7] proposed the first reliability-oriented non-SA placement algorithm for DMFBs, which utilizes the 3-D deferred decision making technique to enumerate only possible placement solutions to reduce computation complexity. If the chip has not been fabricated, solutions for the placement problem can provide chip designers with the information on the size of the array to be manufactured. If the chip has been fabricated, optimized module placement help with area minimization to free up more unit cells for defect tolerance [1].

Even though there has been a large amount of work on high-level synthesis for conventional DMFBs, existing synthesis solutions cannot be directly utilized for MEDA biochips because of the inherent differences between conventional DMFBs and MEDA. Some of these differences are listed below.

- Compared to conventional DMFBs, fluidic operations on a MEDA-based biochip, e.g., lamination mixing [14], can be completed in much less time. Hence, in contrast to assumptions made in prior work on high-level synthesis [4, 5], droplet routing time is no longer negligible and it needs to be carefully considered during the scheduling of fluidic operations.
- There are some novel fluidic operations that are specific to the MEDA architecture. An example of such an operation is the diagonal movement of droplets, which provides greater degree of freedom compared to traditional DMFBs where

droplet movement can only happen along the "x" and "y" directions in a two-dimensional array. This degree of freedom can be exploited in MEDA biochips for more efficient bioassay execution.

- Due to the integration of active CMOS logic, droplet sensing (or real-time detection) can be carried out anywhere on a MEDA-based biochip [15], and response time for sensing (e.g., 10 ms [16]) is much smaller compared to the detection time (e.g., 30 s [4]) needed on a conventional DMFB. A major limitation of today's DMFBs is that sensors can only be integrated in specific areas of the biochip and there can only be a few such sensors on a chip. As described in Chap. 1, a MEDA-based biochip allows a sensor to be integrated below each electrode.
- MEDA permits precise and flexible control of the droplet size and shape. Hence, unlike in traditional high-level synthesis methods, the size of the droplets can no longer be treated as an abstract quantity. Droplet size-aware synthesis is necessary for MEDA-based biochips.

In order to fully adapt to MEDA architecture, this chapter proposes a biochip synthesis approach that can be used for MEDA biochips. In contrast to previous synthesis techniques for conventional DMFBs, the proposed method incorporates the routing of droplets of various sizes, diagonal movement of droplets in a two-dimensional array, and the impact of droplet-transportation time on synthesis.

### 2.2.2 Problem Formulation

The problem of high-level synthesis for MEDA biochips is formulated as follows. We are given a reconfigurable $W \times H$ MEDA biochip ($W$ and $H$ represents the chip width and height in microelectrodes, respectively). Each microelectrode is identified by its coordinate $(x, y)$, with the origin $(0, 0)$ at the bottom-left corner.

There are four inputs to the synthesis problem: (1) The sequencing graph for $G = \{V, E\}$ for the bioassay, where $V = \{V_1, V_2, \ldots, V_m\}$ represents $m$ operations and $E = \left\{\left(V_i, V_j\right)\ 1 \leq i, j \leq m\right\}$ represents the data dependencies between all pairs of operations $i$ and $j$, i.e., the precedence relationships; (2) The digital microfluidic library, which includes the type, size, and corresponding execution time of on-chip fluidic functional modules; (3) The size of droplets dispensed from on-chip reservoirs; (4) The MEDA biochip specifications (e.g., the chip size, the thickness of each layer in the MEDA biochip, and the gap between top and bottom plates).

Outputs include the mapping of the fluidic operations to resources in the digital microfluidic library, the module placement for each fluidic operation, the schedule for the assay operations, and the detailed droplet routes (transportation pathways).

The objective is to minimize the bioassay completion time subject to the precedence constraints imposed by the sequencing graph for a given MEDA biochip.

## 2.3 Droplet Velocity Model

As discussed in Sect. 2.2, droplet transportation time needs to be carefully considered during MEDA synthesis. Droplet transportation time is mainly constrained by the droplet velocity. Because the MEDA architecture can manipulate droplets with different sizes and shapes and droplets can move diagonally on a MEDA biochip, a synthesis algorithm for MEDA must consider the impact of droplet size and diagonal movement on droplet velocity.

Therefore, an analytical model is developed for MEDA biochips to determine the impact of droplet size and diagonal movement on velocity. This model forms a key part of the size-aware droplet routing method of the synthesis algorithm presented in Sect. 2.4.4.

The velocity model describes the dynamic response of a single droplet based on the forces acting on the droplet [17, 18]. The droplet is considered as a single mass, and the droplet is assumed to be moving with an average velocity.

The EWOD-induced droplet motion depends on the actuation force and the forces that oppose droplet motion. According to the energy-based analysis proposed in [17, 19], the EWOD force on the droplet can be calculated using the differential of the effective area. In addition, a threshold EWOD voltage [20] is also required to overcome contact-angle hysteresis.

The forces resisting droplet motion arise from different sources, and some of these forces are nonlinear functions of droplet velocity. The three major sources of resistance reported in the literature [19] are (1) viscous dissipation within the droplet, (2) viscous drag due to droplet movement through the filler fluid, and (3) slippage of the contact-line between the droplet and hydrophobic surface.

The nature and dynamics of several of these opposing forces are not well-understood, which complicates the development of a model that takes these forces into account. However, these phenomena have been successfully addressed using semi-empirical models in the literature [17]. In the model presented in this chapter, each opposing force is represented by simplified algebraic expressions, and the calculation of each force is presented in the following subsection.

### *2.3.1 Model Construction*

In this subsection, we present the calculation of each force for a $m \times n$ droplet (a droplet occupying $m \times n$ microelectrode array) moving long each direction (e.g., x-axis direction, y-axis direction, and diagonal direction) on a MEDA biochip.

*EWOD Force* The actuation force on a droplet during the transition from a group of microelectrodes to an adjacent area can be modeled by analyzing the droplet energy as a function of droplet position. Here, the position of the droplet is tracked by its

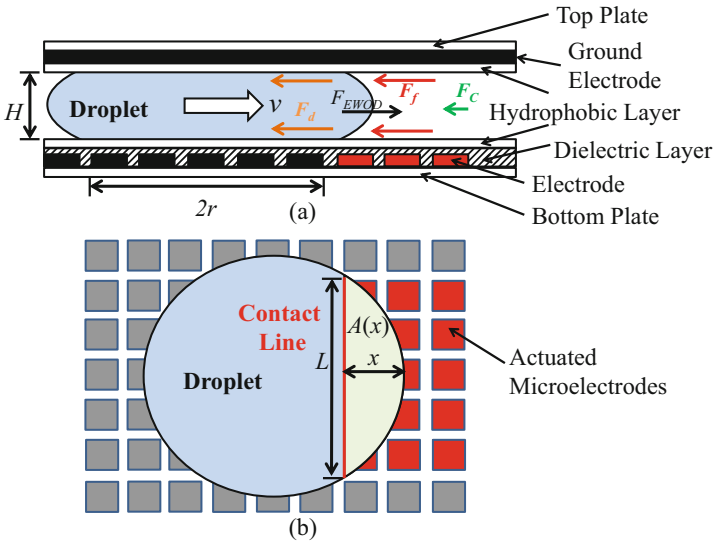



**Fig. 2.2** A droplet undergoing transport on a MEDA biochip: (**a**) side view, and (**b**) top view. The droplet is moving along the x-axis, therefore the position of a droplet can be tracked by the x-coordinate

x-coordinate; see Fig. 2.2. Figure 2.2a and b present the side and top views of an idealized droplet transition, respectively. From [19, 20], the EWOD force on the droplet is given by (2.1):

$$F_{EWOD} = \frac{C_{unit} V^2}{2} \frac{dA(x)}{dx} = \frac{C_{unit} V^2}{2} L_{eff} \tag{2.1}$$

where $V$ is the applied voltage, $x$ is the droplet position, $C_{unit}$ is the structural capacitance per unit area in the dielectric layer, $A(x)$ is the area of the droplet over the activated group of microelectrodes, and $L_{eff}$ is the length of the contact line. Here we assume $L_{eff}$ is constant during droplet movement. Since the dielectric layer for the MEDA biochip is a composition of two layers (a $SiO_2$ layer and a $Si_3N_4$ layer) [21], the unit capacitance $C_{unit}$ can be calculated using (2.2):

$$C_{unit} = \frac{\varepsilon_0}{d_{SiO_2}/\varepsilon_{SiO_2} + d_{Si_3N_4}/\varepsilon_{Si_3N_4}} \tag{2.2}$$

where $d_{SiO_2}$ is the thickness of the $SiO_2$ layer, $d_{Si_3N_4}$ is the thickness of the $Si_3N_4$ layer, $\varepsilon_{SiO_2}$ is the dielectric constant of the $SiO_2$ layer, $\varepsilon_{Si_3N_4}$ is the dielectric constant of the $Si_3N_4$ layer, and $\varepsilon_0$ is the permittivity of vacuum.

When an $m \times n$ droplet is transported from its current location to an adjacent location, $L_{eff}$ is (1) $nd$, (2) $md$, and (3) $\sqrt{(m-1)^2 + (n-1)^2}d$ when droplet is transporting along (1) x-axis direction, (2) y-direction, and (3) diagonal direction,

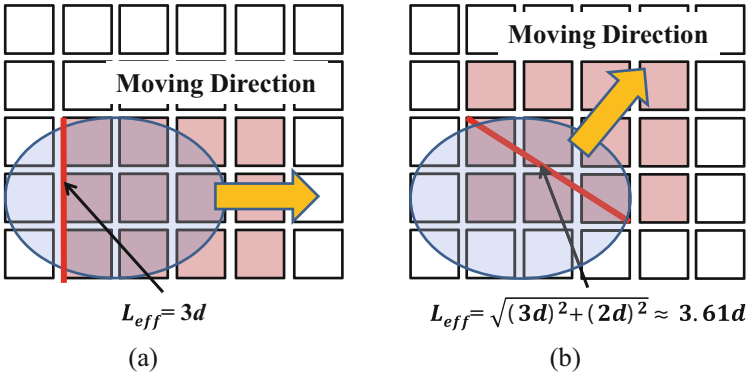



**Fig. 2.3** Illustration of a $4 \times 3$ droplet moving along (**a**) x-axis direction, and (**b**) diagonal direction

respectively (the parameter $d$ refers to the microelectrode pitch). An example of a $4 \times 3$ droplet moving to its adjacent location is shown in Fig. 2.3.

*Shear Force* If we assume a velocity profile in the droplet with zero-slip boundary conditions at the top and bottom plates [17, 18], a conservative estimate for the shear stress can be obtained by assuming a parabolic velocity profile. The total shear force exerted by the top and bottom plates can be obtained using (2.3):

$$F_d = \left(\frac{6u_d v}{H}\right)(2S) = \left(\frac{6u_d v}{H}\right)(2mnd^2) \tag{2.3}$$

where $v$ is the droplet velocity, $H$ is the gap between two plates, $S$ is the area occupied by the $m \times n$ droplet, and $u_d$ is the droplet viscosity.

*Viscous Drag Force* Assuming that the droplet is moving through the filler-fluid as a rigid body, the viscous drag is estimated by (2.4).

$$F_f = 0.5\left(C_D \rho_f v^2\right)\left(L_{eff} H\right) \tag{2.4}$$

where $C_D$ is the drag coefficient for a cylinder in a cross flow and $\rho_f$ is the filler fluid density.

*Contact-Line Friction Force* The contact-line friction force is estimated using (2.5).

$$F_c = 2(m+n)\zeta v \tag{2.5}$$

where $\zeta$ is a proportionality coefficient and the friction force is assumed to be proportional to the droplet velocity [20].

**Table 2.1** Parameters used for model validation

| Parameter | Value |
|---|---|
| $SiO_2$ layer thickness, $d_{SiO_2}$ | $450 \times 10^{-9}$ m |
| $Si_3N_4$ layer thickness, $d_{Si_3N_4}$ | $1 \times 10^{-6}$ m |
| Dielectric constant of $SiO_2$ layer, $\varepsilon_{SiO_2}$ | 3.9 |
| Dielectric constant of $Si_3N_4$ layer, $\varepsilon_{Si_3N_4}$ | 7.5 |
| Vacuum permittivity, $\varepsilon_0$ | $8.854 \times 10^{-12}$ $C^2N^{-1}M^{-2}$ |
| Droplet viscosity, $\mu_d$ | $1.9 \times 10^{-3}$ Pa·s |
| Droplet density, $\rho_d$ | $1000\,kg/m^3$ |
| Filler-fluid (silicone oil) viscosity, $\mu_f$ | $1.7 \times 10^{-3}$ Pa·s |
| Filler-fluid density, $\rho_f$ | $760\,kg/m^3$ |
| Microelectrode (cell) pitch, $d$ | $50 \times 10^{-6}$ m |
| Plate spacing, $H$ | $50 \times 10^{-6}$ m |
| Coefficient of drag, $C_D$ | 30 |
| Contact-line friction coefficient, $\zeta$ | $0.04\,N \cdot s/m^2$ |

Based on the above analysis, the equation governing the droplet motion can be written as in (2.6), where $m$ is the mass of the droplet and $F_{th}$ is the threshold force to incorporate the consideration of contact angle hysteresis as described in [20].

$$m\frac{dv}{dt} = F_{EWOD} - F_d - F_f - F_c - F_{th} \tag{2.6}$$

### 2.3.2 Model Validation

The above velocity model was validated experimentally on a MEDA biochip. The parameters used for model validation, listed in Table 2.1, are based on the fabricated MEDA biochip and the characterization done with it.

The velocities for droplets moving vertically (along the y-axis), and ranging in size from $4 \times 4$ to $16 \times 16$ were first calculated. For these droplets, the corresponding actuation/resistive forces are the same when they are transported either vertically or horizontally. Therefore, there is no difference between the vertical and horizontal velocities for these droplets. Then, experiments on $5 \times 5$, $7 \times 7$, $10 \times 10$, $12 \times 12$, and $15 \times 15$ droplets were performed in the laboratory on a $60 \times 30$ fabricated MEDA biochip.

The velocities for droplets moving diagonally, and ranging in size from $4 \times 4$ to $16 \times 16$ were also calculated. Experiments on $5 \times 5$, $8 \times 8$, $10 \times 10$, $12 \times 12$, and $15 \times 15$ droplets were then performed on the MEDA biochip.

A comparison between predicted analytical results and the actual experimental results is shown in Fig. 2.4. As shown in the figure, the analytical and experimental results are highly correlated. Moreover, there is little difference between the diagonal velocities and the corresponding vertical/horizontal velocities.

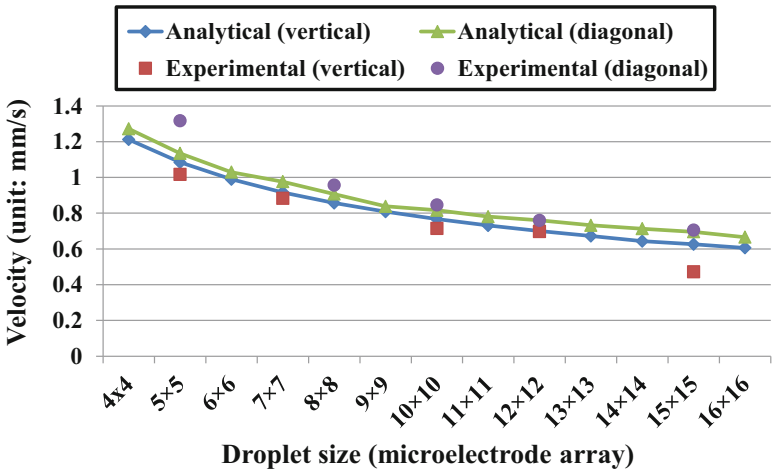



**Fig. 2.4** Comparison between analytical and experimental results

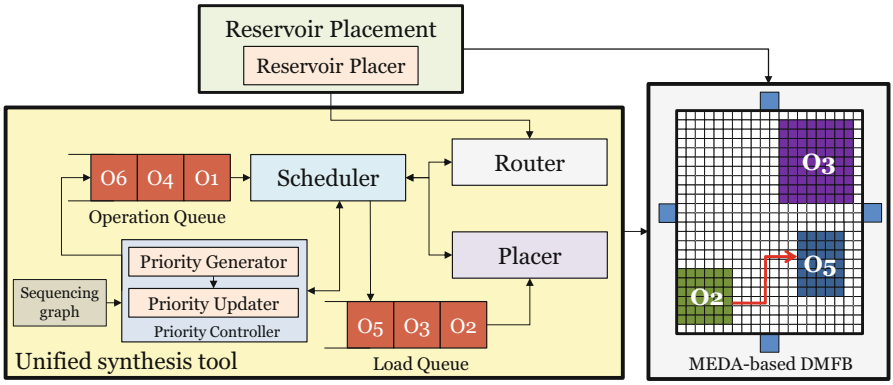



**Fig. 2.5** Unified synthesis flow for MEDA biochips ($O_1$, $O_4$, and $O_6$ represent the operations to be scheduled; $O_2$, $O_3$, and $O_5$ represent the operations that have been scheduled, and the placer will configure these operations on the MEDA biochip)

## 2.4 Unified High-Level Synthesis

In contrast to synthesis methods for conventional DMFBs, we propose a unified synthesis flow that co-optimizes operation scheduling, module placement, and size-aware droplet routing for MEDA biochips as shown in Fig. 2.5. First, the location of peripheral reservoirs is determined by the reservoir placer. After that, the synthesis tool consisting of the priority controller, scheduler, placer, and router is invoked to configure the target bioassay on a MEDA biochip.

The priority controller dynamically generates priorities for a list of operations from the given sequencing graph. The scheduler, placer, and router closely interplay

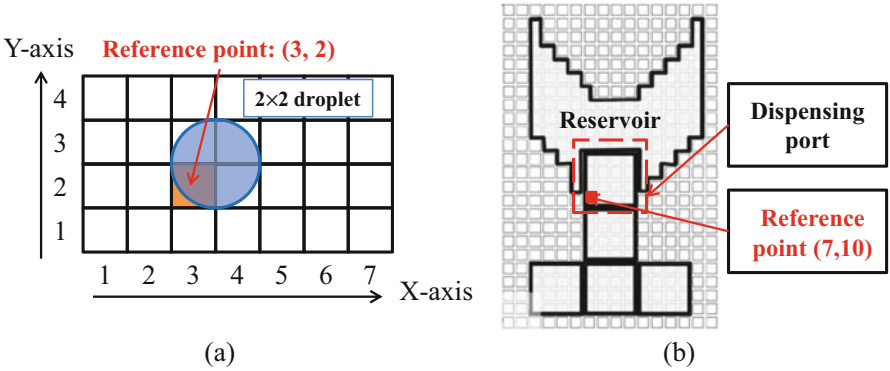



**Fig. 2.6** Illustration of (**a**) a 2×2 droplet located on (3,2) and (**b**) a reservoir located on (7,10)

with each other to determine the start/execution time for each operation, the location of each fluidic module, and droplet pathways between two locations.

The priority controller dynamically generates priorities for a list of operations from the given sequencing graph. The scheduler, placer, and router closely interplay with each other to determine the start/execution time for each operation, the location of each fluidic module, and droplet pathways between two locations.

### *2.4.1 Reservoir Placement*

Since droplets are dispensed from reservoirs, the location of reservoirs on the biochip boundary affects the synthesis results. Furthermore, if reservoirs are close to each other, there might be a conflict, i.e., dispensing cluster, when droplets are dispensed at the same time. Thus, reservoir placement must be carefully considered to avoid potential dispensing clusters and reduce the routing length for subsequent operations.

To tackle this problem, we present an integer linear programming (ILP)-based method to place reservoirs. In the placement model, we use a "reference point" to indicate the location of a droplet. The reference point is the bottom-left point of the droplet. For example, the droplet in Fig. 2.6a occupies a $2 \times 2$ microelectrode array, and the droplet location is (3,2). We can also use the reference point of the dispensing port to represent the location of a reservoir. A reservoir on a MEDA biochip is intrinsically a group of microelectrodes that can hold certain volume of liquid. Droplets are dispensed through a dispensing port. An example is shown in Fig. 2.6b. The red microelectrode is the reference point (bottom-left point) of the dispensing port. The location of the microelectrode is therefore (7,10). Hence, here we used (7,10) to represent the location of the reservoir.

Note that there are two types of reservoirs on a MEDA biochip: (1) functional reservoirs, and (2) waste reservoirs. A functional reservoir is used for storing samples and reagents, and a waste reservoir is used to collect waste droplets. In order to place reservoirs, we need to determine the location of the dispensing port for each reservoir. Here we use Cartesian coordinates to describe the location, i.e., reservoir $R_i$'s location is expressed as $(x_i, y_i)$, where $x_i$ and $y_i$ are integers. For a given $W \times H$ MEDA biochip, we first place the waste reservoir $R_0$ at the center of the bottom side of the biochip, i.e., $R_0$'s location is $(\lceil W/2 \rceil, 0)$. For any functional reservoir $R_i$ $(1 \leq i \leq n)$, variables $x_i$ and $y_i$ are constrained as follows:

$$0 \leq x_i \leq W, \ 0 \leq y_i \leq H \tag{2.7}$$

Since reservoirs are usually placed on the boundary of the chip, variables $x_i$ and $y_i$ $(1 \leq i \leq n)$ need to be further constrained as follows:

$$x_i = 0 \quad \text{OR} \quad x_i = W \quad \text{OR} \quad y_i = 0 \quad \text{OR} \quad y_i = H \tag{2.8}$$

Note that there are (non-linear) logic "OR"s in constraint (2.8), which is not supported by ILP solvers. Accordingly, constraint (2.8) needs to be rewritten to fit ILP formulation. Updated constraints are expressed in (2.9)–(2.11) for $1 \leq i \leq n$.

$$y_i \leq m(1 - t_1), \quad H - y_i \leq m(1 - t_2) \tag{2.9}$$

$$x_i \leq m(1 - t_3), \quad W - x_i \leq m(1 - t_4) \tag{2.10}$$

$$1 \leq t_1 + t_2 + t_3 + t_4 \leq 2 \tag{2.11}$$

where $t_1$, $t_2$, $t_3$, and $t_4$ are binary numbers, and $m$ is a large constant, e.g., $W \times H$. Recall that constraint (2.8) is logically equivalent to the constraint group of (2.9)–(2.11).

As described in [22, 23], in order to avoid fluidic leakage and dispensing clustering, the distance between two reservoirs should be no less than a threshold value $L_{RR}$. A typical value of $L_{RR}$ is four times the length of functional electrodes on a MEDA biochip [22]. Therefore, if we set a functional electrode to be an $m \times m$ microelectrode array, $L_{RR}$ is $4m$. This constraint can be expressed in (2.12).

$$|x_i - x_j| + |y_i - y_j| \geq 4m \quad (0 \leq i \leq n) \tag{2.12}$$

The main objective of reservoir placement is to reduce the routing length for subsequent operations, which can reduce the subsequent droplet routing time. In order to construct the objective function, a reservoir graph $G_R$ is constructed from the given sequencing graph. In $G_R$, each vertex corresponds to a reservoir. If there is an operation that relies on two reservoirs, i.e., an operation with droplets dispensed from reservoirs $i$ and $j$, there is an edge $e_{ij}$ in $G_R$ between the two corresponding vertices $v_i$ and $v_j$. In the constructed reservoir graph $G_R$, each

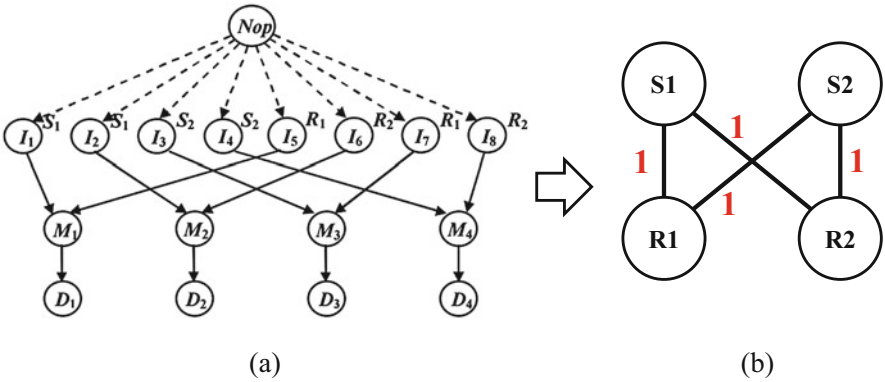



**Fig. 2.7** (**a**) Sequencing graph for a multiplexed bioassay [24] ($S1$, $S2$ are samples, $R1$, $R2$ are reagents, $M1 \sim M4$ are mixing operations, $I1 \sim I8$ are dispensing operations, and $D1 \sim D4$ are detection operations). (**b**) Constructed reservoir graph $G_R$

edge has an associated numerical value, called a weight. The weight $w_{ij}$ of $e_{ij}$ is determined by the number of operations that correspond to reservoirs $i$ and $j$. An example of the constructed reservoir graph $G_R$ from a multiplexed bioassay is shown in Fig. 2.7. The weight of each edge is highlighted in red.

Since the time for droplet routing can be reduced if two corresponding reservoirs are close to each other, the objective function can be expressed as in (2.13).

$$\min : \sum_{1 \le i,j \le n}^{i \ne j} w_{ij} D_{ij} \tag{2.13}$$

where $D_{ij}$ is the Manhattan distance between reservoir $i$ and $j$, i.e., $D_{ij} = |x_i - x_j| + |y_i - y_j|$.

### 2.4.2 Dynamic Priority Assignment

The priority for each unscheduled operation is dynamically determined by the *priority controller*. The priority controller consists of a *priority generator* and a *priority updater*, where the priority generator is used to generate the initial priority for each operation, and the priority updater is used to update the priority whenever an operation has been scheduled or finished.

Two factors are considered in the priority calculation for operation $o_i$: the criticality factor $CF(o_i)$ and the mobility factor $MF(o_i)$. The $CF(o_i)$ is the longest path (in time-steps) from $o_i$ to the sink node $o_{sink}$ in the sequencing graph. The parameter $MF(o_i)$ is calculated based on both As-Soon-As-Possible (ASAP) and As-Last-As-Possible (ALAP) scheduling algorithms, which measure the flexibility

1: **Start**: input sequencing graph $G(V,E)$;
2: Apply ASAP-algorithm and calculate the ASAP-value $AS(o_i)$ for each operation $o_i$;
3: Apply ALAP-algorithm and calculate the ALAP-value $AL(v_i)$ for every operation $o_i$;
4: **for** each unscheduled $o_i$ **do**
5: Calculate $CF(o_i) = dis(o_i, o_{sink})$;
6: Calculate $MF(o_i) = 1/(1 + AL(o_i) - AS(o_i))$;
7: **end for**
8: Normalize $CF(o_i)$ and $MF(o_i)$ for each $o_i$;
9: Calculate normalized factor $NCF(o_i)$ and $NMF(o_i)$ for each $CF(o_i)$ and $MF(o_i)$, respectively;
10: **for** each $o_i$ **do**
11: Initial priority $IP(o_i) = \alpha \times NCF(o_i) + \beta \times NMF(o_i)$;
12: **end for**
13: Assign arriving time $a(o_i) = CF(o_i)$;
14: Return $IP(o_i)$ and $a(o_i)$ for each $o_i$;
15: **End**

**Fig. 2.8** Pseudocode for the priority generator

available to move the operation to different time-slots. We use $AS(o_i)$ and $AL(o_i)$ to represent the scheduled time for operation $o_i$ based on ASAP and ALAP, respectively. After both $AS(o_i)$ and $AL(o_i)$ are derived, $MF(o_i)$ can be calculated using the equation $MF(o_i) = 1/(1 + AL(o_i) - AS(o_i))$. A weighted sum of $CF(o_i)$ and $MF(o_i)$ is used to determine the initial priority for operation $o_i$. In this chapter, we assign $CF(o_i)$ and $MF(o_i)$ weights of 0.8 and 0.2, respectively. This weight combination was found to be the most effective weight combination for minimizing the completion time for synthesis. Meanwhile, the priority generator also assigns the initial arrival time $a(o_i)$ for operation $o_i$. The pseudocode for the priority generator is shown in Fig. 2.8. After some operations have been scheduled or completed, the priority list for the remaining operations may change. Therefore, whenever an operation has been scheduled, the sequencing graph is updated.

Then, the priority updater is invoked to recalculate the priority for each operation. Therefore, both priority list and arrival time for each unscheduled operation can be dynamically updated based on the updated $CF(o_i)$ and $MF(o_i)$.

### 2.4.3 Module Placement

There are three major tasks for the placer: *resource binding*, *space management*, and *module placement*. First, *resource binding* determines the best-fit module $M_i$ from the microfluidic library for operation $o_i$. As reported in [14], lamination mixers/diluters in MEDA biochips can effectively exploit the advantages of low-Reynolds-number fluidics and achieve fast mixing/diluting using as small an area as possible. Therefore, we first attempt to associate the lamination mixer/dilutor with

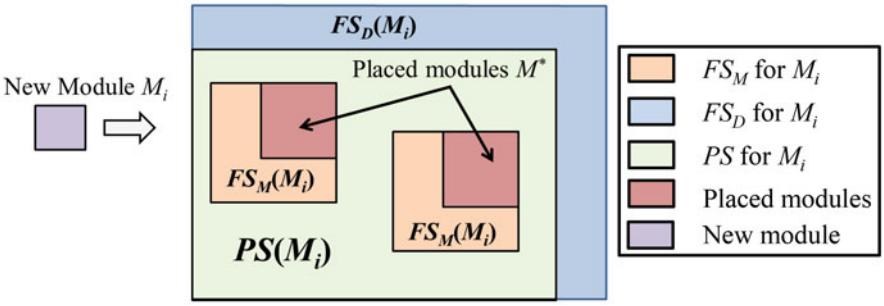



**Fig. 2.9** Illustration of $FS_D$, $FS_M$ and $PS$ relative to a new module $M_i$

a mixing/diluting operation. If the lamination mixer/diluter cannot be placed, we consider conventional mixers/diluters.

The second task, *space management*, identifies all the potential sites where $M_i$ can be placed. A forbidden placement set $FS$ for each new $M_i$, i.e., a set of locations where $M_i$ cannot be placed, is always maintained. There are two types of $FS$s: (1) $FS$s caused by placed modules ($FS_M$) and (2) $FS$s caused by device constraints ($FS_D$). For a new module $M_i$ to be placed on the chip with a set of placed or reserved modules $M^*$, the $FS_M$ is defined as the set of locations where $M_i$ cannot be placed without overlapping any module in $M^*$. We define $FS_D$ as the set of locations where a new module $M_i$ cannot be placed without overlapping with the device boundary. The combined $FS$ is the union of $FS_M$ and $FS_D$. After $FS$ is obtained, the possible placement set $PS$ for $M_i$, i.e., a set of locations where $M_i$ can be placed, is given by $PS = U - FS$, where $U$ is the universal set of all locations on the MEDA biochip. An example illustrating both $FS_M$ and $FS_D$ is shown in Fig. 2.9. As shown in Fig. 2.9, if there is a new module $M_i$ that needs to be placed on the chip, $M_i$ cannot be placed on locations within $FS_M(M_i)$ (the orange region) and $FS_D(M_i)$ (the blue region). A new module $M_i$ can only be placed on locations within $PS(M_i)$ in order to avoid conflict with the chip boundary and existing modules on the chip.

The third task, *module placement*, determines an optimal position for placing the new module in the $PS$. Here we first define the placement cost for a new module $M_i$ ($M_i$ is the associated fluidic module for $o_i$). The placement cost for $M_i$, $Cost(M_i)$, is defined as the maximum droplet routing time from the fluidic modules associated with $o_i$'s parent operations to the location of $M_i$. The placement cost is calculated using (2.14).

$$Cost(M_i) = \max_{o_j \in parent(o_i)} \{RT_{ij}(M_i, M_j)\} \tag{2.14}$$

where $parent(o_i)$ is the set of $o_i$'s parent operations and $M_j$ is the associated fluidic module for $o_j$. The pseudocode for the placer is shown in Fig. 2.10.

A scanning approach can then be used to examine each possible placed region, compute the placement cost for each position, and finally select the optimal location.

```
Start: input new operation o_i;
Bind o_i with the corresponding lamination module M_i;
Calculate FS_D(M_i) and FS_M(M_i);
PS(M_i) = U - FS_D(M_i) ⋃ FS_M(M_i);
if PS(M_i) = ∅ then
    Bind o_i with another fluidic module with the minimum area;
    Recalculate PS(M_i) for the new module;
    if PS(M_i) = ∅ then
        Return FAILED;
    end if
else
    Scan all locations in PS(M_i);
    Output M_i=< x_i, y_i, w_i, h_i > with the minimum cost;
    Return SUCCESS;
end if
End
```

**Fig. 2.10** Pseudocode for the module placer

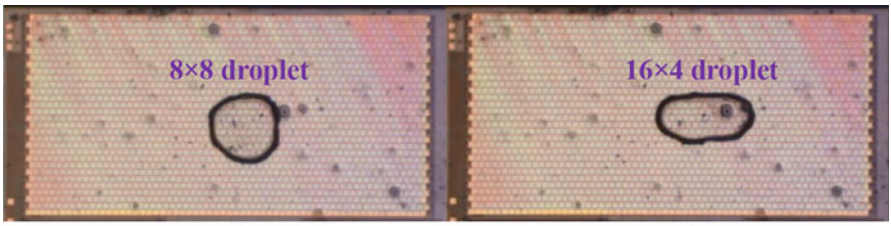



**Fig. 2.11** Experimental demonstration of the morphing of an (**a**) 8 × 8 droplet to a (**b**) 16 × 4 droplet

### *2.4.4 Droplet Routing*

Compared with conventional DMFBs, droplet routing in MEDA biochips is more complicated due to the motion of multiple droplets with different sizes. Furthermore, there are also new fluidic operations that can only be achieved on MEDA biochips, such as shape morphing (see Fig. 2.11) and diagonal droplet motion [14]. These new fluidic operations, which are specific to MEDA biochips, must also be incorporated in the proposed droplet router. The droplet router is used to route droplets to their pre-assigned destinations, which are generated by the placer in Sect. 2.4.3. Due to the differences between MEDA biochips and conventional DMFBs, existing routing algorithms cannot be directly used for MEDA chips. Therefore, a droplet-size-aware router is proposed in this subsection.

The main objective of droplet routing is to determine droplet pathways with minimum lengths, where the length is measured by the number of microelectrodes in the path from the starting location to the destination. Note that fluidic operations

on a MEDA biochip can be completed in much less time compared to a conventional DMFB. Hence, the droplet routing time is comparable to the operation time and it needs to be considered for MEDA biochips.

#### 2.4.4.1 Fluidic Constraints

Similar to conventional DMFBs, there are two types of fluidic constraints for droplet routing in MEDA biochips: (1) static constraint and (2) dynamic constraint [25]. Since droplets with different sizes can be manipulated on a MEDA biochip, the fluidic constraints for MEDA biochips are different from conventional DMFBs. To accomplish routing under these constraints, we first define the following parameters to describe the droplets on the chip (see Fig. 2.12a).

- $R$ is the bottom-left point of the droplet, which is defined as the reference point that indicates the droplet's position;
- $W_d$ is the width of the droplet;
- $H_d$ is the height of the droplet;
- $B$ is defined as the bounding width, which represents the minimum distance between two droplets that will prevent unexpected mixing. In this work, we experimentally demonstrate the bounding width is one microelectrode; i.e., $B = 1$. The experimental demonstration is shown in Fig. 2.12b.

As shown in Fig. 2.12b, the left droplet is a $10 \times 10$ droplet located at (15, 10). The right droplet is a $15 \times 15$ droplet located at (25, 9). Here, we use $(x_i^t, y_i^t, W_i, H_i)$ to represent a $W_i \times H_i$ droplet located at $(x_i^t, y_i^t)$. Accordingly, the left and right droplet are represented as (10, 10, 15, 10) and (15, 15, 25, 9), respectively. Static and dynamic constraints for MEDA biochips are described in following paragraphs.

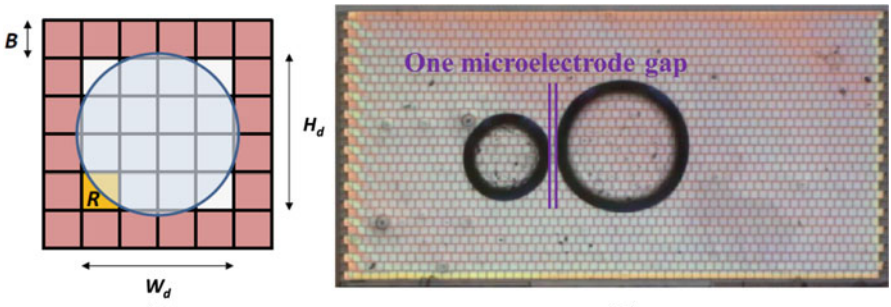



**Fig. 2.12** (**a**) Illustration of droplet parameters, and (**b**) the bounding width between two droplets on a MEDA biochip

*Static Constraint* The static constraint defines the minimum distance between any two droplets $D_i$ and $D_j$ at any time step $t$. As shown in Fig. 2.12b, a minimum distance of one microelectrode must be kept between any two adjacent droplets. Therefore, constraint (2.15) must be satisfied for the static constraint.

$$\begin{aligned} \left|x_i^t - x_j^t\right| &\geq \begin{cases} W_j + 1 & if\ x_i^t > x_j^t \\ W_i + 1 & if\ x_i^t \leq x_j^t \end{cases} \quad or \\ \left|y_i^t - y_j^t\right| &\geq \begin{cases} H_j + 1 & if\ y_i^t > y_j^t \\ H_i + 1 & if\ y_i^t \leq y_j^t \end{cases} \end{aligned} \tag{2.15}$$

Experimental demonstration of the static constraint is presented in Fig. 2.13. To demonstrate (2.15), we forced droplets $D_i$ and $D_j$ to move by activating functional electrodes 2 and 3, and deactivating functional electrodes 1 and 4, simultaneously. Since the new locations of these two droplets, i.e., electrode 2 and 3, were adjacent to each other, they were in contact with each others' surfaces, as shown in Fig. 2.13c. To attain minimum surface energy, the two droplets merged into one droplet; see Fig. 2.13d. Obviously, the violation of constraint (2.15) leads to an unintended mixing of different droplets.

*Dynamic Constraint* The dynamic constraint defines the minimum distance between any two droplets $d_i$ and $d_j$ at two consecutive steps $t$ and $t + 1$. Inequalities (2.16) and (2.17) must be satisfied for the dynamic constraint.

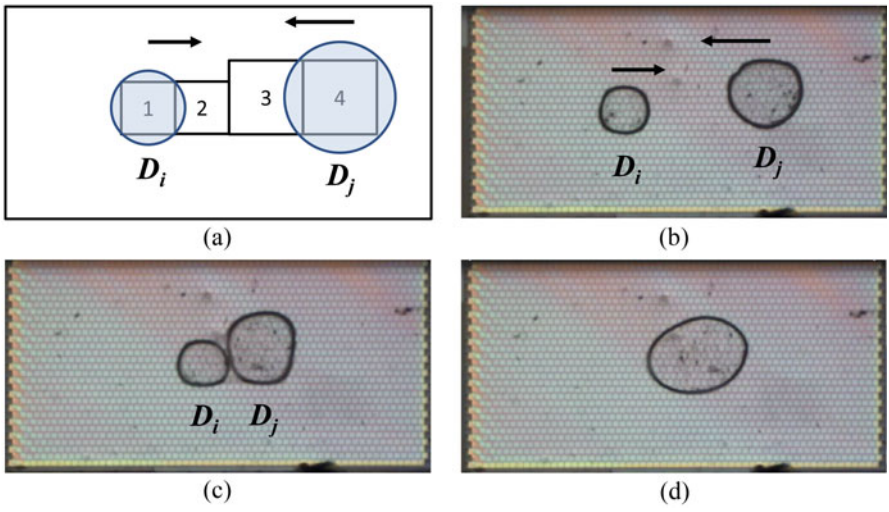



**Fig. 2.13** (**a**) Experimental verification of constraint (2.15); (**b**) Droplets begin moving on functional electrodes 1 and 4; (**c**) Electrodes 2 and 3 are activated, and 1 and 4 deactivated; (**d**) Merged droplet

$$\left|x_i^t - x_j^{t+1}\right| \geq \begin{cases} W_j + 1 & if\ x_i^t > x_j^{t+1} \\ W_i + 1 & if\ x_i^t \leq x_j^{t+1} \end{cases} \quad or$$
$$\left|y_i^t - y_j^{t+1}\right| \geq \begin{cases} H_j + 1 & if\ y_i^t > y_j^{t+1} \\ H_i + 1 & if\ y_i^t \leq y_j^{t+1} \end{cases} \tag{2.16}$$

$$\left|x_i^{t+1} - x_j^t\right| \geq \begin{cases} W_j + 1 & if\ x_i^{t+1} > x_j^t \\ W_i + 1 & if\ x_i^{t+1} \leq x_j^t \end{cases} \quad or$$
$$\left|y_i^{t+1} - y_j^t\right| \geq \begin{cases} H_j + 1 & if\ y_i^{t+1} > y_j^t \\ H_i + 1 & if\ y_i^{t+1} \leq y_j^t \end{cases} \tag{2.17}$$

Experimental demonstration of the dynamic constraint is presented in Fig. 2.14. As shown in Fig. 2.14a–b, two droplets $D_i$ and $D_j$ were placed on functional electrodes 1 and 3 initially. Then, to move $D_i$ and $D_j$ rightward and downward, respectively, electrodes 2 and 4 were actuated simultaneously as shown in Fig. 2.14c. The dynamic constraint was violated due to two activated neighboring cells for $D_j$, i.e., it is directly adjacent to functional electrodes 2 and 4. We observed that droplets $D_i$ and $D_j$ contacted each other, thus leading to the mixing of these two droplets, as shown in Fig. 2.14d.

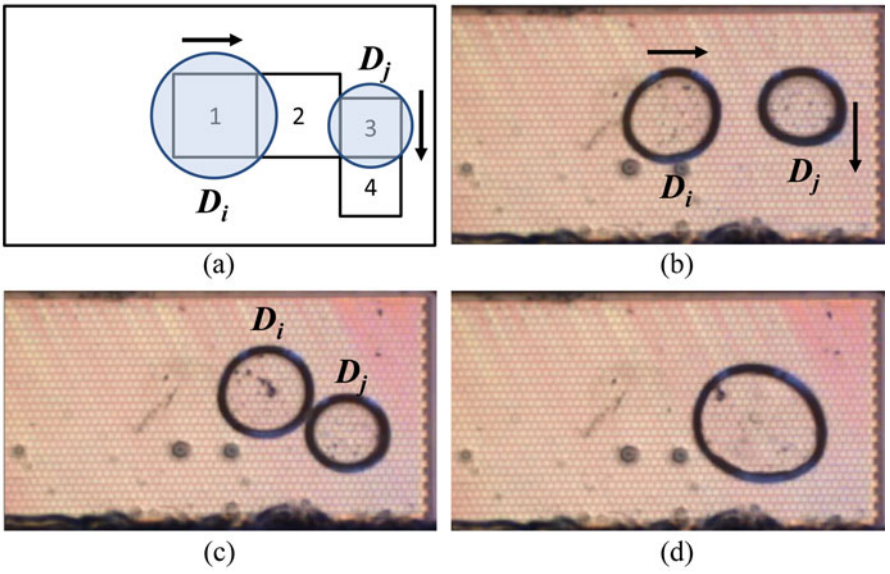



**Fig. 2.14** (**a**) Experimental verification of constraint (2.16) and (2.17); (**b**) Droplets begin moving on functional electrodes 1 and 3; (**c**) Electrodes 2 and 4 are activated, and 1 and 3 deactivated; (**d**) Merged droplet

#### 2.4.4.2 Droplet-Size-Aware Router Design

Since MEDA biochips can exploit the dynamic reconfigurability of the microfluidic array during run-time [15, 26], different 2D placement configurations obtained from the proposed placer can be configured for different time-slots. In this way, we are able to decompose the problem into a series of sub-problems, and each sub-problem can be solved to calculate the routing time between two modules, e.g., the mixer and dilutor. In each sub-problem, the droplets must be routed from their starting locations to their destinations. Fluidic modules and existing droplet routes that are active are regarded as obstacles in the corresponding sub-problem. Based on the above analysis, the routing problem is reduced to finding a route for droplet $D_i$ from its source $S_i$ to its destination $T_i$ avoiding the obstacles on the chip.

In contrast to conventional DMFBs, droplet $D_i$ has to be characterized by its shape and size. Furthermore, shape morphing can also be achieved for droplets on MEDA biochips, which allows a better utilization of the available area. In this chapter, we present a novel router design that is able to consider the diagonal movement, shape morphing, and fluidic constraints presented in Sect. 2.4.4. The router extends a widely used maze routing algorithm, Lee algorithm [27], to determine the droplet pathways. The Lee algorithm can always find connection between two terminals if the connection exists, and it can guarantee the minimum length of the path.

Typically, there are four steps in the Lee algorithm: initialization, wave propagation, backtrace, and clearance [28]. The first step, initialization, aims to identify the source $S$ and the sink $T$ and create routing grids. In the step of wave propagation, the adjacent grids are progressively filled with marks based on the distance of the wavefront from $S$ until the target $T$ is also filled with distance mark. Then, the shortest path is determined from $T$ to $S$ in the backtrace step. Finally, all marks are deleted and the shortest path is preserved.

We adapted Lee algorithm to consider the size and shape of the droplet on a MEDA biochip. Here we refer to the fluidic ports on the boundary of fluidic modules, e.g., mixers and dilutors, as *pins*, and the assignment of the pins is determined by the placer. For simplicity, the bottom-left locations of the fluidic modules are defined as *input pins*. Similarly, we define the bottom-right locations of fluidic modules as *output pins*. After we obtain the module placement results, the source $S_i$ and the destination $T_i$ can be determined. The $m \times n$ MEDA biochip is then divided into $m \times n$ routing grids.

We then start the wave propagation from $S_i$ to $T_i$. Since droplets can move diagonally on a MEDA biochip [14], the distance "wave" can also be propagated diagonally. A comparison between the wave propagation for conventional DMFBs and for MEDA biochips is presented in Fig. 2.15. In Fig. 2.15a, it takes six steps moving from $S_i$ to $T_i$ for conventional DMFBs. In contrast, it takes only three steps for MEDA biochips as shown in Fig. 2.15b. The router has two modes: $diagonal$ mode and $diagonal\_n$ mode. In the $diagonal$ mode, the diagonal motion of a droplet is enabled; In the $diagonal\_n$ mode, diagonal movement is not allowed.

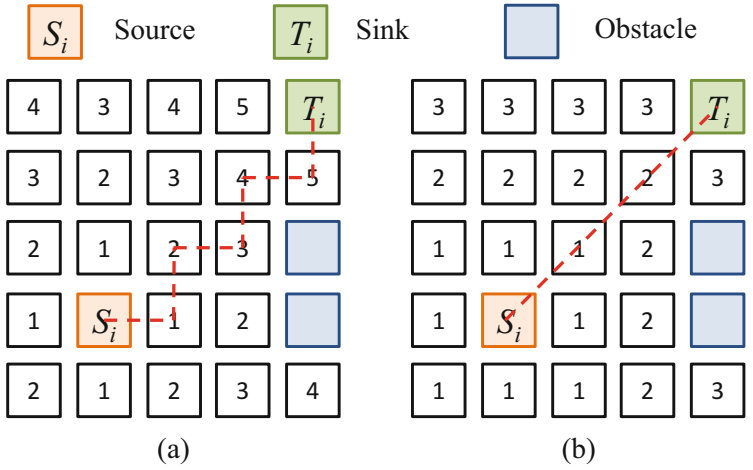



**Fig. 2.15** Examples of wave propagation for (**a**) conventional DMFBs, and MEDA biochips. The droplet pathways from the source to the sink is highlighted using red dotted lines

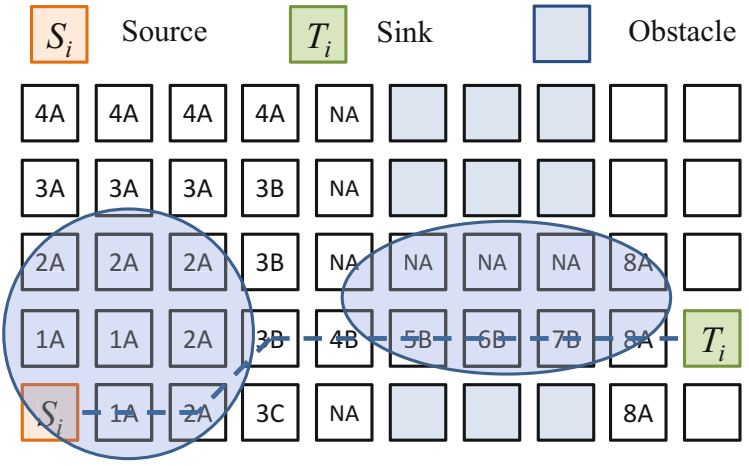



**Fig. 2.16** An example of wave propagation in *diagonal* mode: numeric labels represent the distance information from $S_i$ and alphabetic labels represent the shape information. Label NA means that droplet $D_i$ cannot be transported to that cell in order to avoid the overlap with obstacles. A back-traced droplet route is marked using a blue dotted line

Note that the shape of the droplet can be changed during droplet transportation. Therefore, in addition to the relative distance, droplet shapes should also be recorded in the marks for routing grids. An example is shown in Fig. 2.16. Assume $D_i$ is a $3 \times 3$ droplet, and $D_i$ can be changed to a $4 \times 2$ droplet or a $2 \times 4$ droplet. We use label $A$, $B$, and $C$ to represent the shapes $3 \times 3$, $4 \times 2$, and $2 \times 4$, respectively. In the proposed router, shape morphing can only happen when the droplet meets

other droplets or obstacles (conflict scenario), and the droplet shape is restored when there is no conflict. Therefore, label $A$ is mostly used in Fig. 2.16. If one routing grid cannot be marked with label $A$, and the routing grid can be marked with other labels, we randomly select one feasible label for that routing grid. The wave propagation continues until the destination $T_i$ is marked.

After the wave propagation is completed, the shortest droplet route $R_i$ can be back-traced from $T_i$ to $S_i$. Droplet velocity $v_i$ is first derived from the velocity model presented in Sect. 2.3, and then the routing distance $D_i$ can also be calculated based on the shortest route $R_i$. After we obtain $R_i$ and $v_i$, the droplet routing time ($RT_i$) can be easily calculated using the equation $RT_i = D_i/v_i$. If the start time for $R_i$ is $ST_i$, the active time slot ($ATS_i$) for $R_i$ is defined as $(ST_i, ST_i + RT_i)$.

However, the shortest route is obtained irrespective of the existence of other active droplet routes. A droplet route $R_j$ is defined to be an active droplet route for droplet route $R_i$ if the active time slots of $R_i$ and $R_j$ overlap with each other. Since droplet route $R_i$ and other active droplet routes may share the same microelectrodes or they may be close to each other, we need to ensure that the fluidic constraints specified in Sect. 2.4.4 are not violated. Assume that all active droplet routes for $R_i$ consist of a route set $ACT_{R_i}$. To ensure the correctness of the generated routing path, we need to check whether fluidic constraints are violated between $R_i$ and all routes in $ACT_{R_i}$ in each time slot. If a fluidic constraint is violated between two droplets, we can randomly select one droplet and force the selected droplet to stay in the current location instead of moving. In this way, we can guarantee that no fluidic constraint is violated between any two droplet routes.

### 2.4.5 *Operation Scheduling*

The scheduler determines the start/end times of all operations subject to precedence constraints imposed by the sequencing graph. Resource constraints, e.g., chip area, also need to be satisfied. The pseudocode for the scheduler is shown in Fig. 2.17.

The scheduler maintains two lists: the execution list ($EL$) and the reservation list ($RL$) (line 2); they include currently executing operations and scheduled operations, respectively. In $EL$ ($RL$), the list entries are sorted by end (start) time in ascending order. An operation is inserted into $RL$ when it is accepted by the scheduler. When an operation is scheduled and starts at time $t$, it is removed from $RL$ and is inserted into $EL$. When an operation is completed, it is removed from $EL$.

The basic fluidic operations can be divided into two categories: reconfigurable and non-reconfigurable operations. Reconfigurable operations, e.g., mixing and diluting, can be performed anywhere on a MEDA biochip because they can be triggered by the same pattern of microelectrodes. Other operations, e.g., on-chip dispensing, that cannot be reconfigured are referred as non-reconfigurable operations.

Whenever a non-reconfigurable operation $o_i$ arrives at the scheduler at time $a(o_i)$, it can only be scheduled if the associated resource is available. If the resource

```
1: Start: input o_i from the priority controller;
2: Update RL and EL;
3: if o_i is non-reconfiguable then
4:     Determine whether resource is available;
5:     if yes then
6:         Schedule o_i at a_i(o_i);
7:         Add o_i into RL;
8:     else
9:         Wait for the next time event;
10:    end if
11: else if o_i is reconfiguable then
12:    CRL← RL, CEL← EL;
13:    w_s ← a_i(o_i), w_f ← w_s + e_max(o_i);
14:    for each o_j(a_j(o_j), e_j(o_j)) in CEL do
15:        if a_j + e_j ≤ w_s then
16:            Remove O_j from CEL;
17:        end if
18:    end for
19:    for each o_k(a_k(o_k), e_k(o_k)) in CRL do
20:        if a_k < w_f then
21:            Remove O_k from CRL; Insert O_k into CEL;
22:        end if
23:    end for
24:    Generate M* based on CEL;
25:    if Placer(o_i, M*) returns SUCCESS then
26:        M_i =< x_i, y_i, w_i, h_i >;
27:        if o_i has predecessor operations then
28:            for each M_j associated with predecessor O_j do
29:                Router(M_i, M_j) generates routing time T_ij;
30:            end for
31:            Start time s(o_i) = max{a(o_j) + e(o_j) + T_ij};
32:        else
33:            Start time s(o_i) = a_i(o_i);
34:        end if
35:        o_i =< x_i, y_i, w_i, h_i, s(o_i), e(o_i) >;
36:        Add o_i into RL;
37:    else
38:        Wait for the next time event;
39:    end if
40: end if
41: End
```

**Fig. 2.17** Pseudocode for the scheduler

is not available, operation $o_i$ must wait until the resource becomes available (lines 3–10). On the other hand, if a reconfigurable operation $o_i$ arrives at the scheduler at time $a(o_i)$, the scheduler first checks whether all its parent operations have been completed. If this is the case, the scheduler invokes the placer to bind the operation with a fluidic module $M_i$ and determines an optimal placement for $M_i$ (lines 11–24). If the module can be placed, then the operation can be scheduled (lines 25–36). Otherwise, the operation must wait for the next time event (line 38).

We propose a window-based algorithm for the problem of scheduling reconfigurable operations. Unlike previous work [5], time windows are used instead of time events (e.g., start time); the use of time windows provides more information about each operation. A time window $[w_s, w_f)$ for operation $o_i$ starts and ends at $w_s$ and $w_f$, respectively. The span of the time window $e$ is the execution time, i.e., $e = w_f - w_s$. According to this definition, only tasks whose time windows overlap with $[w_s, w_f)$ will affect the scheduling of $o_i$ (lines 10-14). The analysis based on time windows eliminates the need for a conflict check. For a new operation $o_i$ to be scheduled, the start time $w_s$ is $a(o_i)$ and it is generated by the priority controller. However, the execution time cannot be determined until $o_i$ is bound to a fluidic module $M_i$ (Operation $o_i$ on different modules has different execution times.) To address this problem, we use the worst-case scenario to determine the end time. In other words, $w_f = w_s + e_{max}$, where $e_{max}$ is the maximum execution time for $o_i$ on all fluidic modules in the digital microfluidic library.

The worst-case complexity for the scheduler integrated with the placer is $O(n^2 + nS)$; $n$ and $S$ represent the number of operations and the size of the biochip, respectively. For each operation, the steps corresponding to updating $RL/EL$ (line 2) and generating $M^*$ take $O(n)$ time each. In the worst-case, the placer takes $O(S)$ time to find an optimal location. Therefore, the overall complexity is $O(n(n + n + S)) = O(n^2 + nS)$.

## 2.5 Simulation and Experimental Results

We first present simulation results for three real-life benchmarks and then present an experimental demonstration.

### *2.5.1 Simulation Results*

Three real-life benchmarks, namely PCR [29], sample preparation [30], and protein dilution [2], are used to evaluate the proposed method. Modified list-scheduling [31] and a PRSA-based algorithm [32] for conventional DMFBs are used as baselines for comparison. 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 2.2 (we use the same module library as in [4] for conventional DMFBs). In the simulation, without any loss of generality, we first set the size of one electrode in conventional DMFBs to be equal to a $4 \times 4$ microelectrode array in MEDA DMFBs; therefore, a $2 \times 2$ array for MEDA in Table 2.2 actually represents an $8 \times 8$ microelectrode array. $HS_D$, $HS_N$, $MLS$ and $PRSA$ are used to represent the proposed algorithm with $diagonal$ mode and $diagonal_n$ mode, modified list-scheduling, and the PRSA-based algorithm, respectively. The results generated

**Table 2.2** Experimentally characterized module library for synthesis

| Architecture | Operation | Resource | Time (s) |
|---|---|---|---|
| MEDA | Dispensing | Reservoir | 3 |
| | Mixing | 2 × 2-array mixer | 2 |
| | Diluting | 2 × 2-array diluter | 3 |
| | Sensing | Sensing circuit | 0.01 |

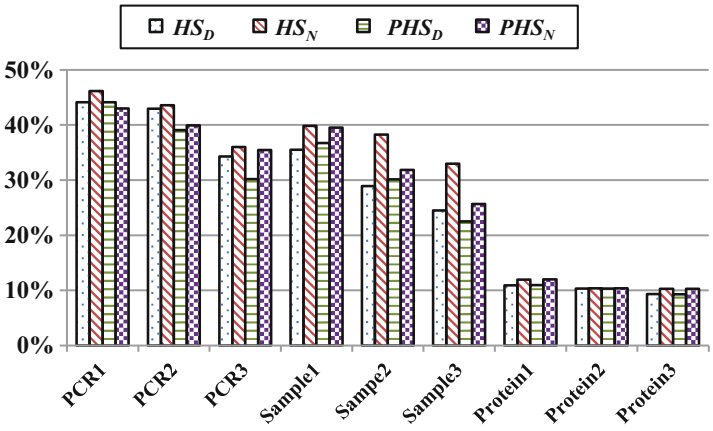



**Fig. 2.18** Comparison between the routing-time percentage from the proposed approach and the approach in [33]

by $MLS$ and $PRSA$ are only for conventional DMFBs, thus no advanced fluidic operations (e.g., lamination mixing and diagonal transporting) are incorporated in those two algorithms. $PCR1$, $PCR2$ and $PCR3$ refer to the PCR benchmark with chip sizes of $10 \times 10$, $8 \times 8$, and $6 \times 6$, respectively. Similarly, $Sample1$, $Sample2$ and $Sample3$ refer to the sample-preparation benchmark with chip sizes of $10 \times 10$, $8 \times 8$, and $6 \times 6$, respectively. Finally, $Protein1$, $Protein2$ and $Protein3$ refer to the protein-dilution benchmark for the three chip sizes.

Figure 2.18 presents the routing-time percentage, defined as the ratio of the routing time to the completion time, for $HS_D$ and $HS_N$. $PHS_D$ and $PHS_H$ are routing-time percentages obtained from [33] with $diagonal$ mode and $diagonal_n$ mode, respectively. As can be seen from Fig. 2.18, the routing percentage for all benchmarks is over 10%, which further demonstrates the necessity of considering the routing time in high-level synthesis. Furthermore, the routing-time percentages from the proposed method is always higher than the routing-time percentages from Fig. 2.18. This is mainly because the routing time is underestimated in [33] and is exactly calculated in the proposed approach.

Simulation results on the completion time of each benchmark are shown in Fig. 2.19. The completion time on a MEDA biochip using the proposed synthesis algorithm is consistently less than the completion time on a conventional DMFB. For the largest benchmark (protein dilution), the completion times obtained using $HS_D$ and $HS_N$ are almost half of the completion times that results from $MLS$ and

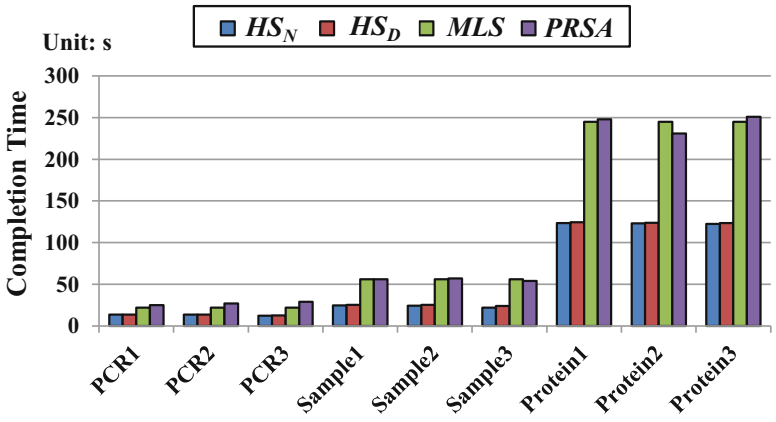



**Fig. 2.19** Comparison between the completion times for different algorithms

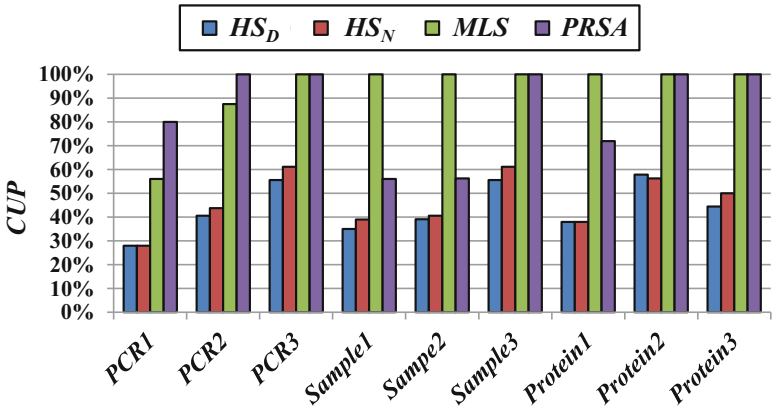



**Fig. 2.20** Comparison between the chip utilization percentages for different algorithms

$PRSA$. Figure 2.19 also shows that there is little difference between the completion times for $HS_D$ and $HS_N$, which illustrates that the ability to transport droplets diagonally does not lead to a significant reduction in the total completion time of the bioassay for these benchmarks. More studies on a more diverse set of benchmarks are needed to better evaluate the benefits of diagonal droplet motion.

Simulation results for the chip utilization percentage ($CUP$) are shown in Fig. 2.20. $CUP$ is defined as the percentage of the used chip area relative to the entire chip area. As shown in Fig. 2.20, the proposed synthesis methods ($HS_D$ and $HS_N$) reduce the used chip area and result in lower $CUP$. Hence they can efficiently utilize dynamic reconfigurability and increase the fault tolerance of the chip. This is because the freed-up area on the chip can either be utilized for more operations in parallel, or for dynamic error recovery in case of faults [32].

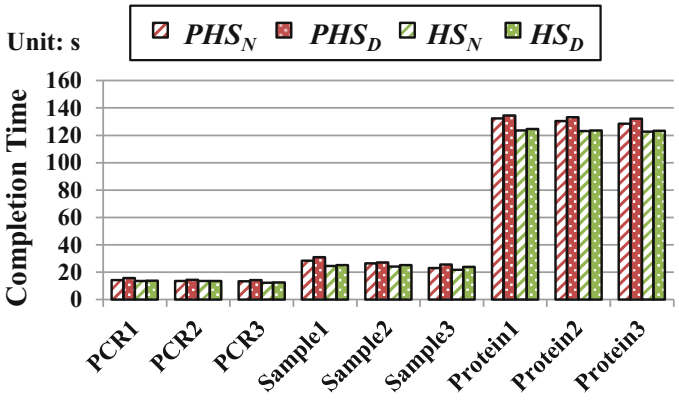



**Fig. 2.21** Comparison between the completion times from the proposed synthesis approach and the approach in [33]

The CPU time for the proposed method (both modes) ranged from 0.19 s to 28.01 s; For MLS (PRSA), the CPU time was in the range of 0.03–1.47 s (8.18–341.69 s).

We finally compared the completion time of benchmarks between the proposed synthesis approach and the approach in [33] in this subsection. Comparison results are presented in Fig. 2.21. Note that both approaches have $diagonal$ mode and $diagonal_n$ mode, and $PHS_D$ and $PHS_N$ are used to present the synthesis approach in [33] with $diagonal$ mode and $diagonal_n$ mode. The major difference between these two approaches is that the routing time is estimated in [33] and is exactly calculated in the proposed approach. As shown in Fig. 2.21, the completion time from [33] is always smaller than the completion time from the proposed approach, which demonstrates that the work of [33] is overly optimistic and the proposed approach provides results that are closer to the real scenario.

### 2.5.2 Experimental Results

Next we present an experimental demonstration of automated synthesis on a MEDA biochip. The sequencing graph for the experiment is shown in Fig. 2.22. Our objective is to compare the simulated completion time with the experimental completion time. The chip micro-photo is shown in Fig. 2.23. The chip has an area of 7.4 $mm^2$ and it was fabricated using a 0.35 μm standard CMOS process. The chip is operated under 3.3 V at 1 MHz frequency for droplet sensing, while the top plate is operated under 25 V at 1 KHz frequency for microfluidic operations. An illustration of the test environment and experimental setup for the demonstration is shown in Fig. 2.24.

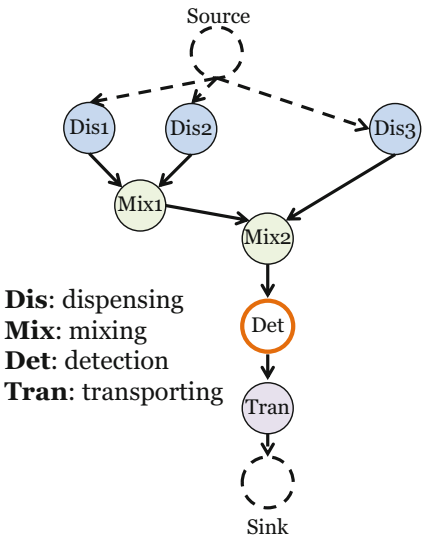



**Fig. 2.22** The sequencing graph used for the experiment

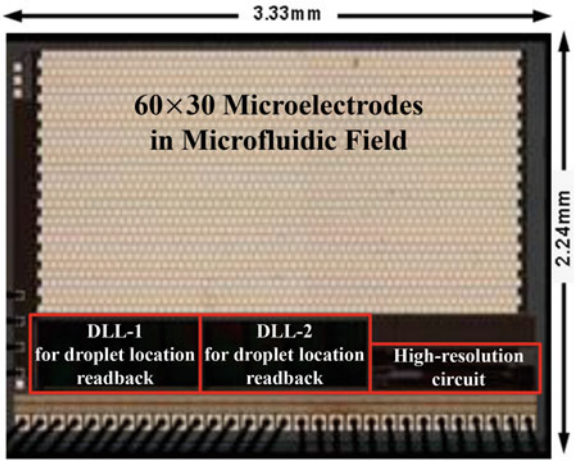



**Fig. 2.23** Micro-photo for the MEDA biochip used in the experiment

We present the experimental results in both $diagonal_n$ and $diagonal$ mode. The proposed synthesis tool first generates the scheduling, module placement and droplet routing results in $diagonal_n$ mode. According to the simulation results, the example bioassay can be completed in 15.48 s. Next, the experiment was carried out using the fabricated chip and the control software. A CCD camera was used to capture the video and images were extracted from the recorded video. The experimental demonstration finished in 17.19 s. A comparison between simulation and experimental results is shown in Fig. 2.25a. Similarly, the comparison between simulation and experimental results in $diagonal$ mode is shown in Fig. 2.25b. The close match between the simulation and experimental results shows that the

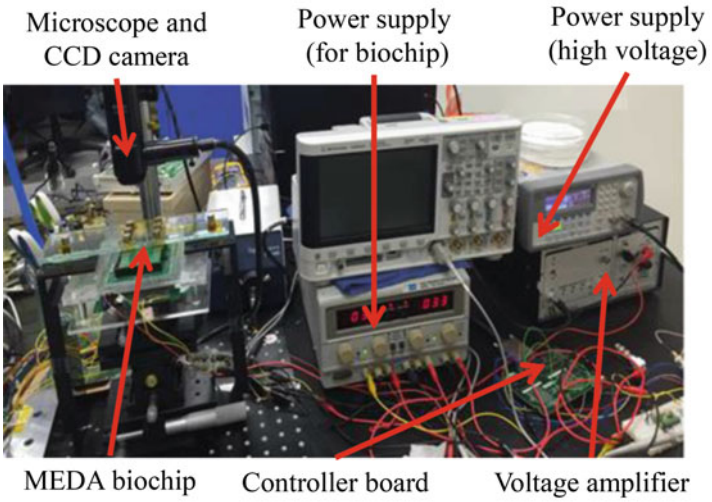



**Fig. 2.24** Illustration of the experimental setup

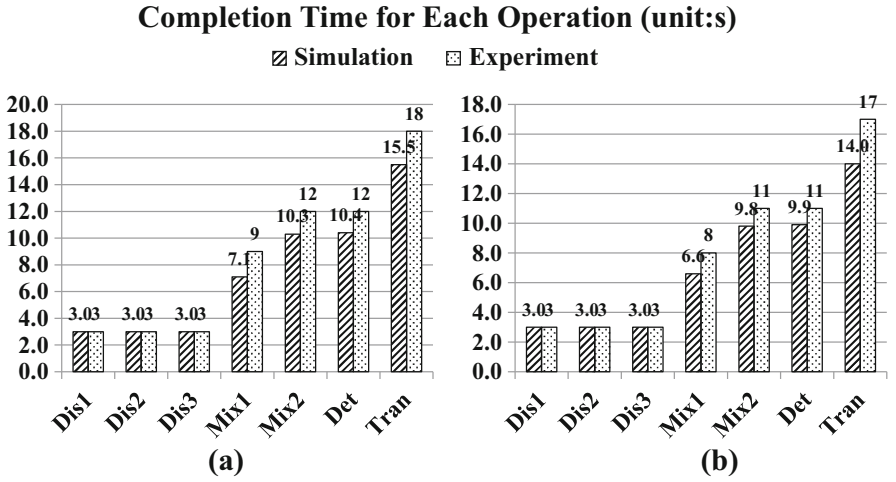



**Fig. 2.25** Schedule obtained using the proposed synthesis tool in (**a**) $diagonal_n$ mode and (**b**) $diagonal$ mode

simulated results can be used as an accurate predictor of results for an actual experimental setting. Videos of experiments in $diagonal_n$ and $diagonal$ mode are available in [34] and [35], respectively; corresponding snapshots of the experiments are also presented in Figs. 2.26 and 2.27.

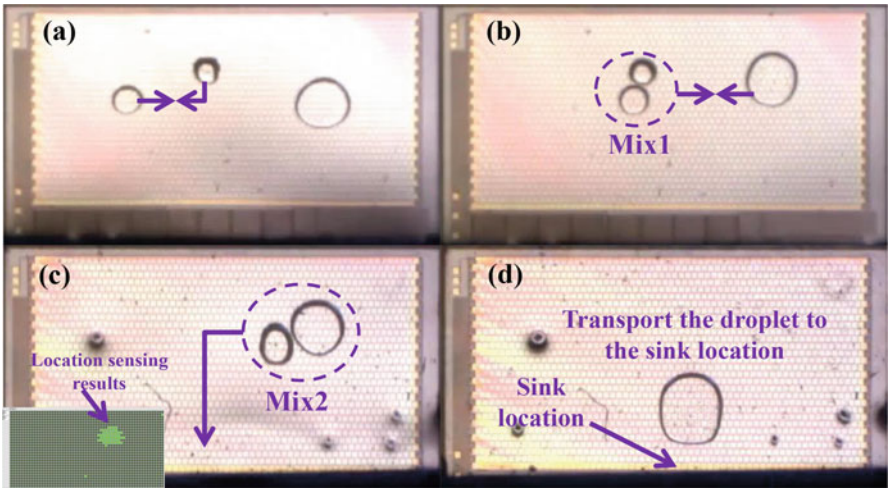



**Fig. 2.26** Frames from the video. (**a**) Initial location of three droplets of different sizes. (**b**) Two droplets are going to be moved horizontally/vertically to the mixing region (Mix1). (**c**) One droplet and the previously mixed droplet are going to be mixed (Mix2); Location sensing is performed after Mix2. (**d**) Transportation to the sink

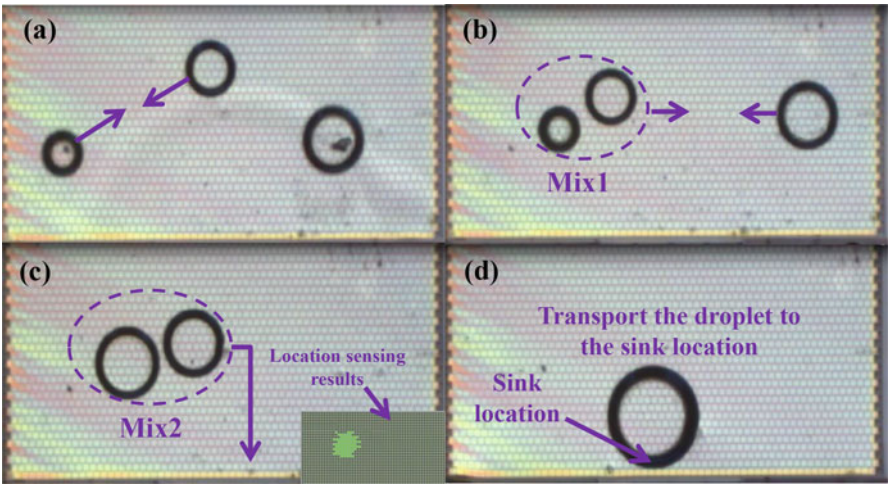



**Fig. 2.27** Frames from the video. (**a**) Initial location of three droplets of different sizes. (**b**) Two droplets are going to be moved diagonally to the mixing region (Mix1). (**c**) One droplet and the previously mixed droplet are going to be mixed (Mix2); Location sensing is performed after Mix2. (**d**) Transportation to the sink

## 2.6 Conclusion

This chapter presents the first droplet size-aware biochemistry synthesis technique for MEDA biochips. The proposed technique performs scheduling, placement, and routing in a unified manner. The synthesis approach is able to route sample droplets of different sizes and it incorporates new MEDA-specific fluidic operations such as diagonal droplet motion and droplet shape morphing. We have presented an analytical model for droplet velocity and validated it experimentally using a fabricated MEDA biochip. Simulation and experimental results highlight the effectiveness of the proposed synthesis approach.

## References

1. Chakrabarty, K. (2010). Design automation and test solutions for digital microfluidic biochips. *IEEE Transactions on Circuits and Systems I: Regular Papers, 57*(1), 4–17.
2. Su, F., & Chakrabarty, K. (2005). Unified high-level synthesis and module placement for defect-tolerant microfluidic biochips. In *Proceedings of ACM/IEEE Design Automation Conference*, pp. 825–830.
3. Tao, X., & Chakrabarty, K. (2006). Module placement for fault-tolerant microfluidics-based biochips. *ACM Transactions on Design Automation of Electronic Systems, 11*(3), 682–710.
4. Xu, T., & Chakrabarty, K. (2007). Integrated droplet routing in the synthesis of microfluidic biochips. In *ACM/IEEE Design Automation Conference*, 948–953.
5. Maftei, E., Pop, P., & Madsen, J. (2009). Tabu search-based synthesis of dynamically reconfigurable digital microfluidic biochips. In *Proceedings of International Conference on Compilers, Architecture, and Synthesis for Embedded Systems*, pp. 195–204.
6. Grissom, D., & Brisk, P. (2012). Fast online synthesis of generally programmable digital microfluidic biochips. In *Proceedings of IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS)*, pp. 413–422.
7. Chen, Y.-H., Hsu, C.-L., Tsai, L.-C., Huang, T.-W., & Ho, T.-Y. (2013). A reliability-oriented placement algorithm for reconfigurable digital microfluidic biochips using 3-d deferred decision making technique. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 32*(8), 1151–1162.
8. Keszocze, O., Wille, R., Ho, T.-Y., & Drechsler, R. (2014). Exact one-pass synthesis of digital microfluidic biochips. In *Proceedings of ACM/IEEE Design Automation Conference*, pp. 1–6.
9. O'neal, K., Grissom, D., & Brisk, P. (2017). Resource-constrained scheduling for digital microfluidic biochips. *ACM Journal on Emerging Technologies in Computing Systems, 14*(1), 7.
10. Su, F., & Chakrabarty, K. (2008). High-level synthesis of digital microfluidic biochips. *ACM Journal on Emerging Technologies in Computing Systems, 3*(4), 1.
11. Ibrahim, M., Li, Z., & Chakrabarty, K. (2015). Advances in design automation techniques for digital-microfluidic biochips. In *Formal Modeling and Verification of Cyber-Physical Systems*, pp. 190–223.
12. Grissom, D., & Brisk, P. (2012). Path scheduling on digital microfluidic biochips. In *Proceedings of ACM/IEEE Design Automation Conference*, pp. 26–35.
13. Ricketts, A. J., Irick, K., Vijaykrishnan, N., & Irwin, M. J. (2006). Priority scheduling in digital microfluidics-based biochips. In *Proceedings of Design, Automation and Test Conference in Europe*, pp. 329–334.
14. Wang, G., Teng, D., & Fan, S.-K. (2011). Digital microfluidic operations on micro-electrode dot array architecture. *IET Nanobiotechnology, 5*(4), 152–160.

15. Wang, G., Teng, D., Lai, Y.-T., Lu, Y.-W., Ho, Y., & Lee, C.-Y. (2013). Field-programmable lab-on-a-chip based on microelectrode dot array architecture. *IET Nanobiotechnology, 8*(3), 163–171.
16. Lai, K. Y.-T., Yang, Y.-T., & Lee, C.-Y. (2015). An intelligent digital microfluidic processor for biomedical detection. *Journal of Signal Processing Systems, 78*(1), 85–93.
17. Bahadur, V., & Garimella, S. (2006). An energy-based model for electrowetting-induced droplet actuation. *Journal of Micromechanics and Microengineering, 16*(8), 1494.
18. Oprins, H., Vandevelde, B., & Baelmans, M. (2012). Modeling and control of electrowetting induced droplet motion. *Micromachines, 3*, 150–167.
19. Bhattacharjee, B., & Najjaran, H. (2009). Size dependent droplet actuation in digital microfluidic systems. In *SPIE Defense, Security, and Sensing*, pp. 73 180H–73 180H.
20. Ren, H., Fair, R. B., Pollack, M. G., & Shaughnessy, E. J. (2002). Dynamics of electro-wetting droplet transport. *Sensors and Actuators B: Chemical, 87*(1), 201–206.
21. Lai, K. Y.-T., Shiu, M.-F., Lu, Y.-W., Ho, Y.-C., Kao, Y.-C., Yang, Y.-T., et al. (2015). A field-programmable lab-on-a-chip with built-in self-test circuit and low-power sensor-fusion solution in 0.35 $\mu$m standard CMOS process. In *Proceedings of IEEE Asian Solid-State Circuits Conference*, pp. 1–4.
22. Luo, Y., Bhattacharya, B. B., Ho, T.-Y., & Chakrabarty, K. (2015). Design and optimization of a cyberphysical digital-microfluidic biochip for the polymerase chain reaction. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 34*, 29–42.
23. Li, Z., Ho, T.-Y., & Chakrabarty, K. (2016). Optimization of 3D digital microfluidic biochips for the multiplexed polymerase chain reaction. *ACM Transactions on Design Automation of Electronic Systems, 21*, 25:1–27.
24. Xu, T., & Chakrabarty, K. (2008). Broadcast electrode-addressing for pin-constrained multi-functional digital microfluidic biochips. In *Proceedings of ACM/IEEE Design Automation Conference*, pp. 173–178.
25. Su, F., Hwang, W., & Chakrabarty, K. (2006). Droplet routing in the synthesis of digital microfluidic biochips. In *Proceedings of IEEE/ACM Design, Automation and Test Conference in Europe, 1*, pp. 1–6.
26. Chen, Z., Teng, D. H.-Y., Wang, G. C.-J., & Fan, S.-K. (2011). Droplet routing in high-level synthesis of configurable digital microfluidic biochips based on microelectrode dot array architecture. *Biochip Journal, 5*(4), 343–352.
27. Lee, C. (1961). An algorithm for path connections and its applications. *IRE Transactions on Electronic Computers, EC-10*, 346–365.
28. Sherwani, N. A. (2012). *Algorithms for VLSI Physical Design Automation*. New York: Springer Science and Business Media.
29. Fei, S. (2004). Module placement for fault-tolerant microfluidics-based biochips. *ACM Transactions on Design Automation of Electronic Systems, 11*(3), 682–710.
30. Roy, S., Bhattacharya, B. B., Chakrabarti, P. P., & Chakrabarty, K. (2011). Layout-aware solution preparation for biochemical analysis on a digital microfluidic biochip. In *Proceedings of IEEE International Conference on VLSI Design*, pp. 171–176.
31. Alistar, M., & Pop, P. (2015). Synthesis of biochemical applications on digital microfluidic biochips with operation execution time variability. *Integration, the VLSI Journal, 51*, 158–168.
32. Luo, Y., Chakrabarty, K., & Ho, T.-Y. (2013). Error recovery in cyberphysical digital microfluidic biochips. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 32*(1), 59–72.
33. Li, Z., Lai, K. Y.-T., Yu, P.-H., Ho, T.-Y., Chakrabarty, K., & Lee, C.-Y. (2016). High-level synthesis for micro-electrode-dot-array digital microfluidic biochips. In *Proceedings of ACM/IEEE Design Automation Conference* (pp. 146:1–146:6). New York: ACM.
34. [Online video], http://people.duke.edu/%7ezl67/videos/1.mp4.
35. [Online video], http://people.duke.edu/%7ezl67/videos/2.mp4.

# Chapter 3
# Efficient and Adaptive Error Recovery


## 3.1 Chapter Overview

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 [1]. In order to improve the confidence of chip experimental results, efficient error recovery is essential for MEDA biochips.

In this chapter, we present an efficient and adaptive error recovery strategy for MEDA biochips. In the proposed strategy, the outcomes of operations are classified into three categories: no error, minor error, and major error. Each outcome is treated in a different way. Instead of utilizing the same error-recovery strategy for all types of errors (e.g., roll-back [2, 3]), the proposed strategy utilizes different probabilistic timed automata (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. An on-line synthesis approach and a control flow are utilized to connect the local recovery procedures to global error recovery for the complete bioassay. An integer linear programming (ILP)-based method is finally proposed for determining the optimal local recovery time for each operation in the bioassay.

The remainder of the chapter is organized as follows. Section 3.2 discusses prior works and presents the motivation for error recovery in MEDA biochips. Section 3.3 presents the PTA-based error-recovery strategy in detail. Section 3.4 presents the method to select the optimal time limit for local error recovery. Section 3.5 describes the proposed method for global error recovery. Section 3.6 presents experimental results. Finally, Sect. 3.7 concludes the chapter.



Z. Li et al., *Micro-Electrode-Dot-Array Digital Microfluidic Biochips*,
https://doi.org/10.1007/978-3-030-02964-7_3

## 3.2 Prior Works and Motivation

Since thousands of electrodes and associated control/sensing circuits can be integrated into a single MEDA biochip, there is more likely to have defects on the chip. In addition to defects and imperfections for fabricated MEDA biochips, faults may also arise during bioassay execution. For example, excessive actuation voltage may lead to electrode breakdown and charge trapping [4], and DNA fouling may lead to the malfunction of multiple electrodes in the biochip [5]. 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 [2, 3, 6–9]. 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 [10]. Moreover, previous strategies use a unified recovery procedure (e.g., roll-back [2, 3]) for all types of errors; this approach can make error recovery inefficient and result in longer recovery time for specific types of errors. An example of roll-back is shown in Fig. 3.1. The initial sequencing graph corresponding to a bioassay is shown in Fig. 3.1a. When an error occurs at operation 9, the system will re-execute the corresponding dispensing

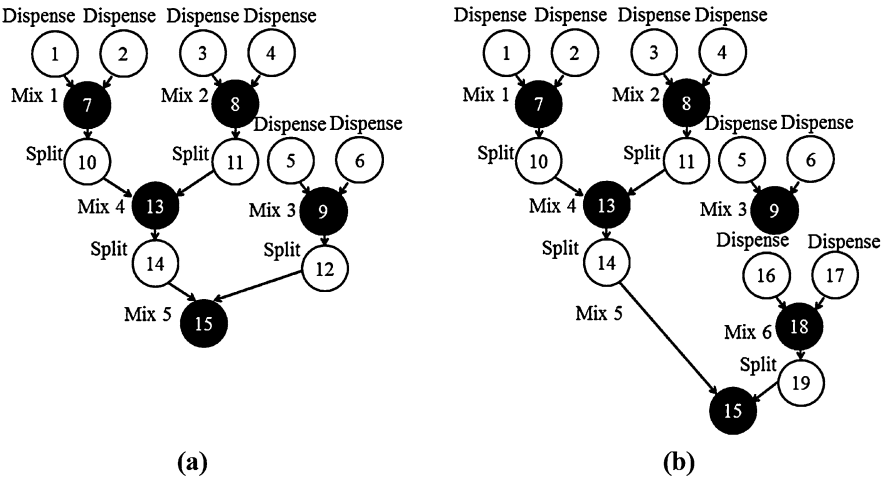



**Fig. 3.1** (**a**) Initial sequencing graph; (**b**) Updated sequencing graph for error recovery

operations and mixing operations (operations 16, 17, and 18). The new sequencing graph for error recovery is shown in Fig. 3.1b. However, the re-execution of operation 16, 17, and 18 may not always be the most-efficient solution since we can recover from the error on operation 9 by redoing the mixing operation. 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 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.

## 3.3 Error Recovery for Local Errors

In this section, we first describe the errors that are being targeted in this chapter. 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 [11]. 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.

### *3.3.1 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 [12]. Fault models can be used to represent the effect of physical defects at some level of abstraction [13]. As described in [7, 14–16], 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 [14]. In this chapter, we assume that chips have been carefully tested using both functional and structural test methods [14, 16] 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 bio-

logical 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.

### *3.3.2 Outcome Classification of Fluidic Operations*

Outcomes of error detection can either be "success" or "failure" in a conventional DMFB [3, 6]. 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.

#### 3.3.2.1 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. 3.2a: 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. 3.2b. We next quantify the error factor ($F_{mix}$) for mixing operation:

$$F_{mix} = \frac{N_P}{N_T} \tag{3.1}$$

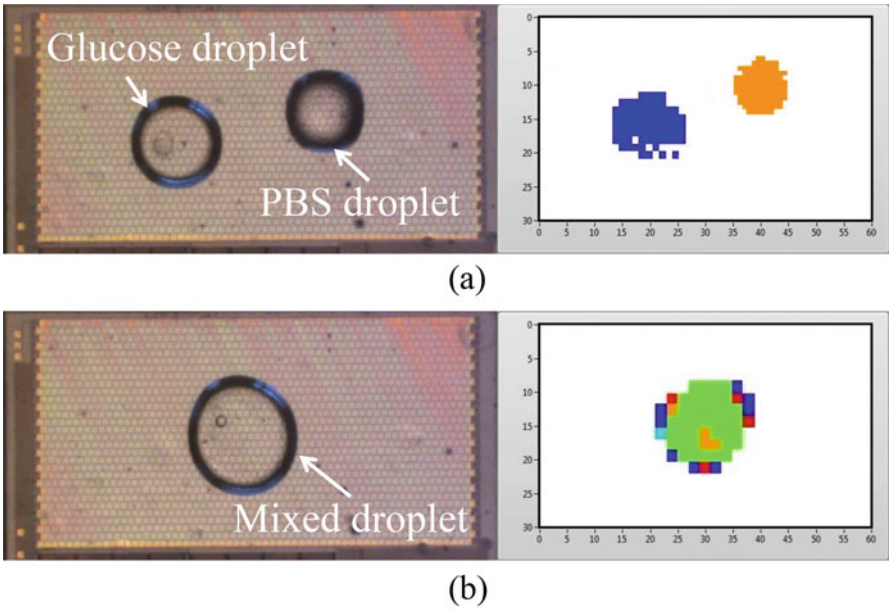



**Fig. 3.2** Illustration of (**a**) two droplets before mixing and the corresponding droplet-property sensing map, and (**b**) the mixed droplet and the corresponding droplet-property sensing map

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 (1) *major error* if $F_{mix} \leq T_{m1}$, (2) *minor error* if $T_{m1} < F_{mix} \leq T_{m2}$, or (3) *no error* if $F_{mix} > T_{m2}$.

#### 3.3.2.2 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 (3.2), 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.

$$F_{split} = 1 - \frac{|N_1 - N_2|}{\max\{N_1, N_2\}} \tag{3.2}$$

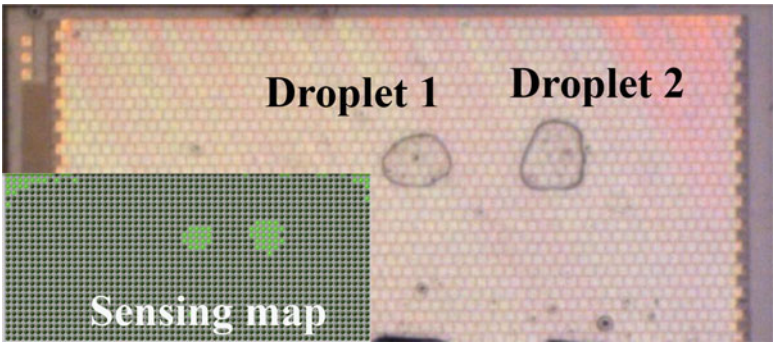



**Fig. 3.3** Illustration of two split droplets on a MEDA-based chip and the corresponding droplet-sensing map

An example is shown in Fig. 3.3. 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. 3.3 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 (1) *major error* if $F_{split} \leq T_{s1}$, (2) *minor error* if $T_{s1} < F_{split} \leq T_{s2}$, or (3) *no error* if $F_{split} > T_{s2}$.

#### 3.3.2.3 Outcome Classification for Dilution

A dilution operation can be regarded as a mixing operation followed by a splitting operation [8, 20]. 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 3.1.

### *3.3.3 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. In this chapter, without any loss of generality, we set $T_{m1}$, $T_{m2}$, $T_{s1}$, and $T_{s2}$ to be 0.70 and 0.90, 0.50, and 0.80, respectively.

**Table 3.1** 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 |

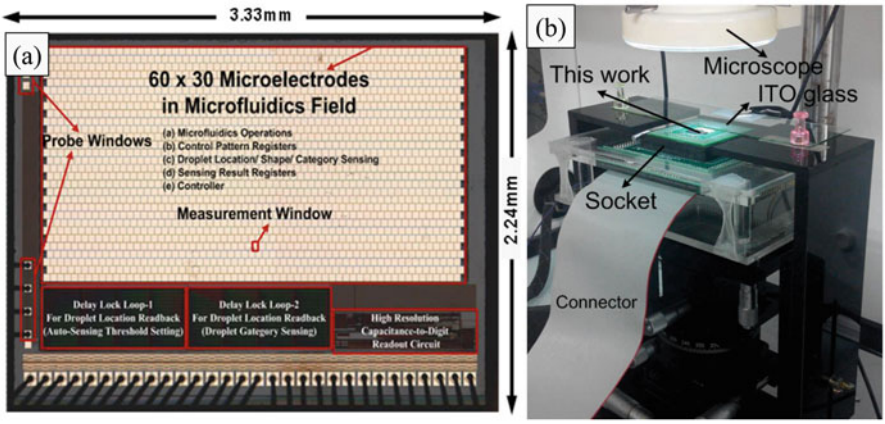



**Fig. 3.4** (**a**) Chip micro-photo and (**b**) experimental setup

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. 3.4. The splitting operation was performed using a $7 \times 7$ deionized (DI) water droplet. The mixing operation was performed using a $10 \times 10$ PBS droplet and a $10 \times 10$ 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 Sect. 3.3.2, 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. Experimentally characterized outcome probabilities are shown in Table 3.2. We observed that the outcome probability of one operation does not depend on the outcome of the previous operation. Therefore, the

**Table 3.2** Experimentally characterized outcome probability

| | Probability | | |
|---|---|---|---|
| Outcome | 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 |

outcome probabilities in Table 3.2 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 3.2, 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.

### *3.3.4 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.

#### 3.3.4.1 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.

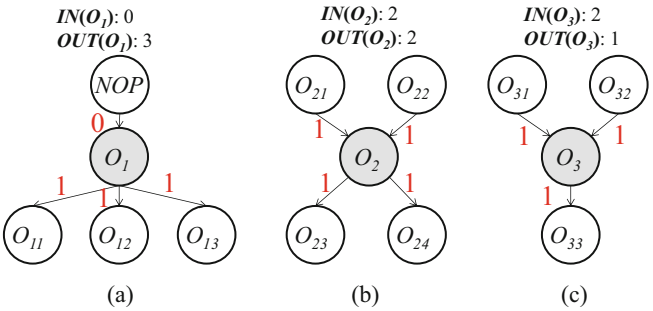



**Fig. 3.5** Examples of operations in different categories in (**a**) Category I, (**b**) Category II, and (**c**) Category III. Operation $NOP$ is a pseudo-source operation. Red numbers represent the number of unit droplets from operations to their successor operations

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. 3.5a. 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. 3.5b. 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. 3.5c. 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.

#### 3.3.4.2 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. 3.6a. 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.

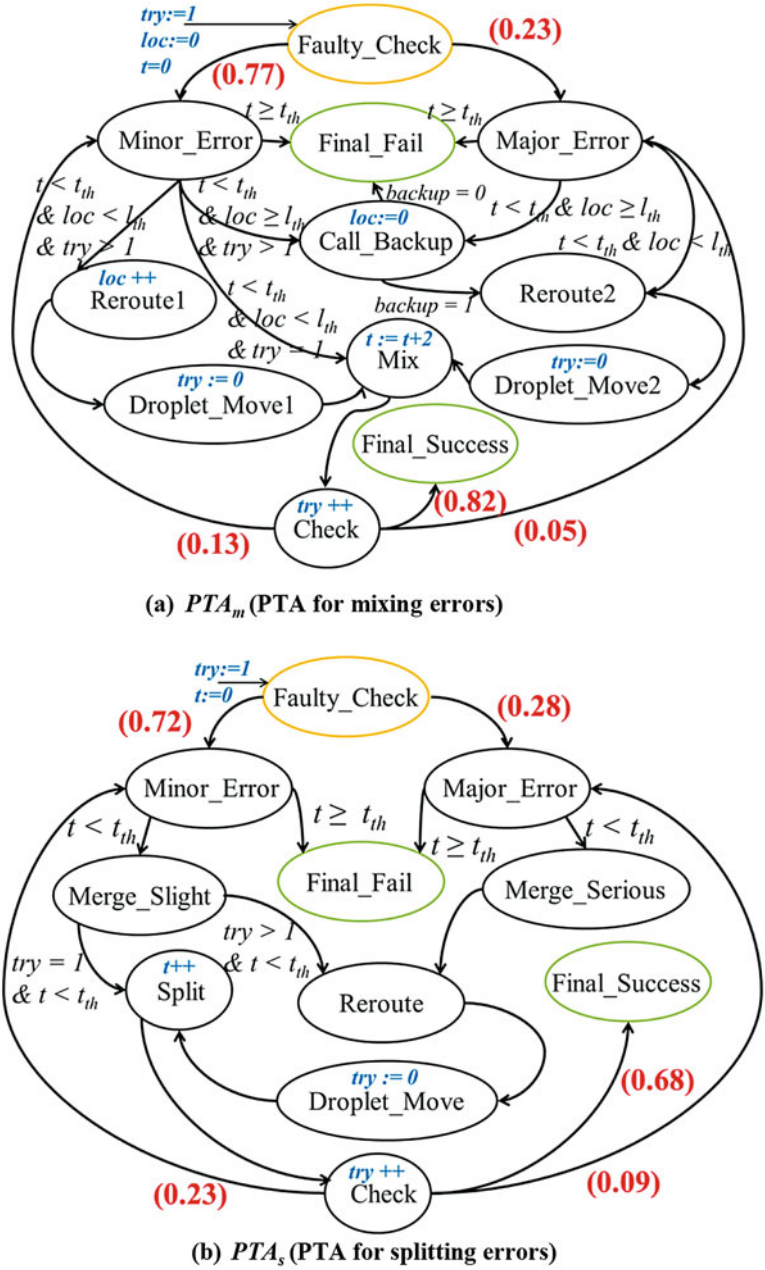



**Fig. 3.6** Illustration of PTAs for recovery from (**a**) mixing errors, and (**b**) splitting errors. Orange (green) outline indicates that the corresponding state is a start (end) state in the state transition diagram

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 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 Sect. 3.3.3 and highlighted in Fig. 3.6a. 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 Sect. 3.3.4 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. 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. 3.6a. 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. 3.6b. 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. 3.7. 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. 3.7). 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 $detect$

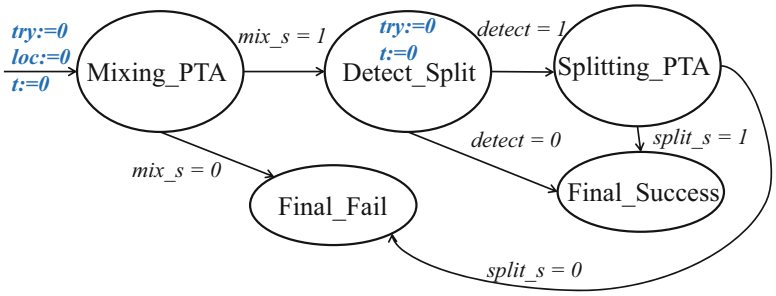



**Fig. 3.7** Illustration of $PTA_d$ (PTA for error recovery of dilution errors)

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 made from the splitting error.

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 [21]. 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 [22]. 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: (1) Final_Success_2, (2) Final_Success_4, (3) Final_Success_6, (4) 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 [23], 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. An example is shown in Fig. 3.8. In this example, $l_{th}$ is 2 and $t_{th}$ is 4 s. Blue numbers represent the transition probability from one state to the next state. For the sake of simplicity, states do not associated with time cost and probability transition are eliminated in Fig. 3.8. After DTMC simulation (probability calculation), the

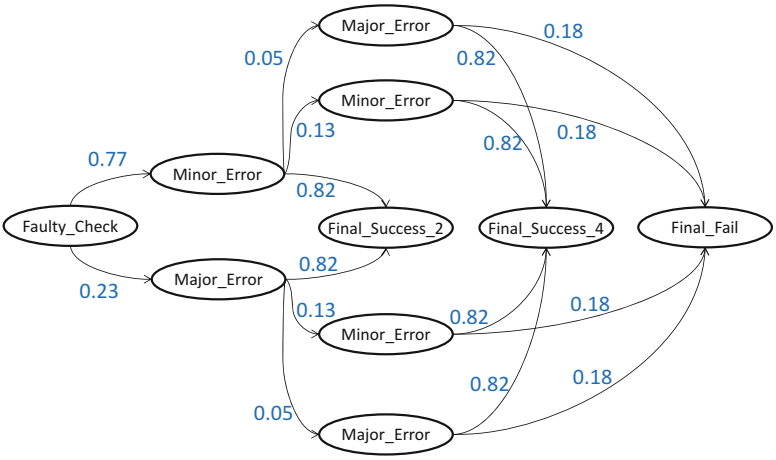



**Fig. 3.8** An example of DTMC for $PTA_m$

probabilities that the recovery finally locates in Final_Success_2, Final_Success_4, and Final_Fail are 0.82, 0.15, and 0.03, respectively. The probability of successful recovery in Fig. 3.8 is $0.97 = 0.82 + 0.15$, i.e., the summation of the probabilities that the final state locates in Final_Success_2 and Final_Success_4. Detailed analysis of PTAs $PTA_s$, $PTA_m$, and $PTA_d$ are presented in Sect. 3.6.

## 3.4 Time-Limit Selection for Local Recovery

PTAs for local recovery proposed in Sect. 3.3.4 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 [1] 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.

### *3.4.1 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 Sect. 3.3.3. 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 Sect. 3.6.1. Based on the above definitions, the probability that the operation $O_i$ can be successfully executed can be calculated as follows.

$$P_i = P_s(tp_i) + P_f(tp_i) \cdot P_{ri}(T_i) \tag{3.3}$$

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 $j$. Accordingly, $P_{ri}(T_i)$ can be calculated using (3.4), where $TR_{max}$ is the predefined upper bound on TLLR for all operations.

$$P_{ri}(T_i) = \sum_{j=1}^{TR_{\max}} \left(S_{ij} \cdot d_{ij}\right) \tag{3.4}$$

Similarly, the TLLR $T_i$ for operation $O_i$ can be calculated using (3.5).

$$T_i = \sum_{j=1}^{TR_{\max}} \left(j \cdot d_{ij}\right) \tag{3.5}$$

Using above defined variables, the objective function of the ILP model is shown in (3.6), where $N$ is the number of operations in the given bioassay.

$$\text{Minimize} \quad \frac{1}{N} \sum_{i=1}^{N} T_i \tag{3.6}$$

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 (3.7).

$$\prod_{i=1}^{N} P_i \geq P_{th} \tag{3.7}$$

It should be noted that (3.7) is a non-linear constraint. We linearize this by introducing a new variable $Q_m = \prod_{i=1}^{m} P_i$. The constraint (3.7) can be transformed to (3.8).

**Fig. 3.9** ILP model for the optimization of TLLR selection

**Objective:**

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

**Subject to:**

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

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

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

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

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

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

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

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

$$Q_1 = P_1$$

$$Q_N \geq P_{th}$$

$$\begin{aligned} POS &= Q_N \\ Q_N &= Q_{N-1} \cdot P_N \\ Q_{N-1} &= Q_{N-2} \cdot P_{N-1} \\ &\cdots \\ Q_2 &= Q_1 \cdot P_2 = P_1 \cdot P_2 \end{aligned} \tag{3.8}$$

Variable $Q_i$ in (3.8) can be calculated using (3.9).

$$\begin{aligned} Q_i &= Q_{i-1} \cdot P_i \\ &= Q_{i-1} \cdot \left( P_s(tp_i) + P_f(tp_i) \cdot \sum_{j=1}^{TR_{\max}} (S_{ij} \cdot d_{ij}) \right) \\ &= P_s(tp_i) \cdot Q_{i-1} + P_f(tp_i) \cdot \sum_{j=1}^{TR_{\max}} \left( S_{ij} \cdot d_{ij} \cdot Q_{i-1} \right) \end{aligned} \tag{3.9}$$

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. 3.9.

Variable $G_{ij}$ is used to represent the product of $S_{ij}$ and $F_{ij}$. Accordingly, (3.9) can be linearized as (3.10).

$$Q_i = P_s(tp_i) \cdot Q_{i-1} + P_f(tp_i) \cdot \sum_{j=1}^{TR_{\max}} G_{ij} \tag{3.10}$$

The complete ILP model is shown in Fig. 3.9.

### 3.4.2 Randomized Rounding

While the ILP model presented in Sect. 3.4.1 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 [24], 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.

## 3.5 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.

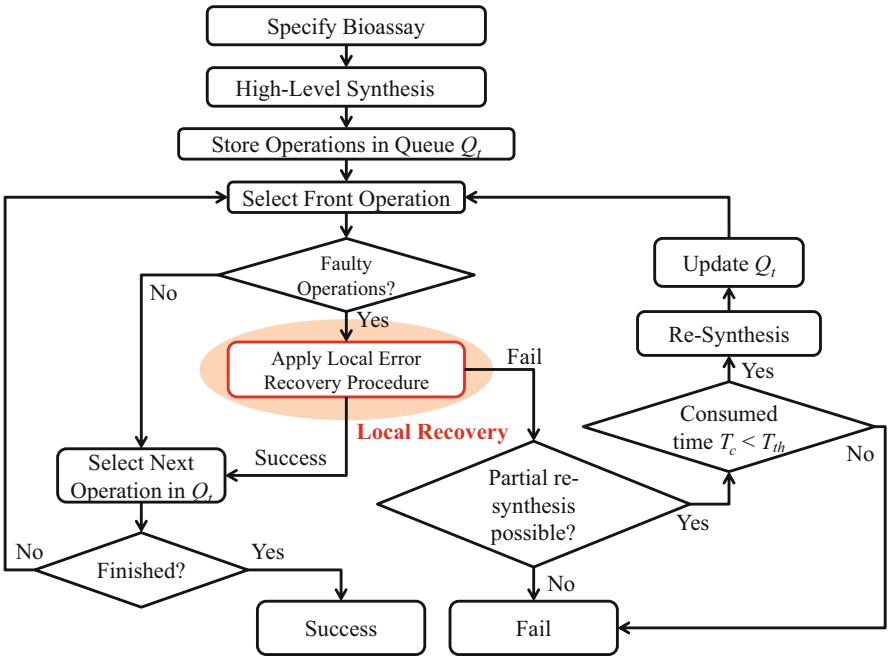



**Fig. 3.10** Control flow for error recovery

### *3.5.1 Control Flow*

The proposed control flow is shown in Fig. 3.10. 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 [22, 25]. 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) [3] 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 Sect. 3.5.2 to dynamically generate synthesis results when errors are detected and to calculate the bioassay completion time.

### *3.5.2 On-Line Synthesis*

As illustrated in Fig. 3.10, the recovery graph is generated and utilized to recover from the detected error when partial recovery synthesis is available [8]. 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 [26]. However, the method in [26] 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 [27]. 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.

#### 3.5.2.1 Adaptive Scheduler

Since error-recovery response time is critical, we use the greedy list-scheduling algorithm with computational complexity $O(n)$ [22], 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 [26], 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. An insertion example is shown in Fig. 3.11, in which operations $O_{r1} \sim O_{r7}$ are inserted droplet-routing operations. The priority for operation $O_i$ is the longest path (in time-steps) from $O_i$ to the sink node in the sequencing graph.

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.

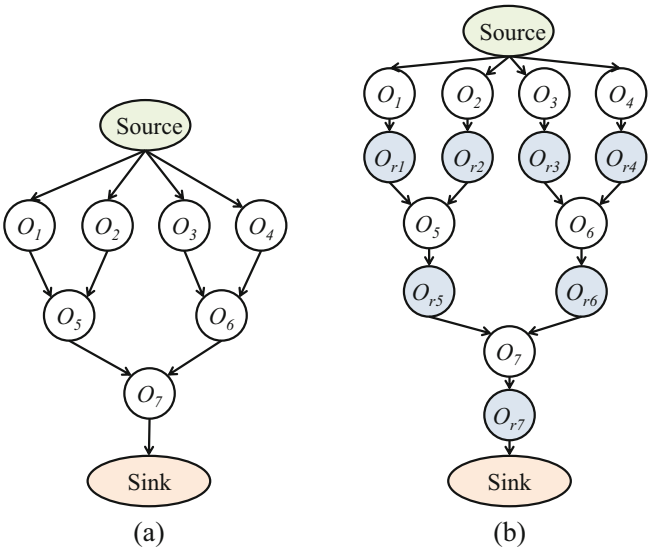



**Fig. 3.11** Sequencing graph (**a**) before and (**b**) after droplet-routing time insertion

#### 3.5.2.2 Adaptive Placer

A forbidden set (FS)-based module placement method is proposed in [26]. 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.

The proposed adaptive placer utilizes the method in [26] 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 [26]. 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.

#### 3.5.2.3 Adaptive Router

Droplet routing problem has been proved be NP-hard [28]. Several heuristic droplet-routing algorithms, including modified Lee algorithm [29], integer linear programming (ILP)-based algorithm [30], and soukup-based algorithm [31], have been proposed for conventional DMFBs. An A*-based algorithm is proposed for MEDA biochips [32]. However, all these algorithms take significant amount of time 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. 3.12. 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 [26, 33], the neighbors of one location $L_i$ include all locations vertically, horizontally, and diagonally adjacent to $L_i$. An example is shown in Fig. 3.12. 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 (1) fluidic constraint and (2) timing constraint. The timing constraint is introduced in Sect. 3.5.2. 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 [34]. 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.

```
Start: input start location S and end location E;
frontier = PriorityQueue()
frontier.put(S, 0)
came_from = { }
came_from[S] = None
while not frontier.empty( ) do
   current = frontier.get( )
   if current == T then
      output droplet route
      break
   end if
   for next in graph.neighbors(current) do
      if next not in came_from  &  not in obstacles then
         priority = Manhattan(T, next)
         frontier.put(next, priority)
         came_from[next] = current
      end if
   end for
end while
End: output derived droplet route.
```

**Fig. 3.12** Pseudocode for the adaptive router

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.

## 3.6 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.

### *3.6.1 Results for Local Faults*

Probability computation in PTAs can be carried using a recursive procedure [35]; therefore, as discussed in Sect. 3.3, we utilize the PRISM model checker to calculate the probability of success (POS) for each type of local error.

**Table 3.3** Probability of success for different local strategies[a]

| | Probability of success with different time cost constraint t | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Local recovery | 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 |

[a]N/A denotes the fact that the local recovery cannot be completed with the corresponding time constraint

Some types of samples, e.g., fibronectin, are known to degrade within 10 s [36]. 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 3.3 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 3.3. As explained in Sect. 3.5, 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 3.3 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 3.3.

**Table 3.4** Experimentally characterized MEDA module library for synthesis

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

### *3.6.2 Results for Bioassays*

We next present simulation results for three real-life benchmarks, namely PCR [37], in-vitro diagnostics [28], and protein dilution [27], 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 3.4. Without any loss of generality, we first set the size of one electrode in conventional DMFBs to be equal to a 4 × 4 microelectrode array in MEDA-based biochips; therefore, a 2 × 2 array in Table 3.4 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 300 times, and then calculate the corresponding completion time and POS for different time limits on local recovery (TLLR). Similar to $t_{th}$ in Sect. 3.3.4, 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 Sect. 3.4 is presented in Sect. 3.6.3. 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 Figs. 3.13, 3.14, and 3.15. 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 Figs. 3.13, 3.14, and 3.15, 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 [6] and [3], we randomly injected errors into the three benchmarks 300 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.

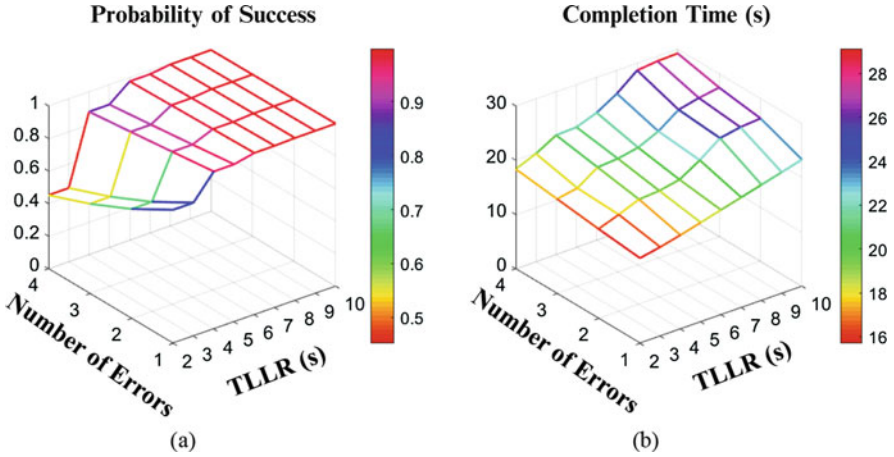



**Fig. 3.13** Illustration of (**a**) the probability of success and (**b**) the completion time as the TLLR and the number of inserted errors are varied for the PCR benchmark

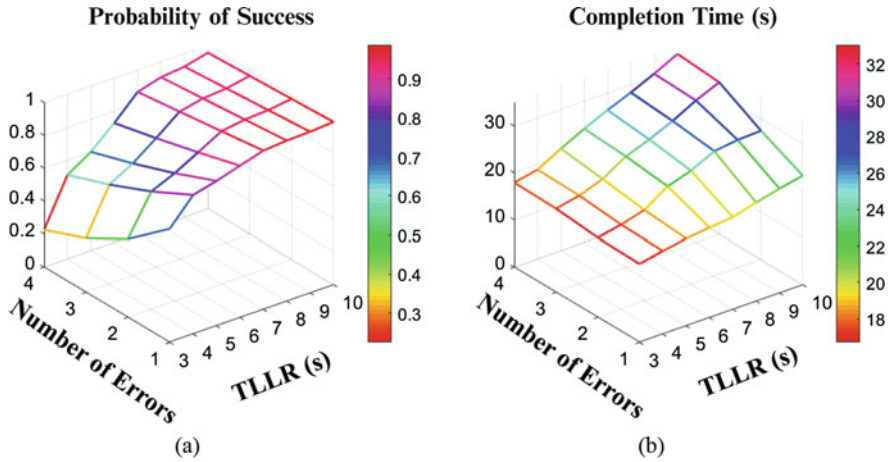



**Fig. 3.14** Illustration of (**a**) the probability of success and (**b**) the completion time as the TLLR and the number of inserted errors are varied for the in-vitro diagnostic benchmark

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 Sect. 3.3 for the computation. The results of this comparison are presented in Fig. 3.16. As the number of errors increases, the POS of the proposed method falls slowly while the POS for [6] falls rapidly for all benchmarks. Moreover, the proposed method also takes less completion time for most cases compared to [6].

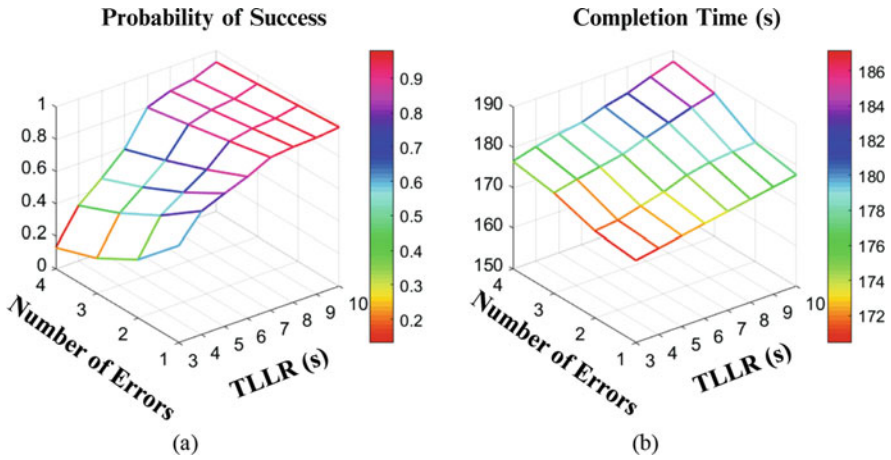



**Fig. 3.15** Illustration of (**a**) the probability of success and (**b**) the completion time as the TLLR and the number of inserted errors are varied for the protein dilution benchmark

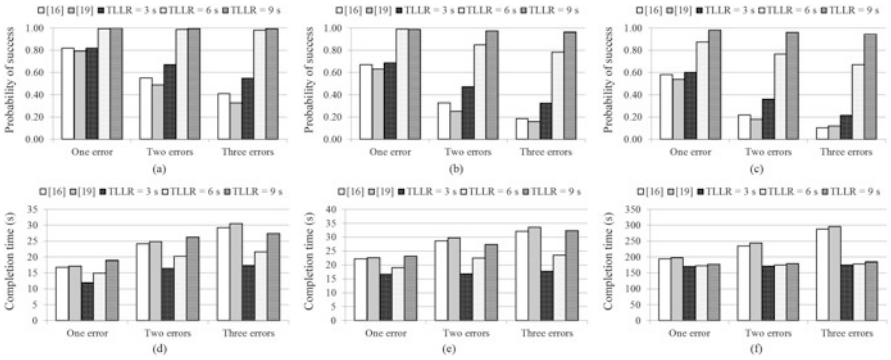



**Fig. 3.16** Comparison between the proposed method with previous methods [6] and [3] on (**a**) the POS for PCR benchmark, (**b**) the POS for in-vitro diagnostic benchmark, (**c**) the POS for protein dilution benchmark, (**d**) the completion time for PCR benchmark, (**e**) the completion time for in-vitro diagnostic benchmark and (**f**) the completion time protein dilution benchmark

### *3.6.3 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 Sect. 3.4 (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 Tables 3.5, 3.6, and 3.7, 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

**Table 3.5** Comparison between the ILP-based and the unified TLLR selection method for PCR benchmark

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

**Table 3.6** Comparison between the ILP-based and the unified TLLR selection method for In-Vitro diagnostic benchmark

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

**Table 3.7** Comparison between the ILP-based and the unified TLLR selection method for protein dilution benchmark

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

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 the corresponding benchmark.

As shown in Tables 3.5, 3.6, and 3.7, 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-based 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.

## 3.7 Conclusion

This chapter presents an efficient and adaptive 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. Li, Z., Lai, K. Y.-T., Yu, P.-H., Chakrabarty, K., Pajic, M., Ho, T.-Y., et al. (2016). Error recovery in a micro-electrode-dot-array digital microfluidic biochip. In *Proceedings of IEEE/ACM International Conference on Computer-Aided Design*, pp. 105:1–105:8.
2. Jaress, C., Brisk, P., & Grissom, D. (2015). Rapid online fault recovery for cyber-physical digital microfluidic biochips. In *Proceedings of IEEE VLSI Test Symposium*, pp. 1–6.
3. Zhao, Y., Xu, T., & Chakrabarty, K. (2010). Integrated control-path design and error recovery in the synthesis of digital microfluidic lab-on-chip. *ACM Journal on Emerging Technologies in Computing Systems, 6*(3), 11.
4. Verheijen, H., & Prins, M. (1999). Reversible electrowetting and trapping of charge: Model and experiments. *Langmuir, 15*, 6616–6620.
5. Yoon, J.-Y., & Garrell, R. L. (2003). Preventing biomolecular adsorption in electrowetting-based biofluidic chips. *Analytical Chemistry, 75*, 5097–5102.
6. Luo, Y., Chakrabarty, K., & Ho, T.-Y. (2013). Error recovery in cyberphysical digital microfluidic biochips. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 32*(1), 59–72.
7. Alistar, M., Pop, P., & Madsen, J. (2015). Redundancy optimization for error recovery in digital microfluidic biochips. *Design Automation for Embedded Systems, 19*(1–2), 129–159.
8. Alistar, M., & Pop, P. (2012). Online synthesis for error recovery in digital microfluidic biochips with operation variability. In *Proceedings of Symposium on Design, Test, Integration and Packaging of MEMS/MOEMS*, pp. 53–58.
9. Ibrahim, M., & Chakrabarty, K. (2015). Efficient error recovery in cyberphysical digital-microfluidic biochips. *IEEE Transactions on Multi-Scale Computing Systems, 1*(1), 46–58.
10. Lai, K. Y.-T., Yang, Y.-T., & Lee, C.-Y. (2015). An intelligent digital microfluidic processor for biomedical detection. *Journal of Signal Processing Systems, 78*(1), 85–93.
11. Norman, G., Parker, D., & Sproston, J. (2013). Model checking for probabilistic timed automata. *Formal Methods in System Design, 43*, 164–190.
12. Ho, T.-Y., Chakrabarty, K., & Pop, P. (2011). Digital microfluidic biochips: Recent research and emerging challenges. In *Proceedings of ACM/IEEE International Conference on Hardware/Software Codesign and System Synthesis*, pp. 335–343.
13. Chakrabarty, K., & Su, F. (2006). *Digital microfluidic biochips: Synthesis, testing, and reconfiguration techniques*. Boca Raton: CRC Press.

14. Su, F., Ozev, S., & Chakrabarty, K. (2003). Testing of droplet-based microelectrofluidic systems. In *Proceedings of IEEE International Test Conference*, pp. 1192–1200.
15. Su, F., Chakrabarty, K., & Fair, R. B. (2006). Microfluidics-based biochips: Technology issues, implementation platforms, and design-automation challenges. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 25*(2), 211–223.
16. Xu, T., & Chakrabarty, K. (2007). Functional testing of digital microfluidic biochips. In *Proceedings of IEEE International Test Conference*, pp. 1–10.
17. Fair, R. B., Khlystov, A., Tailor, T. D., Ivanov, V., Evans, R. D., Griffin, P. B., et al. (2007). Chemical and biological applications of digital-microfluidic devices. *IEEE Design and Test of Computers, 24*(1), 10–24.
18. Zhao, Y., & Chakrabarty, K. (2009). On-line testing of lab-on-chip using reconfigurable digital-microfluidic compactors. *International Journal of Parallel Programming, 37*(4), 370–388.
19. Lai, K. Y.-T., Shiu, M.-F., Lu, Y.-W., Ho, Y.-C., Kao, Y.-C., Yang, Y.-T., et al. (2015). A field-programmable lab-on-a-chip with built-in self-test circuit and low-power sensor-fusion solution in 0.35 $\mu$m standard cmos process. In *Proceedings of IEEE Asian Solid-State Circuits Conference*, pp. 1–4.
20. Fair, R., Srinivasan, V., Ren, H., Paik, P., Pamula, V., & Pollack, M. (2003). Electrowetting-based on-chip sample processing for integrated microfluidics. In *Proceedings of IEEE International Electron Devices Meeting*, pp. 32–35.
21. Brémaud, P. (2013) *Markov Chains: Gibbs Fields, Monte Carlo Simulation, and Queues*. New York: Springer Science and Business Media.
22. Su, F., & Chakrabarty, K. (2008). High-level synthesis of digital microfluidic biochips. *ACM Journal on Emerging Technologies in Computing Systems, 3*(4), 1.
23. Kwiatkowska, M., Norman, G., & Parker, D. (2011). Prism 4.0: Verification of probabilistic real-time systems. In *Proceedings of International Conference Computer Aided Verification*, pp. 585–591.
24. Bertsimas, D., & Tsitsiklis, J. (1997). *Introduction to linear optimization*. Belmont: Athena Scientific.
25. Luo, Y., Chakrabarty, K., & Ho, T.-Y. (2012). A cyberphysical synthesis approach for error recovery in digital microfluidic biochips. In *Proceedings of IEEE/ACM Design, Automation and Test Conference in Europe*, pp. 1239–1244.
26. Li, Z., Lai, K. Y.-T., Yu, P.-H., Ho, T.-Y., Chakrabarty, K., & Lee, C.-Y. (2016). High-level synthesis for micro-electrode-dot-array digital microfluidic biochips. In *Proceedings of ACM/IEEE Design Automation Conference* (pp. 146:1–146:6). New York: ACM.
27. Su, F., & Chakrabarty, K. (2005). Unified high-level synthesis and module placement for defect-tolerant microfluidic biochips. In *Proceedings of ACM/IEEE Design Automation Conference*, pp. 825–830.
28. Su, F., Hwang, W., & Chakrabarty, K. (2006). Droplet routing in the synthesis of digital microfluidic biochips. In *Proceedings of IEEE/ACM Design, Automation and Test Conference in Europe, 1*, pp. 1–6.
29. Zhao, Y., & Chakrabarty, K. (2012). Cross-contamination avoidance for droplet routing in digital microfluidic biochips. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 31*(6), 817–830.
30. Huang, T.-W., & Ho, T.-Y. (2011). A two-stage integer linear programming-based droplet routing algorithm for pin-constrained digital microfluidic biochips. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 30*, 215–228.
31. Roy, P., Rahaman, H., & Dasgupta, P. (2010). A novel droplet routing algorithm for digital microfluidic biochips. In *Proceedings of ACM Great Lakes Symposium on VLSI* (pp. 441–446). New York: ACM.
32. Chen, Z., Teng, D. H.-Y., Wang, G. C.-J., & Fan, S.-K. (2011). Droplet routing in high-level synthesis of configurable digital microfluidic biochips based on microelectrode dot array architecture. *Biochip Journal, 5*(4), 343–352.
33. Wang, G., Teng, D., & Fan, S.-K. (2011). Digital microfluidic operations on micro-electrode dot array architecture. *IET Nanobiotechnology, 5*(4), 152–160.

34. Li, Z., Lai, K. Y.-T., Yu, P.-H., Chakrabarty, K., Ho, T.-Y., & Lee, C.-Y. (2017). Droplet size-aware high-level synthesis for micro-electrode-dot-array digital microfluidic biochips. *IEEE Transactions on Biomedical Circuits and Systems, 11*(3), 612–626.
35. Trivedi, K. S. (2008). *Probability and statistics with reliability, queueing, and computer science applications*. New York: John Wiley and Sons.
36. Gefen, A. (2009). *Bioengineering research of chronic wounds: A multidisciplinary study approach*. New York: Springer Science and Business Media.
37. Fei, S. (2004). Module placement for fault-tolerant microfluidics-based biochips. *ACM Transactions on Design Automation of Electronic Systems, 11*(3), 682–710.

# Chapter 4
# Fault Modeling, Structural Testing, and Functional Testing


## 4.1 Chapter Overview

An potential roadblock in the deployment of MEDA biochips is the lack of test techniques to screen defective devices before they are used for biochemical analysis. MEDA allows increased integration of components; e.g., 1800 microelectrodes and the corresponding actuation/sensing circuits have been integrated in a MEDA biochip [1]. However, as in the case of integrated circuits, an increase in the density and area of microfluidic biochips will result in increased defects and reduced yield [2, 3]. In particular, defects are likely to occur during chip manufacturing and chip usage [4]. Therefore, effective test techniques are required before we can take advantage of high-level synthesis to map bioassay protocols to a MEDA biochip [1, 5].

Test techniques for conventional DMFBs can be classified as structural test methods [6] and functional test methods [7]. Traditional structural test methods use test droplets to traverse the target microfluidic array to detect and locate defect sites [6–8]. More recently, structural test methods for MEDA biochips have also been proposed [9–11], in which MEDA-specific fluidic operations have been exploited to reduce the test-completion time. However, structural test methods cannot provide a sufficient level of confidence that fluidic operations can be reliably performed on the chip [12]. For example, some microelectrodes may pass structural test, i.e., these microelectrodes can successfully actuate and maintain droplets, but they may malfunction (e.g., lead to unbalanced volumes) during droplet splitting. Therefore, it is important to carry out functional testing to verify the reliability of the biochip. In this chapter, we first present various defects and malfunctions that are typical for MEDA biochips, and relate these defects to fault models. We then propose efficient test techniques for both structural and functional test.

The remainder of the chapter is organized as follows. Section 4.2 presents a list of defects for MEDA biochips. We also relate these defects to fault models and observable errors. Section 4.3 discusses prior work on testing techniques for



Z. Li et al., *Micro-Electrode-Dot-Array Digital Microfluidic Biochips*,
https://doi.org/10.1007/978-3-030-02964-7_4

conventional digital microfluidic biochips. Sections 4.4 and 4.5 present the details of test techniques for structural and functional test, respectively. Section 4.6 presents simulation results using benchmarks and experimental results using fabricated MEDA biochips. Finally, Sect. 4.7 concludes the chapter.

## 4.2 Defects and Fault Modeling

As feature sizes scale down in MEDA biochips, the sizes of microelectrodes and distances between microelectrodes are also reduced to achieve higher levels of integration [4]. This increasing density increases the likehood of defects. Defects in MEDA biochips can be attributed to both fabrication processes and environmental reasons, such as imperfections in molds for top and bottom plates, pollutants, and failure in the hydrophobic-layer coating. Some typical defects and the corresponding fault models are listed below.

1. *Damage to the hydrophobic layer*: The hydrophobic layer can be damaged by chemical reaction or a physical scratch. A damaged hydrophobic layer, as shown in Fig. 4.1a, cannot provide sufficient electrowetting force when electrodes are actuated. The reduced force impedes droplet transportation. Accordingly, it can be modeled as electrode stuck-off, i.e., electrode actuation is not possible.
2. *Dielectric Breakdown*: High voltage during actuation causes dielectric breakdown, which can directly expose the droplet to high voltage. An example of dielectric breakdown is shown in Fig. 4.1b. In this case, droplet electrolysis, as shown in Fig. 4.1d, occurs and the droplet cannot be controlled. Dielectric breakdown can be modeled as droplet-electrode short, i.e., a short between the droplet and the electrode.
3. *Short-Circuited Microelectrodes*: A short between two adjacent microelectrodes leads to a "larger" macro-electrode. Once a droplet resides on this larger macro-electrode, we are no longer able to create the desired variation of interfacial surface tension along the droplet transportation path. An electrode short can be used to model this defect.
4. *Broken Wire*: A broken wire between droplet actuation circuits and microelectrodes can result in the failure of droplet transportation. A broken wire is usually caused by abnormal metal layer deposition or etch variation. Broken wire can be modeled as electrode stuck-off.
5. *Transistor Failure*: Transistor failure in actuation/sensing circuits can result in the malfunction of microelectrode functionalities, i.e., droplet actuation, maintenance, and sensing. An example of a transistor failure in the high-resolution capacitive sensing circuit [4] is shown in Fig. 4.1c. Traditional fault models, such as stuck-at-0 and stuck-at-1, can be used to model transistor failures in MEDA biochips.
6. *Parasitic Leakage*: Parasitic leakage in the actuation circuit can result in a degradation of the high-voltage level on the bottom microelectrode. Accordingly,

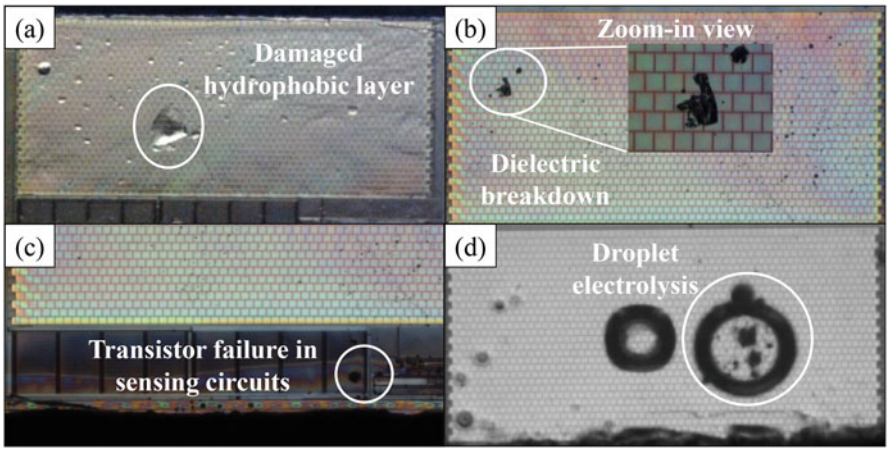



**Fig. 4.1** Images of some typical visible defects in a fabricated MEDA biochip. (**a**) Defect in the hydrophobic layer. (**b**) Defect in the dielectric layer. (**c**) Defective transistors in peripheral sensing circuits. (**d**) Droplet electrolysis

there is not enough voltage difference between the top and the bottom electrode, which leads to a failure in droplet actuation. This defect can be modeled as an electrode stuck-off.

Besides these typical defects, malfunctions in fluidic operations must also be addressed. Here we derive fault models for some typical malfunctions.

1. *Dispensing Failure*: A dispensing failure is caused by irreversible charge concentration on the dispensing electrode. If a dispensing failure occurs, no droplet can be dispensed from the reservoir. This failure can be modeled as dispensing stuck-on, i.e., a droplet is dispensed but is not fully cut off from the reservoir.
2. *Transportation Failure*: Droplet transportation failure can be induced by the defects listed above. These defects result in insufficient electrowetting force when microelectrodes are actuated. Therefore, transportation failure can be modeled as electrode stuck-off.
3. *Mixing Failure*: On MEDA biochips, two droplets are merged at one location and are either (1) routed to move or (2) repetitively split and merged to accelerate the mixing procedure. Details of the mixing methods on MEDA biochips are described in Sect. 4.5.3. Mixing failure is usually induced by the failure of droplet actuation. Therefore, electrode stuck-off can also be used to model this malfunction.
4. *Splitting Failure*: Charges can be trapped within microelectrodes [13], which may result in unequal actuation voltages and eventually cause splitting failure. The observable error corresponding to a splitting failure is the unbalanced volumes of the droplets obtained from the split operation. The electrowetting force gradient can be used to model the splitting failure.

## 4.3 Prior Works

The testing of conventional DMFBs has been investigated over the past decade [6–8, 14–19]. A unified test methodology for DMFBs was presented in [15], whereby faults can be detected by controlling and tracking droplet motion. Most catastrophic faults result in a complete cessation of droplet transportation. Therefore, the status of the tested DMFB can be simply determined by observing the arrival of test droplets at the "sink" location. Reference [16] modeled the microfluidic array for a DMFB as an undirected graph, and the pathway for the test droplet can be determined by solving a Hamiltonian path problem. The method in [17] proposed a solution based on Euler paths for detecting electrode shorts. In order to further reduce test cost, a test methodology, referred to as parallel scan-like test, has been proposed in [18]. This method manipulates multiple test droplets in parallel to traverse the target DMFB.

A drawback of the above test methods is that they only focus on physical defects. These methods can only guarantee that a DMFB is defect-free, but they cannot guarantee that the DMFB is malfunction-free. Therefore, functional test methods have been developed in [7] to test fundamental fluidic operations on conventional DMFBs. A built-in self-test (BIST) method has also been proposed in [8]. This method utilizes digital microfluidic logic gates (AND, OR, and NOT gates) to compress test-outcome droplets into a single droplet signature, which obviates the need for capacitive sensing test-outcome circuits for analysis.

To ensure that manufactured biochips are competitive in the emerging low-cost market for disposable biochips, test methods should be inexpensive, quick, and effective. However, test droplets are required for all previous methods. Generating and manipulating test droplets takes a considerable amount of time, thus preventing the possibility of a quick test. Furthermore, previously developed methods use external capacitive sensing circuits to read and analyze test outcomes, which is especially prone to errors arising from inaccuracies in sensor calibration. Moreover, because of intrinsic differences between MEDA biochips and conventional DMFBs, MEDA-specific operations, such as diagonal splitting and real-time droplet sensing, cannot be tested using previous test methods. Therefore, to ensure robust execution of the target bioassay, we must develop more comprehensive test techniques for MEDA biochips, which not only target defective cells, but also lead to the detection of malfunctioning microfluidic modules.

## 4.4 Structural Test for MEDA Biochips

In this section, we propose test techniques that targets the physical defects discussed in Sect. 4.2. These defects affects the functionality of the unit cell (microelectrode). Because the proposed test technique aims to ensure structural integrity of the

MEDA biochip, here we refer to the test procedures as structural test. The proposed structural test techniques eliminate the usage of test droplets, which leads to reduced test time.

### 4.4.1 Overview of Test Method

Due to the high integration level in MEDA biochips, microelectrode cells (MCs) are connected together using a scan-chain structure, i.e., the output of one MC is connected to the input of the next MC [20]. An illustration of the scan-chain structure is shown in Fig. 4.2. An actuation pattern can be applied serially to a MEDA biochip through the scan chain, and the droplet detection results can also be clocked into scan registers first and then serially shifted out. This scan-chain structure can significantly simplify wiring and increase design scalability. Using this scan-chained control structure, the MEDA biochip can achieve 100% individual controllability of each MC with a minimum number of control pins. In the state-of-the-art MEDA design [20], there are only seven ports in a chip, namely $CONT$, $SET$, $\overline{RESET}$, $ACT$, $ACT\_b$, $IN$, and $OUT$ (see Fig. 4.3).

With the help of the scan chain, we are also able to efficiently apply test patterns and capture test responses for a MEDA biochip. Since there is only a limited number of pins available to access the actuation/sensing circuit for each MC, a test structure is proposed and incorporated in each MC to alleviate the difficulty and complexity of chip testing. Details of the proposed test structure is described in

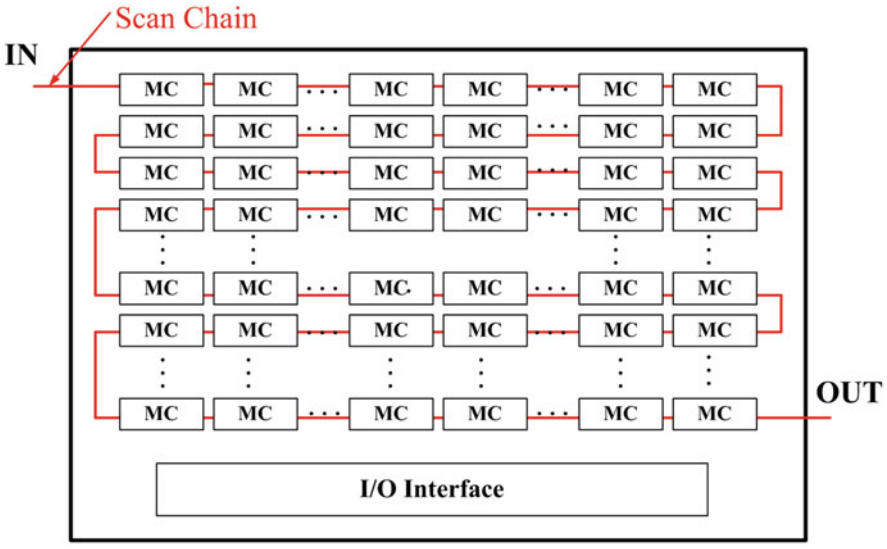



**Fig. 4.2** Illustration of the scan-chain structure in a MEDA biochip

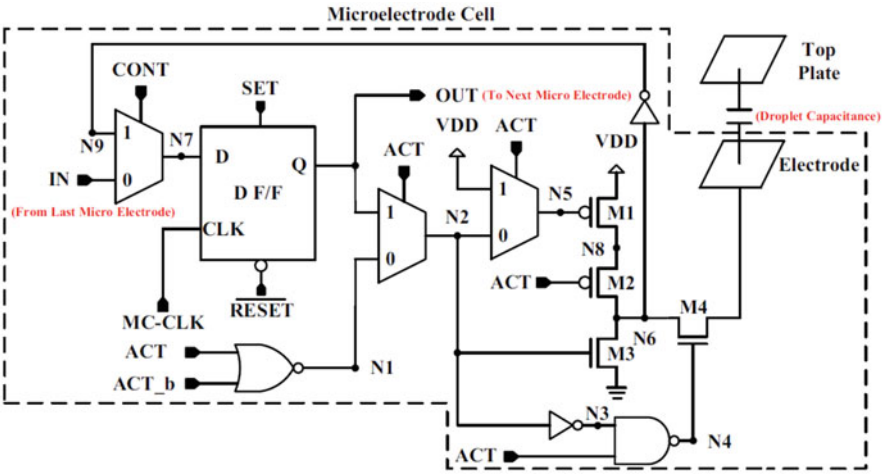



**Fig. 4.3** Illustration of the actuation/sensing circuit in each MC [20]

Sect. 4.4.3. Accordingly, three elements make up the updated MC: a microelectrode, an actuation/sensing circuit, and a specially designed test structure.

The proposed structural test covers two categories of tests: scan-chain test and MC test. The scan-chain test helps to guarantee that a test pattern can be successfully loaded into each MC; the MC test checks the three main functionalities (droplet maintenance, droplet actuation and droplet sensing) of each MC. During scan-chain testing, scan registers are triggered, and other control signals do not influence the operations of these registers. The test responses shifted out of the scan chain are compared with the fault-free responses to detect faults in the scan chain. During MC testing, specially generated test patterns are applied to the MEDA biochip. According to the proposed test structure in Sect. 4.4.3, the fault-free test response of the chip is exactly the same as the test input. The test output can then be compared with the fault-free test response to determine the faulty/fault-free status of the MEDA biochip.

### *4.4.2 Scan-Chain Test Strategy*

It has been reported that 10–30% of all defects in integrated circuits (ICs) can cause scan chains to fail [21] and scan chain failure accounts for almost 50% of chip failures in ICs [22]. Therefore, a scan-chain test is required to ensure that test patterns can be successfully applied to the MEDA biochip.

Scan-chain fault models typically include functional faults and timing faults. For targeting functional faults, the stuck-at (stuck-at-0 and stuck-at-1) and bridging fault models are commonly used for scan-chain tests [21]; For detecting timing

problems, "slow faults" (slow-to-rise and slow-to-fall) and "fast faults" (fast-to-rise and fast-to-fall) are considered [22]. Slow faults and fast faults result from setup time violations and hold-time violations, respectively. Setup time violation occurs due to *too-late signal arrival* at a scan flip-flop's input, while hold-time violation is due to *too-early signal change* at a scan flip-flop's output.

Testing of the scan chain in MEDA biochips can be done using *shift test* (*flush test*) [23]. The MEDA biochip is set in scan mode by setting $CONT = 0$. Now all registers form a shift register between $IN$ and $OUT$. Three flush test patterns are utilized to test both functional faults and timing faults in the scan chain.

1. First, test pattern 1111... of length $n_{sff}$, where $n_{sff}$ is the total number of flip-flops in the MEDA biochip, is applied to detect all stuck-at-0 faults;
2. Next, test pattern 0000... of length $n_{sff}$, is applied to detect all stuck-at-1 faults in the MEDA biochips;
3. Finally, test pattern 00110011... of length $n_{sff}+4$, is applied at $IN$. The pattern is clocked through the scan chain using the functional clock. This test pattern generates all four transitions, $0 \rightarrow 0$, $0 \rightarrow 1$, $1 \rightarrow 1$ and $1 \rightarrow 0$, in each flip-flop and shifts the output to the observable output $OUT$. This pattern is able to detect all timing faults in the scan chain.

Since fault-free outputs are exactly the same as the input test pattern during the scan-chain testing, we can compare the output with the test pattern to detect faults in the scan chain.

### *4.4.3 MC Test Strategy*

The MC test can be used to evaluate the following operations for each MC: (1) droplet actuation, (2) droplet maintenance, and (3) droplet sensing. For (1) and (2), we must test whether the bottom microelectrode can be driven to the HV value (e.g., 25 V [1, 3] for droplet maintenance), or to the ground voltage for droplet actuation.

In this subsection, we first develop a SPICE macro-model for the simulation of the EDMOS in MEDA biochips. Therefore, both the low- and high-voltage transistors can be simulated simultaneously. We then propose a capacitive voltage divider (CVD) to regulate the voltage on the microelectrode in the operating voltage range (e.g., 0 V to $V_{DD}$). Finally, we describe the strategy for MC test.

#### 4.4.3.1 SPICE Model for EDMOS

Figure 4.4a shows the structure of the EDMOS used in MEDA biochips [4]. The $P+$ field ring technique and local oxidation of silicon (LOCOS) are used to create the buffer region between the drain and the gate, which results in a large drain resistance ($R_d$) to significantly increase the avalanche breakdown voltage [24]. Based on

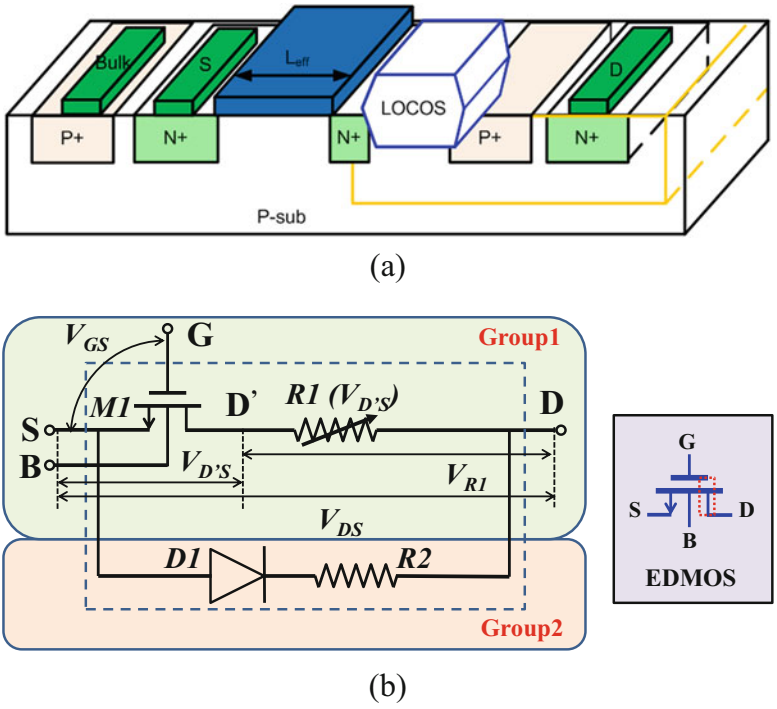



**Fig. 4.4** (**a**) Illustration of the structure of EDMOS [4]. (**b**) Illustration of the proposed SPICE macro-model and the corresponding symbol

the experiments carried out using EDMOS, it has been reported that the average saturation current is around 65 μA and the breakdown voltage is around 25 V [4].

The modeling of CMOS-compatible HV transistors has been widely studied using the BSIM3v3 SPICE model [25]. Therefore, we propose a SPICE macro-mode for the EDMOS in MEDA biochips. As shown in Fig. 4.4b, the macro-model consists of two groups: (1) Group 1, including a standard NMOS transistor ($M1$) and a voltage-controlled resister ($R1$), and (2) Group 2, which includes a reverse-biased diode ($D2$) and a resistor ($R2$). Group 1 is used for simulating the EDMOS in both linear and saturation region; Group 2 is modified from the model in [26], and is used to simulate the breakdown behavior of the EDMOS.

We describe how to calculate the parameters (e.g., the resistance of $R2$ and the width-to-length ratio of $M1$) used in the proposed macro-model. When the EDMOS is forward-biased, it can be either in the saturation region or in the linear region. If the EDMOS is in the saturation region, $M1$ should also be in the saturation region. Since the channel-length modulation parameter for the 0.35 μm process is extremely small (e.g., 0.05 μm/V for an n-type device [27]), channel-length modulation has little effect on the drain current. Therefore, the saturation current $I_{sat}$ for $M1$ can be expressed using (4.1).

$$I_{sat} = \frac{1}{2}\mu_n C_{ox}(W/L)(V_{GS} - V_T)^2 \tag{4.1}$$

where $\mu_n$ represents the electron mobility, $C_{ox}$ denotes the oxide capacitance per unit area, $W/L$ denotes the width-to-length ratio of $M1$, $V_{GS}$ represents the voltage between the gate and the source of $M1$, and $V_T$ is the threshold voltage of $M1$.

Since $D1$ is reverse-biased when the EDMOS is forward-biased, the current going through $D2$ is extremely small (e.g., only in the nA range [28]) and it can be neglected. Therefore, the saturation current of $M1$ and the EDMOS can be assumed to be the same. In the proposed macro-model, a first-order polynomial approximation is used for the voltage-controlled resistor $R1$, i.e., $R1$ can be expressed as $R_1 = k_1 V_{D'S} + k_0$. Parameter $V_{D'S}$ denotes the drain-source voltage of $M1$; $k_1$ and $k_0$ are two constants. The drain-source voltage of the EDMOS ($V_{DS_s}$) can be expressed using (4.2) if the EDMOS is in the saturation region.

$$V_{DS_s} = I_{sat}(k_1 V_{D'S_s} + k_0) + V_{D'S_s} \tag{4.2}$$

where $V_{D'S_s}$ is the drain-source voltage of $M1$ when the EDMOS is in the saturation region.

Similarly, we have Eqs. (4.3) and (4.4) if the EDMOS is in the linear region.

$$I_{lin} = \frac{1}{2}\mu_n C_{ox}(W/L)\left[2(V_{GS} - V_T)V_{D'S_l} - V_{D'S_l}^2\right] \tag{4.3}$$

$$V_{DS_l} = I_{lin}(k_1 V_{D'S_l} + k_0) + V_{D'S_l} \tag{4.4}$$

where $I_{lin}$ and $V_{DS_l}$ denote the drain current and the drain-source voltage, respectively, of the EDMOS when the EDMOS is in the linear region, and $V_{D'S_l}$ represents the corresponding drain-source voltage of $M1$.

As reported in [4], if $V_{GS} = 3.3$ V and $V_{DS_s} = 5$ V, $I_{sat}$ is 65 μA; if $V_{GS} = 3.3$ V and $V_{DS_l} = 1$ V, $I_{lin}$ is 25 μA. Parameters $\mu_n$, $C_{ox}$ and $V_T$ are determined by the standard TSMC 0.35 μm process. After substituting these values into Eqs. (4.1)–(4.4), the parameters $W/L$, $k_0$ and $k_1$ are calculated to be 0.13, $16.71\times10^3$ and $12.14\times10^3$, respectively.

In the macro-model, diode $D1$ and resistor $R2$ are used to simulate the breakdown behavior of the EDMOS. As reported in [4], the drain-source breakdown voltage of the EDMOS is 25 V ($V_{DS}$=25 V) when $V_{GS}$ is 0 V. The breakdown current is measured to be 10 nA. When $V_{DS}$ is larger than 25 V, the drain current $I_{DS}$ increases linearly with $V_{DS}$ and the slope is around $14.28\times10^3$. Therefore, the breakdown voltage and saturation current of the SPICE model for $D1$ are set to be 25 V and 10 nA, respectively; the resistor R2 is set to be 14.28 kΩ.

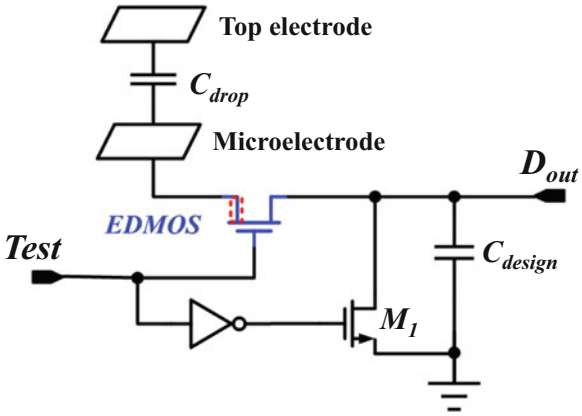



**Fig. 4.5** Proposed CVD design for voltage regulation

#### 4.4.3.2 CVD Design

As described in Chap. 1, the microelectrode is driven to the high voltage value (e.g., 25 V [4, 20]) for droplet maintenance or to the ground voltage for droplet actuation. However, the voltage signal on the microelectrode cannot be directly latched; otherwise, high voltage may damage the actuation/sensing circuit. Therefore, we propose a CVD design to regulate the voltage into the operation voltage range. The proposed design is shown in Fig. 4.5.

The working principle of the proposed CVD is described as follows. When $TEST = 0$, the CVD works in the non-test mode. In this mode, the EDMOS is off and transistor $M1$ is on. Therefore, the current going through the EDMOS is negligible, and the voltage on the microelectrode is not influenced. Output $D_{out}$ is also zero in the non-test mode. When $TEST = 1$, the CVD works in the test mode. In this mode, the EDMOS is on and transistor $M1$ is off. If the voltage on the microelectrode is zero, $D_{out}$ is also zero; if the voltage on the microelectrode is a high voltage $V_{HV}$, $D_{out}$ can be estimated to be $V_{HV}C_{drop}/(C_{drop} + C_{design})$ if we neglect the parasitic capacitances of the EDMOS and $M1$. Here $C_{drop}$ and $C_{design}$ represent the capacitance between the top electrode and the microelectrode and the capacitance of the designed capacitor, respectively. Therefore, we are able to identify the voltage on the microelectrode based on the value of $D_{out}$.

The droplet capacitance $C_{drop}$ can be calculated using the relationship $C_{drop} = \varepsilon_0\varepsilon_d L^2/d$, where $\varepsilon_0$ represents the permittivity of vacuum, $\varepsilon_d$ is the relative permittivity of the filler medium (silicone oil), $L$ denotes the microelectrode pitch, and $d$ is the gap height between the top electrode and the bottom microelectrode. The capacitance $C_{drop}$ is calculated to be 2.21 fF when $L$ and $d$ are 50 and 25 μm, respectively. In our design, we set $C_{design}$ to be 22.13 fF. Therefore, $D_{out}$ is 2.27 V when the microelectrode is driven to 25 V. Output $D_{out}$ can then be latched and used as a flag to indicate whether the microelectrode is driven to the high voltage value or to the ground. Note that the target technology (TSMC 0.35 μm process)

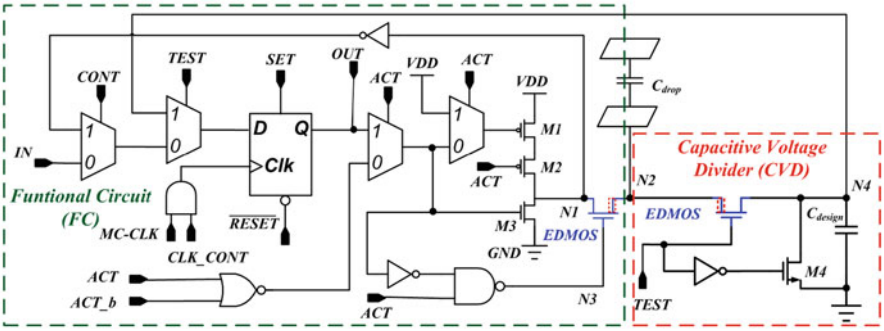



**Fig. 4.6** Illustration of the circuit diagram with capacitive voltage divider for each MC

offers Poly-insulator-Poly (PiP) capacitors with 864 aF/$\mu$F$^2$ capacitance per unit area. Hence the value of 22.13 fF for the designed capacitance gives a reasonable chip area of 24.31 $\mu$m$^2$.

#### 4.4.3.3 Test Strategy

We now integrate the CVD with the original actuation/sensing circuit. A multiplexer, controlled by the signal $TEST$, is also added to select the input for the D flip-flop. The new circuit diagram is presented in Fig. 4.6. Note that the circuit size for each MC is constrained by the size of the microelectrode ($50\times50$ $\mu$m [4]) because the circuit is located directly under the microelectrode. The silicon area for the design in Fig. 4.6 is 2071.5 $\mu$m$^2$, which satisfies the area constraint.

When $TEST = 0$, the MC works in the functional mode. The EDMOS in the CVD is off; therefore, the functional circuit of the MC (enclosed by the green dotted lines) is not influenced. When $TEST = 1$, the MC works in the test mode. In this mode, V($N4$) (the output of the proposed CVD design) of each MC can be captured and shifted out through the scan chain. Based on the voltage value of V($N4$), we are able to determine whether droplet actuation and droplet maintenance are successfully achieved on the MEDA biochip. We can next carry out the droplet sensing for each MC. Based on the sensing results, we can identify the microelectrodes with malfunctions in droplet sensing. Detailed strategies for MC test are described as follows.

- **Test of Droplet Actuation:** The key step in droplet-actuation test is to determine whether the microelectrode can be driven to the ground voltage. The voltage on $N2$, i.e., V($N2$), cannot be directly captured and stored because V($N2$) can be HV during droplet maintenance. Therefore, we capture and shift out the voltage on $N4$, i.e., V($N4$), instead. High voltage is applied on the top electrode for the test of droplet actuation. We first set $CONT$, $ACT$, and $TEST$ to logic 0, 1, and 0, respectively. Therefore, the MC now operates in functional mode.

We then enable the system clock and apply the test pattern 1111... of length $n$ to the biochip, where $n$ is the number of MCs. When the test pattern is applied, droplet actuation is executed for each MC. We then disable the clock and set $TEST$ to logic 1. Accordingly, all MCs start working in the test mode. If there is no fault involving droplet actuation, V($N4$) is supposed to be zero for every MC. We can then enable the system clock again to shift out the captured V($N4$) values from all MCs. If the output is 0000..., the chip passes the test.

- **Test of Droplet Maintenance:** Similar to the test of droplet actuation, the key in droplet-maintenance test is to determine whether the microelectrode can be successfully driven to high voltage. The only difference between the test of droplet actuation and of droplet maintenance is that we change the test pattern to 0000.... If there is no fault with droplet maintenance, V($N4$) is expected to be logic 1 based on the proposed CVD design. We can shift out the captured V($N4$) values from all MCs. If the output is 1111..., the chip passes the test.
- **Test of Droplet Sensing:** High voltage is disabled during droplet-sensing test. We first set $CONT$, $ACT$, and $TEST$ to logic 1, 1, and 0, respectively. We then switch $ACT$ to logic 0, and droplet sensing is invoked for each MC. Capacitor $C_{drop}$ is first charged through transistors $M1$ and $M2$ and then discharged through M3. Based on the differences in charging and discharging times, each MC can sense the capacitance of $C_{drop}$. Sensing results are stored as 1-bit data and are used to construct a 2D sensing map with a digital delay-locked-loop (DLL). If the sensing results of all MCs are the same, the chip passes the sensing test.

Through the MC test, microelectrodes with faults (abnormal microelectrodes) can be distinguished, and these microelectrodes will not be used during the execution of bioassays. At the same time, the abnormal microelectrodes will not be tested during functional testing.

## 4.5 Functional Test for MEDA Biochips

In this section, we propose a comprehensive test procedure, referred to as functional test, which targets fluidic operations on MEDA biochips. Functional testing checks whether a group of cells can be used to perform certain operations, e.g., droplet mixing and splitting. If the operation cannot be executed on the target fluidic module, the fluidic module is labeled as a malfunctioning module, which implies that the synthesis tool cannot configure any operation on this module. The proposed functional test is divided into dispensing test, routing test, mixing test, and splitting test.

### 4.5.1 *Dispensing Test*

The dispensing test focuses on testing the malfunctioning of dispensing operations. A reservoir on a MEDA biochip is intrinsically a group of microelectrodes that can hold certain volume of liquid. Another group of microelectrodes forms a functional electrode, which works as a dispensing port. Droplets can be dispensed from the reservoir through the dispensing port. Note that the size of the dispensing port is determined by the size of the dispensed droplet. Our implementation of a reservoir on a MEDA biochip is shown in Fig. 4.7. As shown in the figure, the reservoir in Fig. 4.7a consists of a group of microelectrodes, and the experimental demonstration of a dispensing cluster is also shown in Fig. 4.7b.

Figures 4.8 and 4.9 present a comparison between normal dispensing and a dispensing failure that we have observed and recorded. As shown in Fig. 4.8a, a

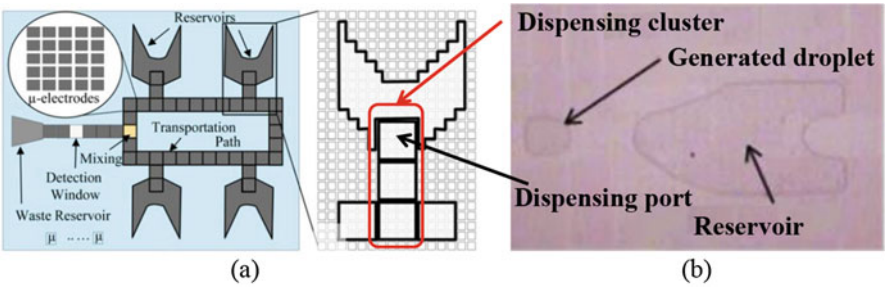



**Fig. 4.7** (**a**) Illustration of a configured reservoir on a MEDA biochip and (**b**) experimental demonstration of a reservoir on a fabricated MEDA biochip [29]

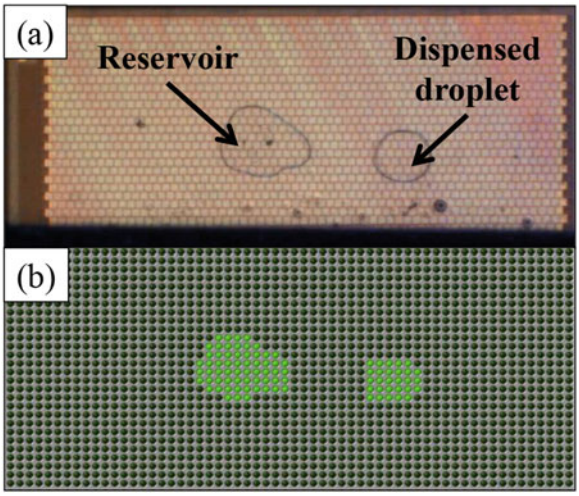



**Fig. 4.8** Illustration of (**a**) a normal dispensing and (**b**) the corresponding sensing map (green dots represent the corresponding microelectrodes are occupied by droplets)

droplet is successfully dispensed from the reservoir. In Fig. 4.9a, the droplet cannot be detached from the reservoir.

Here we propose a test method based on a droplet-sensing technique to detect the dispensing failure. It has been shown in the literature that dispensing involves a reservoir and three closest functional electrodes [7]. We define the three electrodes as a dispensing cluster. An example of a dispensing cluster is shown in Fig. 4.7a. Note that the functional electrode in Fig. 4.7a is intrinsically a 5×5 microelectrode array.

When the dispensing test starts, a droplet is dispensed from the reservoir until it reaches the third functional electrode in the dispensing cluster. At the same time, the real-time droplet-location sensing feedback can form a sensing map. Examples of sensing maps are presented in Figs. 4.8b and 4.9b. Based on the sensing map, we are able to determine whether the dispensed droplet is detached from the reservoir. If the dispensed droplet is still connected with the droplet in the reservoir (see Fig. 4.9b), the dispensing operation fails. Otherwise, we compare the size of the dispensed droplet (based on the sensing map) with the anticipated size. If the sensed size is close or equal to the anticipated size, the dispensing operation succeeds. An example is provided here. In Fig. 4.8b, the dispensed droplet is supposed to be a 5×5 droplet. Based on the sensing map, the dispensed droplet occupies 27 microelectrodes. Therefore, we regard the dispensing operation in Fig. 4.8 as a successful operation.

Dispensing test identifies the dispensing failure for reservoirs. Since reservoirs can be dynamically configured on a MEDA biochip, once a reservoir fails the dispensing test, the reservoir can be moved to another location.

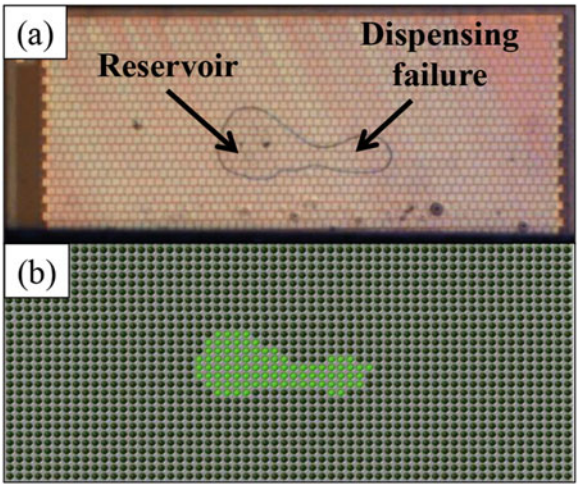



**Fig. 4.9** Illustration of (**a**) a dispensing failure and (**b**) the corresponding sensing map (green dots represent the corresponding microelectrodes are occupied by droplets)

### 4.5.2 *Routing Test*

The routing test focuses on the ability of functional electrodes to transport droplets. Based on the droplet size, the complete MEDA biochip can be divided into a number of functional electrodes. Note that the microelectrodes within a functional electrode are always activated or deactivated at the same time. During routing test, a test droplet is dispensed and routed to traverse all functional electrodes.

An example of routing test is shown in Fig. 4.10. The abbreviation FE stands for functional electrode. In this example, a functional electrode consists of a $6\times6$ microelectrode array. The routing test in Fig. 4.10 identifies the routing failure from FE2 to FE3.

Similar to the procedure proposed for structural test for conventional DMFBs [15, 17], the routing problem can be solved by mapping the functional electrode arrays to an undirected graph and applying the Euler-path-based method. An example is shown in Fig. 4.11. In Fig. 4.11a, a microelectrode array is divided into nine functional electrodes. Figure 4.11b presents the derived undirected graph for conventional DMFBs, i.e., only horizontal and vertical droplet transportation are supported. A node in Fig. 4.11b represents a functional electrode. In contrast to conventional DMFBs, MEDA biochips can also support diagonal droplet transportation. Therefore, a test droplet must be routable in eight directions relative to the target functional electrode to test one functional electrode; see Fig. 4.11a. The corresponding undirected graph for MEDA biochips is presented in Fig. 4.11c. We can then map the routing problem for functional test to a directed graph in Fig. 4.11d, which can be easily derived by replacing every edge in Fig. 4.11c

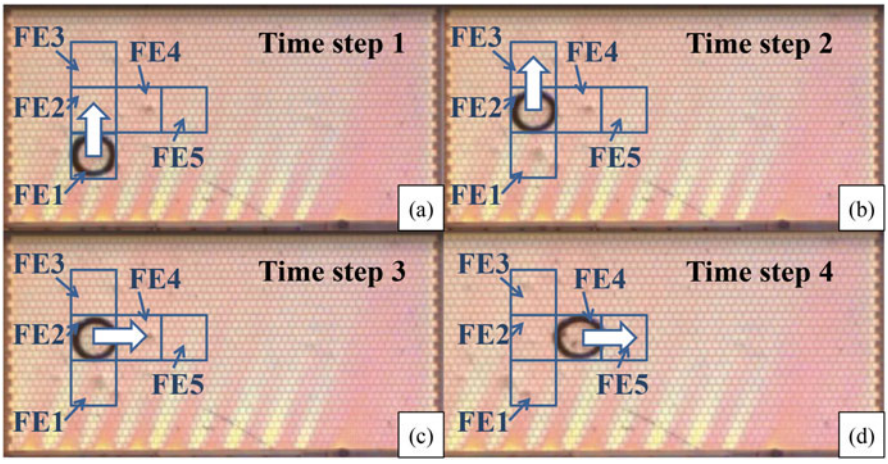



**Fig. 4.10** (**a**) Initial location of the test droplet, and FE2 is actuated; (**b**) The test droplet is successfully transported to FE2, and FE3 is actuated; (**c**) The test droplet cannot be transported to FE3, and the droplet changes its direction of motion; (**d**) The test droplet is successfully transported to FE4

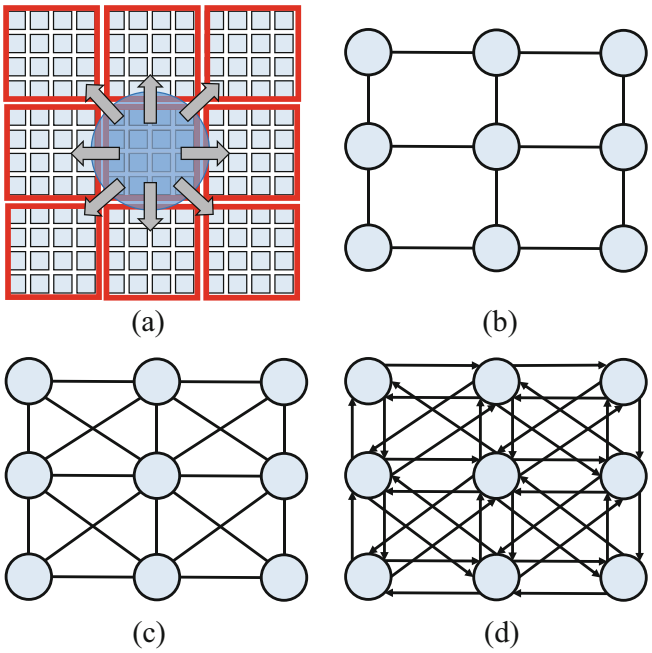



**Fig. 4.11** (**a**) The microelectrode array is divided into a 3×3 functional electrode array. (**b**) Undirected graph for conventional DMFBs. (**c**) Undirected graph for MEDA biochips. (**d**) Directed graph for the functional routing test

with two directed edges in the opposite directions. After the directed graph is constructed, Euler-path-based method [17] is applied to derive a functional test plan for the routing test.

Recall that the structural test procedure described in Sect. 4.4 identifies all faulty microelectrodes. If all microelectrodes in a functional electrode fail structural test, the functional electrode will not be tested during routing test. Routing test can detect all types of defects described in Sect. 4.2. The functional electrodes that fail routing test will be avoided during droplet-route design in high-level bioassay synthesis.

### *4.5.3 Mixing and Splitting Test*

Next we discuss functional testing of three widely used microfluidic operations: mixing, splitting, and dilution. Since a dilution operation can be regarded as a mixing operation followed by a splitting operation, we only need to test the mixing and splitting operations.

In contrast to conventional DMFBs, diagonal droplet splitting can be achieved on a MEDA biochip. An example of diagonal splitting achieved in the laboratory is shown in Fig. 4.12. Moreover, MEDA biochips can also support more advanced

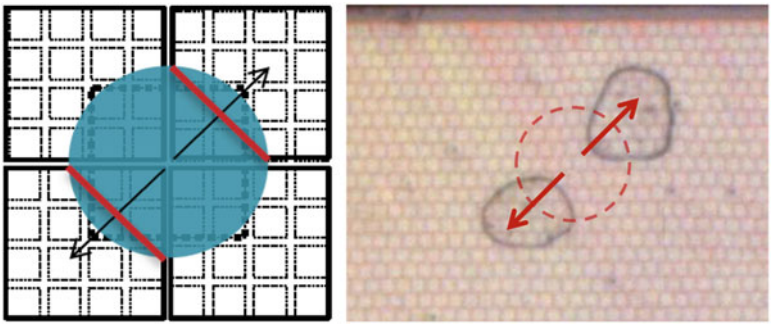


**Fig. 4.12** Illustration of a diagonal splitting on a fabricated MEDA biochip

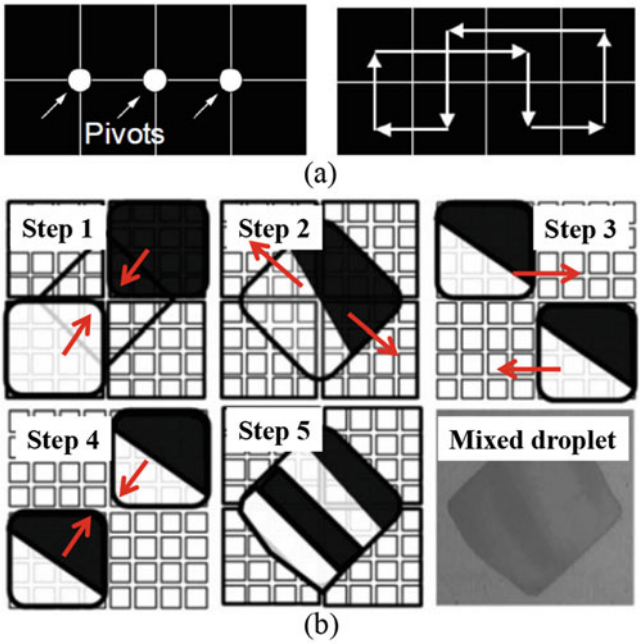



**Fig. 4.13** (**a**) Pivots and routing path within a 2×4 microfluidic mixer for routing-based mixing [7] and (**b**) different steps in one mixing cycle for SAR-based mixing [29]

mixing operations. On a conventional DMFB, two droplets are mixed within a cluster of electrodes, referred to as the mixer. Two droplets are merged at one electrode and routed to move around some pivots in the mixer, as shown in Fig. 4.13a. We refer to this mixing method as routing-based mixing method. Besides routing-based mixing method, MEDA can also support a novel split-and-recombination-based (SAR-based) mixing method, which can efficiently reduce mixing time [29]. Three basic steps, droplet splitting, droplet rearrangement, and droplet recombination, are

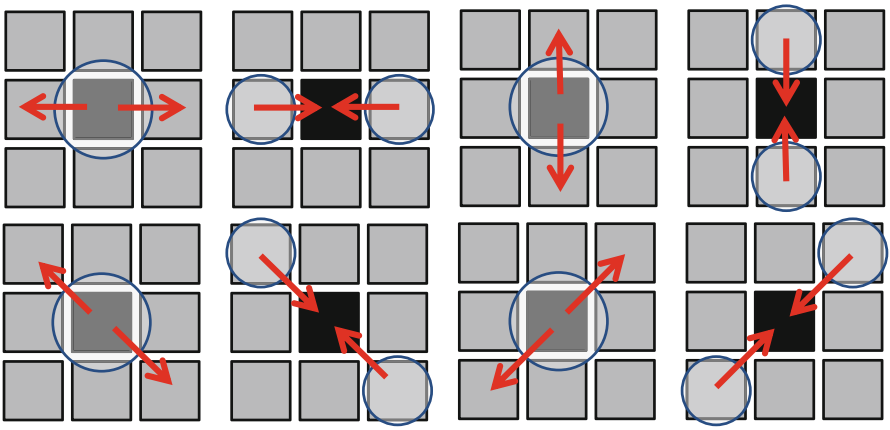


**Fig. 4.14** Merging-and-splitting test within a MAS cluster; the black square represents the target functional electrode while other grey squares represent the adjacent functional electrode in the MAS cluster

required for SAR-based mixing method. An illustration of SAR-based mixing is shown in Fig. 4.13b. In Fig. 4.13b, two droplets are repetitively split and merged to complete mixing operation.

Thus, a functional test for mixing is equivalent to the testing of merging, splitting, and routing operations within the target cluster cell. Recall that routing test has been discussed in Sect. 4.5.2; therefore, mixing test is reduced to a combination of merging test and splitting test, which checks (1) whether two 1X (unit) droplets can be merged and (2) whether one 2X droplets can be split into two 1X droplets on each functional electrode.

The test outcome is evaluated by the real-time droplet sensing results provided by MEDA. For merging test, if the sensing result does not demonstrate that two 1X droplets have been merged into a 2X droplet, the test fails. For a splitting test, if (1) the 2X droplets cannot be split to two droplets or (2) the sensed droplet sizes of two split droplets are not equal to each other, the test fails. Similar to routing test in Sect. 4.5.2, once a functional electrode fails structural test, it will not be targeted during mixing and splitting test.

We can combine the merging test and the splitting test into a unified test procedure. For each functional electrode, we define the electrode with the adjacent functional electrode as a merging-and-splitting (MAS) cluster. Figure 4.14 presents the test procedure within a MAS cluster. Two 1X droplets are required to complete the MAS test within a cluster.

If a MAS cluster is not located on the boundary of a MEDA biochip, there are eight steps associated with the MAS cluster, i.e., horizontal merging, horizontal splitting, vertical merging, vertical splitting, left-diagonal merging, left-diagonal splitting, right-diagonal merging, and right-diagonal splitting, and the MAS cluster occupies nine functional electrodes. An example is shown for the MAS cluster cor-

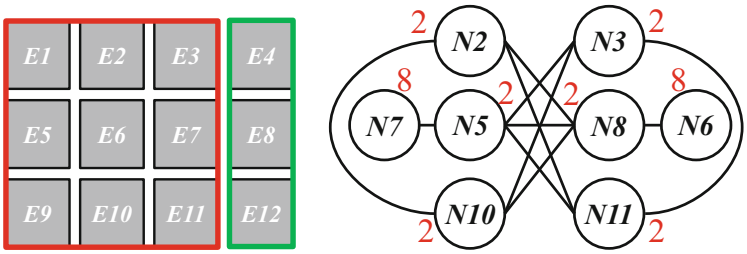



**Fig. 4.15** (**a**) Illustration of MAS clusters; red outline and green outline highlight the MAS cluster corresponding to functional electrodes $E6$ and $E8$, respectively. (**b**) Mapping of the functional electrodes to an undirected graph

responding to functional electrode $E6$ in Fig. 4.15a. Otherwise, there are only two steps (either horizontal merging-and-splitting or vertical merging-and-splitting), and the MAS cluster occupies three functional electrodes; an example of the MAS cluster corresponding to functional electrode $E8$ is shown in Fig. 4.15a. Note that there is no MAS cluster for the functional electrode at a corner. This is because neither merging nor splitting operation can be performed there.

As shown in Fig. 4.14, eight manipulation steps are required to test one functional electrode. If functional electrodes are sequentially tested, we need a large number of manipulation steps ($\sim 8MN$) for an $M \times N$ functional electrode array. In order to achieve higher test efficiency, we utilize multiple test droplets and apply a parallel test method for array testing. The key idea is to concurrently test functional electrodes whose MAS clusters are disjoint.

We refer to two functional electrodes as *compatible functional electrodes* if they can be tested concurrently. For example, $E6$ and $E8$ in Fig. 4.15a are compatible functional electrodes, i.e., merging-and-splitting tests for $E6$ and $E8$ can be concurrently executed. Therefore, we are able to reduce the number of manipulation steps by finding mutually-compatible functional electrodes. We first partition the set of functional electrodes into groups. Functional electrodes in the same group must be pairwise-compatible. Manipulation steps are also associated with each functional electrode. Our goal is to find an optimal partitioning that leads to the minimum number of manipulation steps in total.

This problem can be mapped to the problem of clique partitioning from graph theory [30]. An example of this mapping is illustrated in Fig. 4.15. Based on the compatibility between functional electrodes, an undirected graph, referred to as the *MAS-compatibility graph*, is constructed; see Fig. 4.15b. Each node in the graph represents a configured functional electrode, and each node is associated with a weight indicating the number of manipulation steps to complete the MAS testing for the corresponding functional electrode; see red numbers in Fig. 4.15b. An edge between two nodes indicates that these two functional electrodes are compatible with each other. After the construction of the functional electrode graph, we are able to divide the graph into cliques. A *clique* is defined as a complete subgraph,

1: **Start**: input a MAS-compatibility graph $G_{MAS}$ and the maximum number of concurrent merging-and-splitting tests $N_{max}$;
2: Apply Bron-Kerbosch algorithm to obtain the set $S_c$ containing all cliques in $G_{MAS}$;
3: Eliminate cliques with the clique size larger than $N_{max}$ in $S_c$;
4: **for** each clique $C_i$ in $S_c$ **do**
5: Initialize #manipulation_steps for sequential testing: $N_{is} = 0$;
6: Initialize #manipulation_steps for parallel testing: $N_{ip} = 0$;
7: **for** each vertex $V_j$ in $C_i$ **do**
8: $N_{is} = N_{is} + weight(V_j)$;
9: **end for**
10: $N_{ip} = \max_j \{weight(V_j)\}$;
11: Calculate the benefit for clique $C_i$: $B_{C_i} = N_{is} - N_{ip}$;
12: **end for**
13: Select the clique $C_{max}$ with the maximum benefit value;
14: Remove $C_{max}$ from $G_{MAS}$ and update the graph;
15: Repeat above steps until $G_{MAS}$ becomes an empty graph;
16: **End**: output #manipulation_steps based on the clique partition results.

**Fig. 4.16** Pseudocode for the clique partitioning procedure

i.e., any two nodes in this subgraph are connected by an edge [31]. The size of each partitioned clique should also be constrained by the number of available test droplets. The number of test manipulation steps for each clique is determined by the maximum vertex weight within the clique, and the total number of manipulation steps for the complete chip is the summation of the steps for each clique.

After the partitioning of the functional electrode graph, all functional electrodes in each group can be tested concurrently. Therefore, the number of final manipulation steps can be significantly reduced. For example, if all functional electrodes in Fig. 4.15a are sequentially tested, the total number of manipulation steps is 28. If functional electrodes in groups $\{E2, E8, E10\}$ and $\{E3, E5, E11\}$ are concurrently tested, the total number of manipulation steps for the parallel testing reduces to 20.

We propose an algorithm to solve the clique-partition problem in an efficient manner. The pseudocode is presented in Fig. 4.16. After the construction of the MAS-compatibility graph $G_{MAS}$, we first apply the Bron-Kerbosch algorithm [32] to find all cliques in $G_{MAS}$ and eliminate cliques of size larger than the maximum number of concurrent MAS tests, $N_{max}$ (lines 2–3). Note that $N_{max}$ is determined by the number of available test droplets. If there are $N$ test droplets on the chip, $N_{max}$ is $\lfloor N/2 \rfloor$ (two 1X droplets are needed in each MAS cluster). For each clique $C_i$, we calculate the number of manipulation steps for both sequential testing and parallel testing, i.e., $N_{is}$ and $N_{ip}$ (lines 5–10). The benefit for clique $C_i$ is calculated as the difference between $N_{is}$ and $N_{ip}$ (line 11). We then select the clique with the maximum benefit value and remove the selected clique from $G_{MAS}$ (lines 13–14). We repeat above steps until $G_{MAS}$ becomes an empty graph. Based on the clique partition results, we are able to calculate the number of total number of manipulation steps for the parallel testing.

## 4.6 Experimental Results

In this section, we present SPICE simulation results and experimental results for the structural test. We also evaluate the effectiveness of the proposed test procedures for functional test. We finally provide a system-dependability analysis for both the structural and functional test methods.

### *4.6.1 Results for Structural Test*

We evaluated the MC design of Fig. 4.6 using HSPICE and a 350 nm library from a foundry. We first present simulation results for the macro-model proposed in Sect. 4.5.3, and compare simulation results with experimental results. We then present simulation and experimental results for the droplet-actuation, droplet-maintenance, and droplet-sensing test.

A comparison between SPICE simulation results and experimental results related to EDMOS for two fabricated MEDA biochips is shown in Fig. 4.17. There is a good correlation between the simulation and the experimental results, which demonstrates the accuracy of the proposed SPICE model.

Simulation results for the droplet-actuation test for a single MC are shown in Fig. 4.18. We first set the MC to the functional mode and apply the test pattern (logic 1) to the MC. We then switch the MC to the test mode and capture the voltage on the node $N4$ in Fig. 4.6. For the fault-free MC, the voltage on the microelectrode is 0 V during actuation test. Therefore, the output is logic 0; see fault-free signals V($N4$) and $OUT$ in Fig. 4.18. We then injected a stuck-at-0 fault on $N3$. The EDMOS

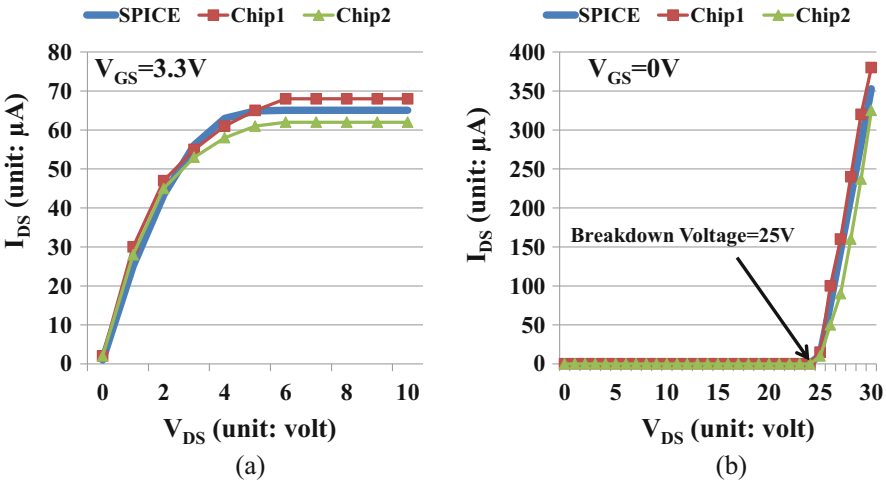



**Fig. 4.17** Comparison between the simulation results and the experimental results when the EDMOS is (**a**) forward-biased and (**b**) reverse-biased

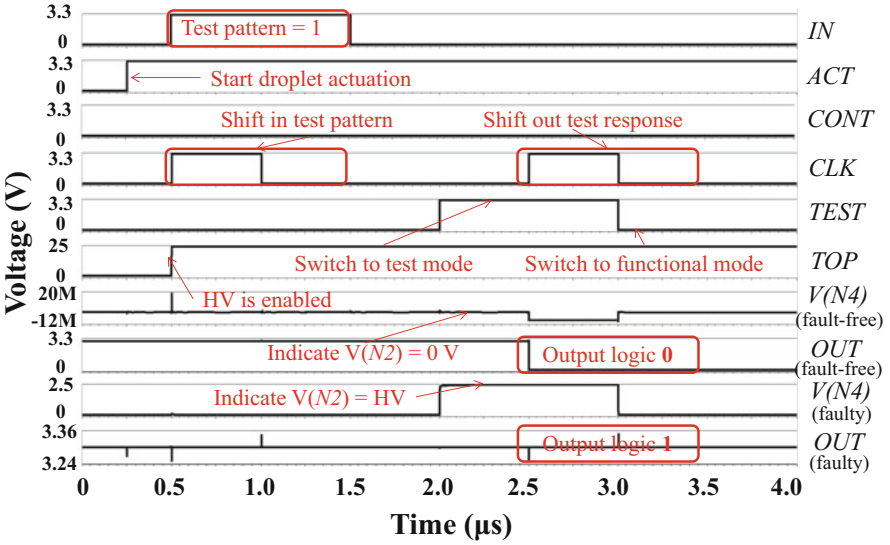



**Fig. 4.18** Simulation results for the droplet-actuation test in an MC

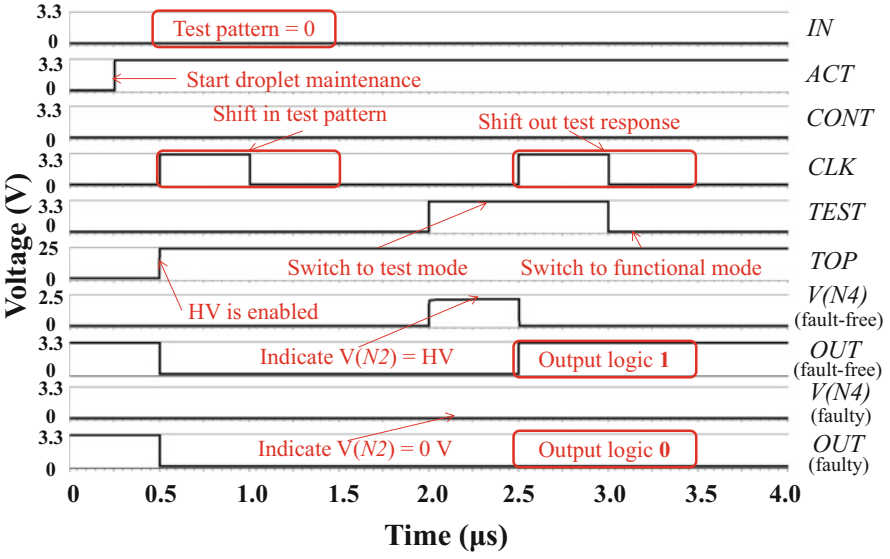



**Fig. 4.19** Simulation results for the droplet-maintenance test in an MC

cannot be turned on now during the droplet-actuation test, and the output is logic 1; see faulty signals V($N4$) and $OUT$ in Fig. 4.18.

Simulation results for the droplet-maintenance test for a single MC are shown in Fig. 4.19. Here we applied the test pattern, logic 0, to the MC. Similar to the droplet-actuation test, we first set the MC to functional mode and then switch to test mode.

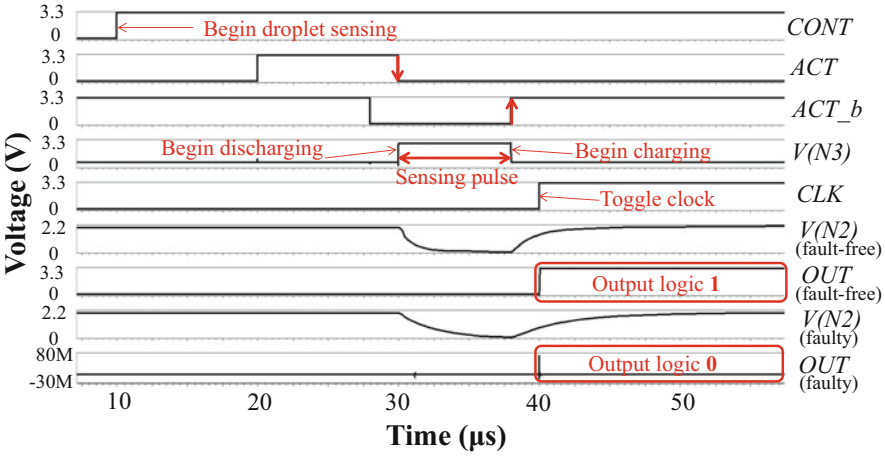



**Fig. 4.20** Simulation results for the droplet-sensing test in an MC

If the MC is fault-free, the voltage on the microelectrode is high. Therefore, the output from the MC is logic 1; see fault-free signals V($N4$) and $OUT$ in Fig. 4.19. We then injected a stuck-at-0 on the node N2. Therefore, the microelectrode cannot be driven to HV, and the output changes to logic 0; see faulty signals V($N4$) and $OUT$ in Fig. 4.19.

Simulation results for the droplet-sensing test are shown in Fig. 4.20. We first set $CONT$ to logic 1 to drive the MC to the sensing mode. The voltage on the top electrode was switched to the ground level. We then used signals $ACT$ and $ACT_b$ to generate a sensing pulse. At the rising edge of the sensing pulse, $C_{drop}$ was discharged through the EDMOS and transistor M3; at the falling edge of the sensing pulse, $C_{drop}$ was charged through transistors M1, M2 and the EDMOS. Different values of the capacitance $C_{drop}$ result in different charging and discharging times. As shown in Fig. 4.20, the output is a logic 1 for the fault-free MC. We then injected a fault on $C_{drop}$ by increasing its capacitance by 15 fF. Corresponding simulation results are also shown in Fig. 4.20. For the faulty MC, the MC sensing output is logic 0. The output of each MC can be utilized to generate a global sensing map. If an MC outputs logic 1, we use a green dot to represent that MC in the sensing map; otherwise, we use a black dot to represent that MC. Examples are shown in Fig. 4.22.

We also applied the droplet-sensing test to two fabricated MEDA biochips. The chip micro-photo and setup for the demonstration are shown in Fig. 4.21. The chip has an area of 7.45 $mm^2$ and it was fabricated using a 0.35 µm standard CMOS process.

In order to measure the effectiveness of the droplet-sensing test, we intentionally damaged a group of microelectrodes using extremely high voltage (electrode breakdown) on one MEDA biochip; see Fig. 4.22a. We also intentionally etched part of the Teflon layer using alcohol solution on the other MEDA biochip; see Fig. 4.22b.

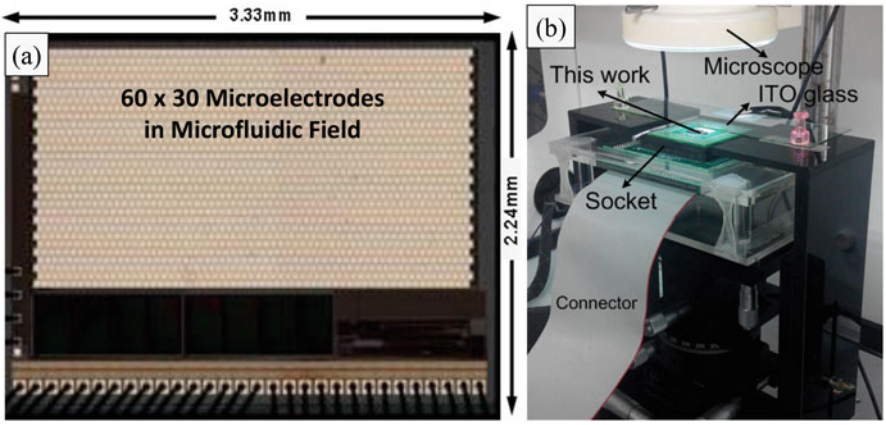



**Fig. 4.21** Chip micro-photo and experimental setup

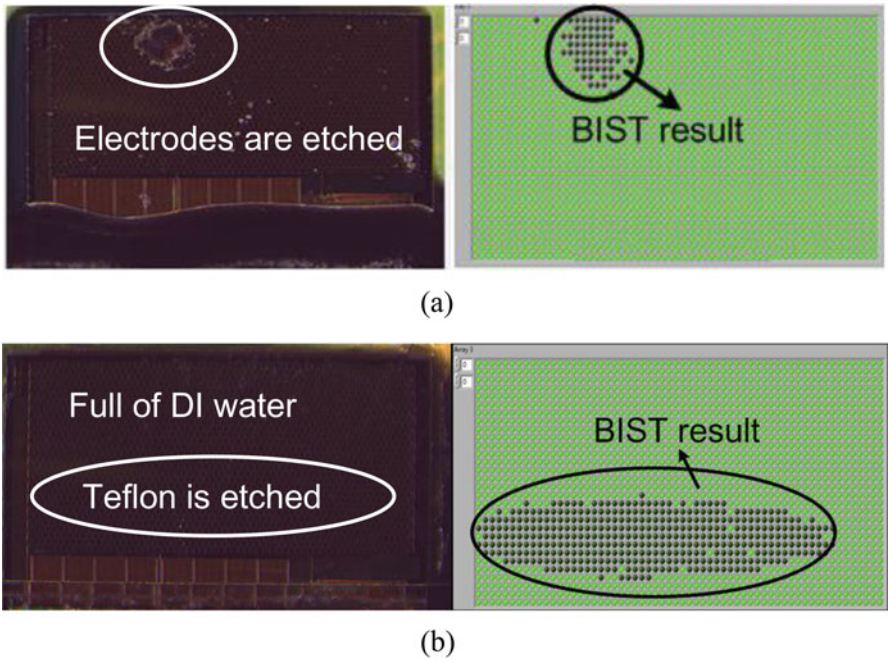



**Fig. 4.22** MEDA biochips with: (**a**) etched microelectrodes; (**b**) etched Teflon layer and corresponding sensing maps

We then applied droplet-sensing tests on these two chips. The experimental results for the droplet-sensing test are also shown in Fig. 4.22, which demonstrates that we are able to detect faulty MCs based on the generated sensing map.

**Table 4.1** Results on droplet-routing steps for two MEDA biochips

| Chip size | FE size | FR = 0.0 | FR = 0.1 | FR = 0.2 |
|---|---|---|---|---|
| 60×30 [4] | 3×3 | 1424 | 1158 | 896 |
| | 5×5 | 472 | 378 | 338 |
| | 6×6 | 314 | 264 | 200 |
| | 10×10 | 94 | 86 | 66 |
| 30×30 [20] | 3×3 | 684 | 536 | 470 |
| | 5×5 | 220 | 188 | 134 |
| | 6×6 | 144 | 118 | 104 |
| | 10×10 | 40 | 34 | 26 |

### *4.6.2 Results for Functional Test*

In this subsection, we use the proposed functional test methods to (1) generate droplet pathways for routing test, and (2) generate the test plan for mixing and splitting tests. All simulations are carried out on an Intel Core i7 platform with a 2.67 GHz CPU and 8 GB of RAM.

We first present the results for routing test. We define the movement of one droplet from its original location to the adjacent functional electrode as one *routing step*. In order to explore the relationship between the number of routing steps and the size of the functional electrode, we generate droplet routing paths for MEDA biochips with different sizes of functional electrodes. Simulation results are presented in Table 4.1. In this table, chip size is described using the microelectrode array; $FE$ size represents the size of the functional electrode. The parameter $FR$ represents the fail ratio, i.e., the ratio of the number of functional electrodes that fail structural test over the total number of functional electrodes on the chip.

As shown in Table 4.1, with an increase in the FE size, the number of routing steps decreases significantly. For example, if FR is zero (no functional electrodes fail structural test) and the FE size is $3 \times 3$ for the chip with the size of $60 \times 30$ microelectrode array, the total number of routing steps is 1424. However, if the FE size changes to $10 \times 10$, the corresponding number of routing steps reduces to 94, which is 93% less than the number for the case of a $3 \times 3$ functional electrode. Moreover, with an increase in FR, the number of routing steps also decreases. This clearly illustrates that the structural test can help to alleviate the difficulty and complexity associated with functional testing.

We next present the results for the mixing and splitting test. We compare the number of manipulation steps between parallel and sequential testing. Note that the number of manipulation steps is directly related to the testing time. Therefore, a smaller number of manipulation steps leads to shorter testing time. Simulation results are presented in Table 4.2. In Table 4.2, $PT$ and $ST$ represent the parallel testing and the sequential testing, respectively. For parallel testing, we also set the maximum number of concurrent MAS tests (denoted by $N$ in the table) to different values, i.e., from one to four.

**Table 4.2** Results on the number of manipulation steps for two MEDA biochips

| Chip size | FE size | PT | | | | ST |
|---|---|---|---|---|---|---|
| | | N = 1 | N = 2 | N = 3 | N = 4 | |
| 60×30 [4] | 3×3 | 1256 | 720 | 506 | 394 | 1256 |
| | 5×5 | 376 | 244 | 176 | 146 | 376 |
| | 6×6 | 236 | 164 | 126 | 100 | 236 |
| | 10×10 | 52 | 34 | 32 | 30 | 52 |
| 30×30 [20] | 3×3 | 576 | 338 | 240 | 190 | 576 |
| | 5×5 | 160 | 114 | 92 | 80 | 160 |
| | 6×6 | 96 | 78 | 74 | 74 | 96 |
| | 10×10 | 16 | 12 | 12 | 12 | 16 |

As shown in Table 4.2, when $N$ is larger than one, parallel testing always outperforms sequential testing. When $N$ is one, there is no difference between the two test methods. Table 4.2 also illustrates that an increase in $N$ does not always reduce the number of manipulation steps. For example, when the FE size is $10 \times 10$ for a $30 \times 30$ MEDA biochip, the number of manipulation steps is always 12 when $N$ is larger than one; see the last line in Table 4.2. This can be explained as follows. When the FE size is large and the chip size is small, the constructed MAS-compatibility graph is also small, which results in a maximum clique of small size in the graph. When $N$ is larger than the size of the maximal clique, an increase in $N$ cannot influence the clique partitioning results. Accordingly, the number of manipulation steps is also not influenced.

In this subsection, we evaluate system dependability of the synthesized design on a MEDA biochip. The protein dilution bioassay [33] is selected as the design to be synthesized. A $60 \times 30$ MEDA biochip [4] is used to execute the assay and the functional electrode size is set as a $5 \times 5$ microelectrode array. The microfluidic library is the same as the library used in [1].

We evaluate system dependability for three cases: (1) no testing is carried out, (2) only structural test is carried out, and (3) both structural and functional test are carried out. We generate 200 simulated samples of faulty arrays. Each faulty array is derived by randomly injected a number of defects. The number of injected defects is determined by the defect probability $p$, where $p$ is defined as the probability that a unit cell (functional electrode) has a defect. Here we let the probability $p$ take two values, 0.01 and 0.05. We define malfunction probability $q$ as the probability that a module malfunctions even if there is no defect within it. We consider three values of $q$: 0.01, 0.02, and 0.03. Since some malfunctions are latent [34], we define a third parameter, error probability $r$, as the probability that a module malfunctions during operation execution even if the module passes functional test. We consider two values, 0.00 and 0.01, for the error probability $r$.

We then apply the synthesis method from [1] to map the protein dilution assay on 200 synthetic arrays. Here we assume that all defects can be detected by the

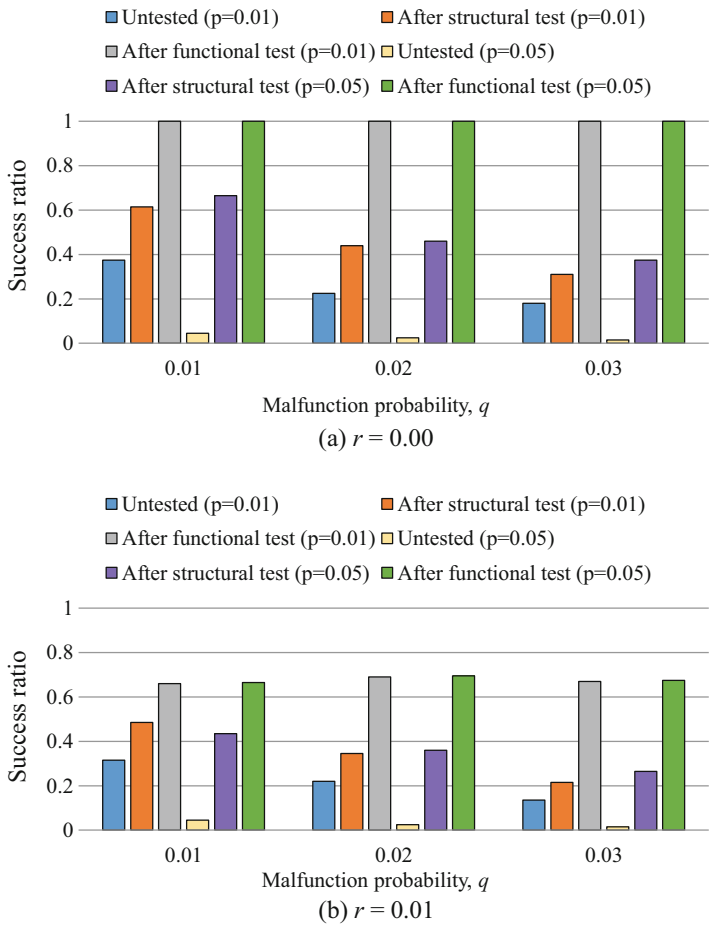



**Fig. 4.23** Success rate for synthesized design without any testing, after the structural test, and after the functional test with different error probabilities

proposed structural test and all malfunctions can be detected by the proposed functional test. If structural test is carried out, the assay is mapped to defect-free regions. If both the structural test and functional test are carried out, all injected defects and malfunctions are bypassed during the chip synthesis; only the latent malfunctions can influence the synthesis results.

We finally determine the success ratio $SR$ for the three scenarios. The metric $SR$ is calculated as the ratio of the number of successful synthesis attempts over the total number of experiments. Note that a synthesis is deemed to fail if any module suffers from a defect or a malfunction. The simulation results are shown in Fig. 4.23. If no testing is carried out, the success ratio is very low. If structural testing is carried out, the success rate is higher, but it is still lower than the success

ratio after functional test. Moreover, as the malfunction probability $q$ increases, the $S_R$ decreases considerably even when structural testing is applied, which clearly illustrates the importance of the functional testing.

In Fig. 4.23a, the probability $r$ is 0.00, which indicates that there is no latent malfunction during the assay execution. Therefore, the success rate $S_R$ is always 1.0 after the functional test. In Fig. 4.23a, the probability $r$ is 0.01. The $S_R$ decrease from 1.0 to around 0.65. However, the success rate is still obviously higher than the rates for the other two scenarios.

## 4.7 Conclusion

This chapter presents a comprehensive test techniques that allows both structural and functional testing for MEDA biochips. We have proposed an HSPICE macro-model for the simulation of the EDMOS and a test structure design for the structural test within each MC. We have also presented test procedures for the functional testing of MEDA biochips. These procedures address fundamental microfluidic operations, such as droplet dispensing, transportation, droplet mixing, and splitting. Simulation and experimental results have been presented to highlight the need of functional testing and demonstrate the effectiveness of the proposed test techniques.

## References

1. Li, Z., Lai, K. Y.-T., Yu, P.-H., Ho, T.-Y., Chakrabarty, K., & Lee, C.-Y. (2016) High-level synthesis for micro-electrode-dot-array digital microfluidic biochips. In *Proceedings of ACM/IEEE design automation conference (DAC)* (pp. 146:1–146:6). New York: ACM.
2. Hu, K., Ibrahim, M., Chen, L., Li, Z., Chakrabarty, K., & Fair, R. (2015). Experimental demonstration of error recovery in an integrated cyberphysical digital-microfluidic platform. In *IEEE biomedical circuits and systems conference (BioCAS)* (pp. 1–4).
3. Li, Z., Lai, K. Y.-T., Yu, P.-H., Chakrabarty, K., Pajic, M., Ho, T.-Y., & Lee, C.-Y. (2016). Error recovery in a micro-electrode-dot-array digital microfluidic biochip. In *Proceedings of IEEE/ACM international conference on computer-aided design (ICCAD)* (pp. 105:1–105:8).
4. Lai, K. Y.-T., Shiu, M.-F., Lu, Y.-W., Ho, Y.-C., Kao, Y.-C., Yang, Y.-T., Wang, G., Liu, K.-M., Chang, H.-C., & Lee, C.-Y. (2015). A field-programmable lab-on-a-chip with built-in self-test circuit and low-power sensor-fusion solution in 0.35 μm standard cmos process. In *Proceedings of IEEE Asian solid-state circuits conference (A-SSCC)* (pp. 1–4).
5. Li, Z., Lai, K. Y.-T., Yu, P.-H., Chakrabarty, K., Ho, T.-Y., & Lee, C.-Y. (2017). Droplet size-aware high-level synthesis for micro-electrode-dot-array digital microfluidic biochips. *IEEE Transactions on Biomedical Circuits and Systems (TBioCAS), 11*(3), 612–626.
6. Li, Z., Dinh, T. A., Ho, T.-Y., & Chakrabarty, K. (2014). Reliability-driven pipelined scan-like testing of digital microfluidic biochips. In *Proceedings of IEEE Asian test symposium (ATS)* (pp. 57–62).
7. Xu, T., & Chakrabarty, K. (2007). Functional testing of digital microfluidic biochips. In *Proceedings of IEEE international test conference (ITC)* (pp. 1–10).
8. Zhao, Y., Xu, T., & Chakrabarty, K. (2008). Built-in self-test and fault diagnosis for lab-on-chip using digital microfluidic logic gates. In *Proceedings of IEEE international test conference (ITC)* (pp. 1–10).

9. Shukla, V., Ali, N. B. Z., Hussin, F. A., & Zwolinski, M. (2013). On testing of MEDA based digital microfluidics biochips. In *Proceedings of IEEE Asian symposium of quality electronic design (ASQED)* (pp. 60–65).
10. Shukla, V., Ali, N. B. B. Z., Hussin, F.A., Hamid, N. H., & Sheikh, M. A. (2016). Fault modeling and simulation of MEDA-based digital microfluidics biochips. In *Proceedings of IEEE international conference on VLSI design (VLSID)* (pp. 469–474).
11. Li, Z., Lai, K. Y.-T., Yu, P.-H., Chakrabarty, K., Ho, T.-Y., & Lee, C.-Y. (2016). Built-in self-test for micro-electrode-dot-array digital microfluidic biochips. In *Proceedings of IEEE international test conference (ITC)* (pp. 1–10).
12. Xu, T., & Chakrabarty, K. (2009). Fault modeling and functional test methods for digital microfluidic biochips. *IEEE Transactions on Biomedical Circuits and Systems (TBioCAS), 3*, 241–253.
13. Berry, S., Kedzierski, J., & Abedian, B. (2007). Irreversible electrowetting on thin fluoropolymer films. *Langmuir, 23*, 12429–12435.
14. Su, F., Ozev, S., & Chakrabarty, K. (2003). Testing of droplet-based microelectrofluidic systems. In *Proceedings of IEEE international test conference (ITC)* (pp. 1192–1200).
15. Su, F., & Chakrabarty, K. (2005). Ensuring the operational health of droplet-based microelectrofluidic biosensor systems. *IEEE Sensors Journal, 5*, 763–773.
16. Su, F., Ozev, S., & Chakrabarty, K. (2006) Test planning and test resource optimization for droplet-based microfluidic systems. *Journal of Electronic Testing Theory and Application (JETTA), 22*(2), 199–210.
17. Su, F., Hwang, W., Mukherjee, A., & Chakrabarty, K. (2007). Testing and diagnosis of realistic defects in digital microfluidic biochips. *Journal of Electronic Testing Theory and Application (JETTA), 23*(2–3), 219–233.
18. Xu, T., & Chakrabarty, K. (2007). Parallel scan-like test and multiple-defect diagnosis for digital microfluidic biochips. *IEEE Transactions on Biomedical Circuits and Systems (TBioCAS), 1*, 148–158.
19. Mitra, D., Ghoshal, S., Rahaman, H., Chakrabarty, K., & Bhattacharya, B. B. (2011). Test planning in digital microfluidic biochips using efficient eulerization techniques. *Journal of Electronic Testing Theory and Application (JETTA), 27*, 657–671.
20. Lai, K. Y.-T., Yang, Y.-T., & Lee, C.-Y. (2015). An intelligent digital microfluidic processor for biomedical detection. *Journal of Signal Processing Systems, 78*(1), 85–93.
21. Guo, R., Huang, Y., & Cheng, W.-T. (2015). Fault dictionary based scan chain failure diagnosis. August 18 2015, uS Patent 9,110,138.
22. Huang, Y., Guo, R., Cheng, W.-T., & Li, J. C.-M. (2008). Survey of scan chain diagnosis. *IEEE Design & Test of Computers, 25*, 240–248.
23. Bushnell, M. M., & Agrawal, V. D. (2000). *Essentials of electronic testing for digital, memory and mixed-signal VLSI circuits*. Berlin: Springer.
24. Parpia, Z., Salama, C., & Hadaway, R. (1987). Modeling and characterization of CMOS-compatible high-voltage device structures. *IEEE Transactions on Electron Devices (TED), 34*, 2335–2343.
25. Cheng, Y., Jeng, M.-C., Liu, Z., Huang, J., Chan, M., Chen, K., Ko, P. K., & Hu, C. (1997). A physical and scalable IV model in BSIM3v3 for analog/digital circuit simulation. *IEEE Transactions on Electron Devices (TED), 44*, 277–287.
26. Li, Y.-M., & Connelly, J. A. (2003). Macromodeling with SPICE for the voltage breakdown behavior in bipolar junction and field-effect transistors. In *Proceedings of IEEE southwest symposium on mixed-signal design* (pp. 166–169).
27. Hodges, D. A., & Jackson, H. G. (2005). *Analysis and design of digital integrated circuits*. New York: Tata McGraw-Hill Education.
28. Taur, Y., & Ning, T. H. (2013). *Fundamentals of modern VLSI devices*. Cambridge: Cambridge University Press.
29. Wang, G., Teng, D., & Fan, S.-K. (2011). Digital microfluidic operations on micro-electrode dot array architecture. *IET Nanobiotechnology, 5*(4), 152–160.

30. Bhasker, J., & Samad, T. (1991). The clique-partitioning problem. *Computers & Mathematics with Applications, 22*, 1–11.
31. Gross, J. L., & Yellen, J. (2005). *Graph theory and its applications*. Boca Raton: CRC Press.
32. Johnston, H. (1976). Cliques of a graph-variations on the bron-kerbosch algorithm. *International Journal of Parallel Programming, 5*, 209–238.
33. Su, F., & Chakrabarty, K. (2005). Unified high-level synthesis and module placement for defect-tolerant microfluidic biochips. In *Proceedings of ACM/IEEE design automation conference (DAC)* (pp. 825–830).
34. Luo, Y., Chakrabarty, K., & Ho, T.-Y. (2013). Error recovery in cyberphysical digital microfluidic biochips. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), 32*(1), 59–72.

# Chapter 5
# Generalized Error-Correcting Sample Preparation


## 5.1 Chapter Overview

A potentially important application of MEDA biochips lies in *sample preparation* via a series of dilution steps. The dilution of fluids plays a fundamental role in biomolecular protocols [1]. Sample preparation is used to produce droplets with desired concentrations from stock solutions of reagents and samples. During sample preparation, reagent and sample droplets are repeatedly mixed with buffer droplets until the target concentration is obtained. According to [2], sample preparation accounts for 90% of the cost and 95% of the analysis time in biochemical experiments. Therefore, it is critical to provide automation for sample preparation on MEDA biochips. The main objective in sample preparation is to minimize: (1) the volume of sample needed (i.e., the number of sample droplets); (2) the amount of waste (i.e., the number of generated waste droplets), which corresponds to the waste-handling overhead; (3) the number of dilution operations, which determines the sample-preparation time.

In this chapter, we present a droplet size-aware and error-correcting sample-preparation method for MEDA biochips. In contrast to previous methods, the proposed approach considers droplet sizes and incorporates various mixing models in sample preparation. It utilizes MEDA-enabled microfluidic operations, and fully exploits the feature of real-time droplet sensing on MEDA biochips.

The remainder of the chapter is organized as follows. Section 5.2 reviews previous works on sample preparation and introduces MEDA-enabled operations and corresponding experimental demonstrations. Section 5.3 describes sample preparation on MEDA biochips and presents the problem formulation. Section 5.4 describes the proposed sample-preparation approach in more detail. Section 5.5 presents the proposed error-recovery approach for sample preparation. Section 5.6 presents simulation results on test cases as well as real-life examples. Finally, Sect. 5.7 concludes the chapter.



Z. Li et al., *Micro-Electrode-Dot-Array Digital Microfluidic Biochips*,
https://doi.org/10.1007/978-3-030-02964-7_5

## 5.2 Prior Work and MEDA-Enabled Operations

Several sample-preparation algorithms have been proposed for conventional DMFBs [3–9] and for flow-based microfluidic biochips [10, 11]. The bit scanning (BS) method guarantees a minimum number of dilution operations [3]. The algorithm for dilution and mixing with reduced wastage (DMRW) [4] and the improved dilution/mixing algorithm (IDMA) [5] adopt a graph-based dilution strategy with droplet sharing. Both DMRW and IDMA minimize the amount of waste droplets. The reactant minimization algorithm (REMIA) [7] minimizes reactant usage. The graph-based optimal reactant minimization algorithm (GORMA) [8] utilizes a systematic method to exhaustively examine all feasible dilution combinations to produce an optimal dilution graph. The work in [6] presents an optimal sample preparation algorithm based on a minimum-cost maximum-flow model. Since droplet size is usually fixed to be one unit for mixing operations on conventional DMFBs, all of these algorithms are limited to the (1:1) mixing model. In the (1:1) mixing model, equal volumes of a sample/reagent and buffer must be used in each mixing step. The algorithm in [12] extends the (1:1) mixing model to (1:2) and (2:1) mixing models. However, specially designed mixers are required for this extension, which inhibits the usage of the proposed algorithm for general DMFBs. Different with conventional DMFBs, droplets of different sizes can be manipulated on MEDA biochips. Therefore, a general ($m$:$n$), where neither $m$ nor $n$ need to be 1, mixing model can be used. The opportunity to accomplish more efficient on-chip sample preparation using this generalized mixing model has not been exploited thus far.

Because of the inherent uncertainty associated with fluidic operations, the outcome of microfluidic operations can be erroneous even if the chip is deemed to be defect-free [13, 14]. It has been reported that 80% of errors in sample preparation can be attributed to volume variation in droplets [15]. Droplet-volume variation may cause errors in the concentration of target droplets. Therefore, error recovery is also desired for sample preparation. Several error-recovery techniques have been proposed for conventional DMFBs [9, 13]. These techniques use inserted checkpoints for verifying the correctness of fluidic operations, but they cannot fully utilize the advantages specific for MEDA biochips, e.g., MEDA-enabled fluidic operations and real-time droplet sensing.

MEDA allows the dynamic grouping of microelectrodes to form different fluidic modules, and MEDA can provide users with detailed information about the outcomes of on-chip operations. Because of the programmability and the ability for droplet sensing, some advanced microfluidic operations can be achieved only on MEDA biochips. In order to fully utilize the MEDA benefits, here we introduce two MEDA-enabled operations, namely (1) *pseudo-dispensing* and (2) *droplet-regulation*, that can be used for sample preparation.

As shown in [16], a large droplet $D_i$ on a MEDA biochip can work as a pseudo reservoir, and small droplets can be dispensed from $D_i$. We refer to the generation of a smaller droplet from a larger droplet as a pseudo-dispensing operation. The chip used for experiments has an area of 7.4 $mm^2$ (3.33$\times$ 2.24 $\mu$m) and it was

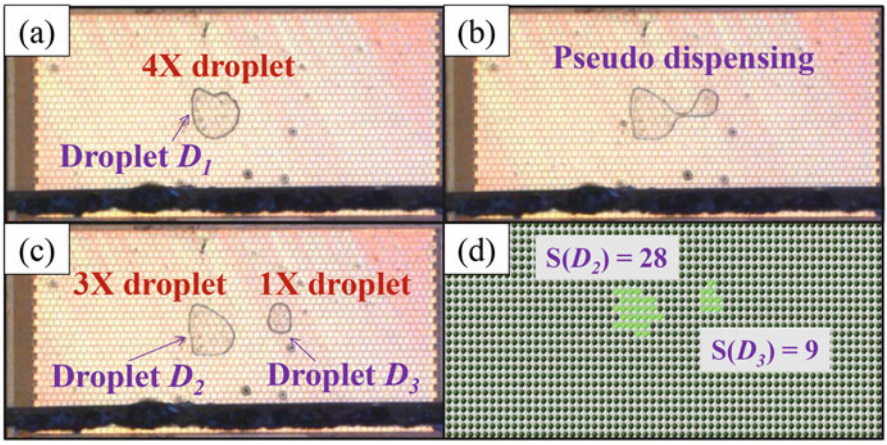



**Fig. 5.1** Experimental demonstration of a pseudo- dispensing operation. (**a**) A 4X droplet $D_1$. (**b**) Snapshot of pseudo-dispensing from $D_1$. (**c**) Droplet $D_3$ is dispensed from $D_1$, and the droplet $D_2$ is left at the original location. (**d**) Droplet-sensing results for $D_2$ and $D_3$ ($S(D_i)$ represents the droplet size for $D_i$)

fabricated at TSMC using a 0.35 μm CMOS process. The microelectrode pitch is 50 μm. In the fabricated MEDA biochip, the minimum and the maximum droplet volumes that can be manipulated are 2 × 2 and 30 × 30, respectively. The pseudo-dispensing operation was performed using a 6×6 deionized (DI) water droplet. Fig. 5.1 illustrates an experimental demonstration that we have carried out in the laboratory. In Fig. 5.1a–c, a 1X droplet ($D_2$) is dispensed from a 4X droplet ($D_1$). Figure 5.1d presents the droplet-size sensing results.

The droplet-regulation operation can be used to precisely control droplet size. For example, if the size of $D_1$ is larger than the size of $D_2$, a small "regulating" droplet $D_c$ can be dispensed from $D_1$ and merged with $D_2$ to balance the size of $D_1$ and $D_2$. We refer to this operation as a droplet regulation. It can be used to correct volumetric errors during sample preparation. Experimental demonstration of the operation is shown in Fig. 5.2. Two droplets in Fig. 5.2a are split from a 7×7 DI water droplet. As shown in Fig. 5.2a–d, after $D_c$ is merged with $D_2$, the new droplet $D_4$ occupies 24 microelectrodes and $D_3$ occupies 25 microelectrodes. Therefore, the original droplet-size difference between $D_1$ and $D_2$ is almost completely eliminated.

## 5.3 Sample Preparation on MEDA

During sample preparation, sample and buffer droplets initially have 100% and 0% concentrations, respectively. Mixing and splitting are two fundamental operations that are frequently used to dilute a sample. For conventional DMFBs, only a (1:1)

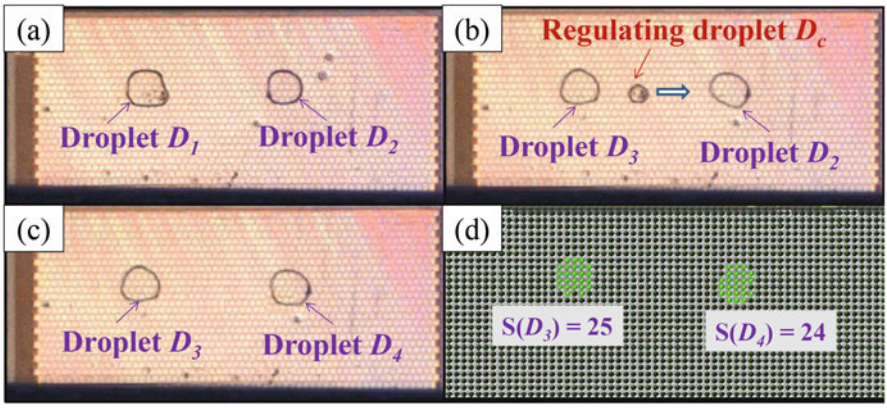



**Fig. 5.2** Experimental demonstration of droplet regulation. (**a**) Two droplets $D_1$ and $D_2$. (**b**) Compensating droplet $D_c$ is dispensed from $D_1$, and is going to be merged with $D_2$. (**c**) Compensating droplet $D_c$ is merged with $D_2$ to generated droplet $D_4$. (**d**) Droplet-sensing results ($S(D_i)$ represents the size of $D_i$)

mixing/splitting ratio can be used. In contrast, MEDA chips provide more precise control on droplet size. Different sizes of droplets can be manipulated on the chip. Therefore, a general ($m$:$n$) mixing/splitting ratio can be achieved.

When the target concentration $C_t$ and the degree of accuracy $\delta$ are given, any concentration $C_i$ can be expressed as $N_i/\lceil 1/\delta \rceil$ ($N_i$ is an integer) and the target concentration can be expressed as $||c_t/\delta||/\lceil 1/\delta \rceil$. Note that the symbol "|| ||" means that the value needs to be rounded to the nearest integer. For the sake of simplicity, we present a concentration value only by its numerator ($N_i$) instead of the fraction ($N_i/\lceil 1/\delta \rceil$) for a given number of $\delta$. For example, if $\delta$ is 1/64, a concentration of 56% is expressed as 36/64, and we only use the number 36 to represent the concentration.

We first define a pair of a mixing operation and a pseudo-dispensing operation as a "MEDA-enable dilution" operation. In a MEDA-enable dilution operation, a droplet of size $S_i$ and concentration of $C_i$ is diluted with another droplet of size $S_j$ and concentration $C_j$ to produce a diluted droplet of concentration $C_{mix}$ and size $S_{mix}$. Note that $C_{mix}$ and $S_{mix}$ satisfy the constraints (5.1) and (5.2) listed below. Note that we constrain the size of a generated droplet after a MEDA-enable dilution operation to be an integer; see (5.2). This constraint ensures that droplet sizes are always integer multiples of the minimum volume possible in a MEDA biochip.

$$C_{mix} = ||\big(S_i C_i + S_j C_j\big)/\big(S_i + S_j\big)|| \tag{5.1}$$

$$S_{mix} \leq S_i + S_j, \quad \text{where } S_{mix} \in \mathbb{N} \tag{5.2}$$

### 5.3.1 *Dilution Graph for MEDA*

A directed graph, referred to as the *dilution graph*, is used to illustrate sample preparation [4, 7]. It presents a dilution strategy that can be used to generate the target droplet. Each vertex is labeled by a concentration value (denoted only by its numerator) and the droplet size, denoted by $c_{i\,(s_i)}$, where $c_i$ is the concentration and $s_i$ is the droplet size.

Figure 5.3 presents dilution graphs generated using different methods. The parameters $N_s$, $N_w$, and $N_o$ in Fig. 5.3 represent the number of sample droplets, waste droplets, and dilution operations, respectively. The target droplet is a 1X droplet with a concentration of $\frac{1}{3}$, and the degree of concentration accuracy $\delta$ is $\frac{1}{64}$. Therefore, the target droplet can be expressed as 21(1) in Fig. 5.3. Note that the BS and REMIA methods can only use 1X droplets while the sample-preparation method for MEDA can utilize both 1X and 2X droplets in this example. As shown in Fig. 5.3, all three methods use the same number of sample droplets, but the proposed method outperforms previous methods in terms of the number of operations and the number of waste droplets. Details are presented in Sect. 5.4.

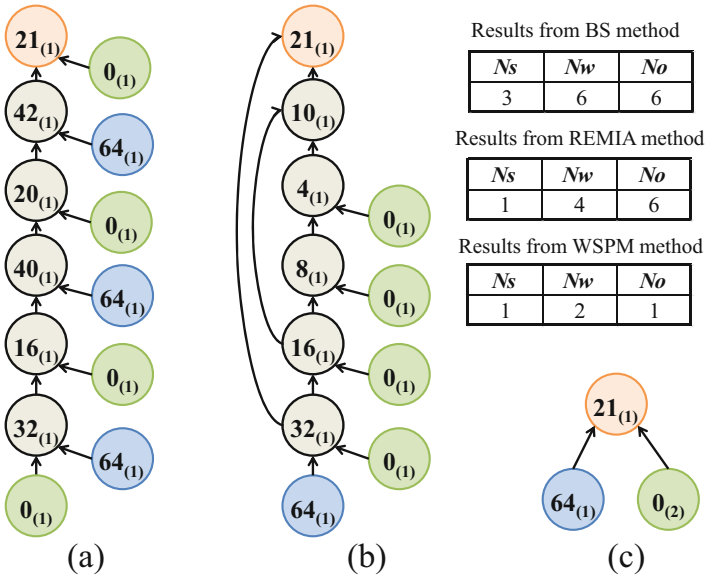



**Fig. 5.3** Dilution graphs generated by: (**a**) BS [3], (**b**) REMIA [7], and (**c**) the proposed WSPM method

### 5.3.2 Motivation for Error Recovery

One important issue in sample preparation is the correctness of the concentration of droplets. It has been observed that most of the errors in sample preparation can be attributed to volume variation in droplets [15, 17]. Furthermore, those volume variations can result in concentration errors in target droplets. As reported in [1, 9], major errors in sample preparation are dispensing errors and unbalanced splitting.

Previous error-recovery techniques for conventional DMFBs are mainly based on repeating previous operations [18] or using a *roll-back strategy* [13, 19]. More recently, an error-recovery technique has been proposed to utilize erroneous droplets to mutually cancel or reduce the concentration error when the target droplet is reached [9]. However, all these methods relay on the availability of on-chip sensors. Due to cost constraints, only a limited number of sensors can be included in a DMFB [13]. As a result, the latency for error detection during sample preparation is significant. Since MEDA provides real-time sensing, error recovery can be carried out more efficiently during sample preparation.

### 5.3.3 Problem Formulation

The problem of error-correcting sample preparation on MEDA biochips is formally described as follows.

**Inputs**

1. The target droplet with concentration $C_t$ and size $S_t$;
2. The degree of accuracy ($\delta$) of the target droplet;
3. The droplet-size constraint ($S_c$).

**Outputs**

1. A sample preparation process (described by the dilution graph) that can generate the target droplet;
2. Error-recovery procedures for any potential concentration errors during the sample preparation.

**Objective**

The derived sample-preparation process and error-recovery procedures should have minimum cost in terms of the numbers of sample droplets, waste droplets, and operations. In this chapter, we address a three-priority optimization problem. The first ("most important") priority is to minimize the number of sample droplets used, the second priority is to minimize the number of dilution operations, and the third priority is to minimize the number of generated waste droplets.

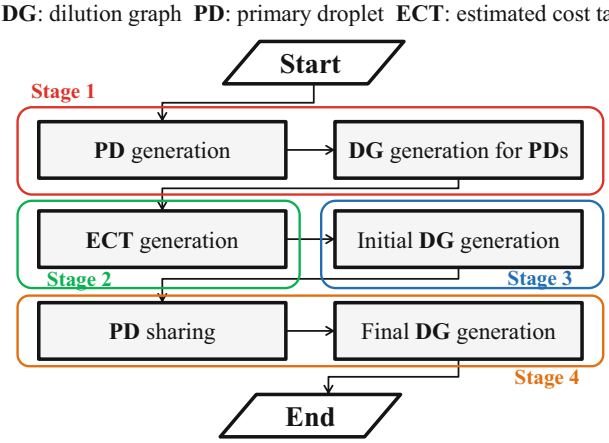



**Fig. 5.4** Algorithm flow for the proposed sample-preparation method

## 5.4 Proposed Sample Preparation Method

As outlined in Fig. 5.4, there are six steps in the proposed sample-preparation algorithm to generate a dilution strategy that can generate the target droplet: (1) primary droplet (PD) generation, (2) dilution graph (DG) generation for all PDs, (3) estimated cost table (ECT) generation, (4) initial DG generation for the target droplet, (5) PD sharing to reduce the generation cost, and (6) final DG generation. Since various mixing models can be supported on a MEDA biochip, we refer to the method for DG generation as *weighted sample-preparation method* (WSPM). Above six steps in WSPM can be categorized in four stages: primary droplet generation, estimated cost table construction, initial DG generation, and primary-droplet sharing.

### *5.4.1 Primary Droplet Generation*

In WSPM, a *primary dilution operation* is defined as a MEDA-enable dilution operation where one of the two input droplets must be a buffer droplet. According to (5.1), the concentration $C_{out}$ after a primary dilution operation can be calculated as $C_{out} = ||C_{in} \times d||$, where $d$ $(0 < d < 1)$ is the concentration-scaling factor (CSF).

On a conventional DMFB with a (1:1) mixing/splitting model, $d$ is always 1/2. However, since various mixing/splitting models are available on a MEDA biochip, different CSFs can be achieved. For example, if the droplet-size constraint $S_c$ is 3,

```
1: Start: input S_c and δ;
2: S ← {all sample droplets under constraint S_c};
3: B ← {all buffer droplets under constraint S_c};
4: P ← ∅; N ← ∅;
5: for each droplet d_s in S do
6:     P ← P⋃{d_s}, N ← N⋃{d_s};
7:     while N ≠ ∅ do
8:         N ← ∅;
9:         for each droplet d_n in N do
10:            for each droplet d_b in B do
11:                PD ← PD_generation{d_n, d_b};
12:                N ← N⋂{PD − PD⋂P};
13:                Return associated costs for each droplet in N;
14:                Return waste droplets for each droplet in N;
15:            end for
16:        end for
17:        P ← P⋃N;
18:    end while
19: end for
20: Output all droplets in P and their associated costs;
21: Output dilution graph (DG) for all droplets in P;
22: End
```

**Fig. 5.5** Pseudocode for enumerating all PDs

i.e., the maximum droplet size is 3X, (1:1), (1:2), and (2:1) mixing/splitting models can be utilized, and the corresponding CSFs are 1/2, 1/3, and 2/3, respectively. A droplet that can be produced through a series of primary dilution operations is referred to as a primary droplet (PD). Other droplets are referred to as intermediate droplets (IDs). Note that initial sample and buffer droplets are also PDs. The objective of PD generation is to obtain the DG and associated generation cost for each PD under (1) the droplet-size constraint, and (2) the concentration-accuracy constraint.

Given $S_c$ and $\delta$, all PDs and their associated generation costs are enumerated through a bread-first search (BFS)-based algorithm. The pseudocode for enumerating all PDs is shown in Fig. 5.5. All sample and buffer droplets with different sizes are stored in the set $S$ and $B$, respectively (lines 2–3). Sets $N$ and $P$ are created for newly generated droplets and PDs, respectively (line 4). Droplets in set $N$ are exhaustively diluted with buffer droplets in set $B$ until no new PD can be generated (lines 5–16). Function $PD\_generation$ is utilized to generate all possible droplets (with different sizes) after a MEDA-enable dilution operation (line 11). The cost associated with the generation of each PD is calculated and stored (line 13). Through the enumeration method, DGs for PDs can also be obtained. Based on generated DGs, waste droplets associated with each PD can also be obtained using the method proposed in [7] (line 14). The information about waste droplets is used in the step of PD sharing in Sect. 5.4.4.

### 5.4.2 Estimated Cost Table Generation

After PD generation, generation costs for all PDs are obtained and stored. However, generation costs associated with IDs are still unknown. Here we describe a method to estimate generation costs for all IDs and to construct the estimated cost table (ECT). The cost information in the ECT will be used in the next step, viz., initial DG generation for the target droplet.

For the sake of simplicity, we use $SC(D_i)$, $WC(D_i)$, and $OC(D_i)$ to represent the sample cost, waste cost, and operation cost associated with droplet $D_i$, respectively. If $D_i(C_i,S_i)$ is used to represent a droplet $D_i$ with concentration $C_i$ and size $S_i$ in the ECT, droplets $D_1(C_1,S)$ and $D_2(C_2,S)$ can be used to generate a droplet $D_3(||(C_1+C_2)/2||, S)$ through a MEDA-enable dilution operation. If the generation costs for droplets $D_1$ and $D_2$ are known, generation cost for droplet $D_3$ can then be estimated using following equations.

$$SC(D_3) = SC(D_1) + SC(D_2) \tag{5.3}$$

$$WC(D_3) = WC(D_1) + WC(D_2) + |S| \tag{5.4}$$

$$OC(D_3) = OC(D_1) + OC(D_2) + 1 \tag{5.5}$$

Here we define droplets with known generation costs as "known droplets", and other droplets as "unknown droplets". Before the step of ECT generation, only PDs are "known droplets", and all IDs are "unknown droplets". We then recursively apply (5.3)–(5.5) to estimate generation costs for "unknown droplets" until all droplets become "known droplets". Finally, estimated costs for all IDs are also stored in the ECT.

### 5.4.3 Initial Dilution Graph Generation

In this step, an initial dilution graph (IDG) for the given target droplet $D_t$ is built. Note that $D_t$ is the root node in the IDG, and the IDG is constructed in a top-down manner, i.e., from the root node to leaf nodes. In the IDG, all leaf nodes are PDs. Therefore, the dilution graph for each leaf node can be obtained through PD-generation step.

The algorithm for IDG construction is based on a greedy strategy. The pseudocode for generating IDGs is shown in Fig. 5.6. Each node $O_i$ in the IDG represents a droplet $D_i$. In each iteration, existing leaf nodes are regarded as the parent nodes, and two new nodes $O_{c1}$ and $O_{c2}$ are derived from each parent node $O_p$, i.e., droplet $D_p$ can be generated through child droplets $D_{c1}$ and $D_{c2}$. Based on the mixing/splitting model proposed in Sect. 5.3, a droplet $D_p$ can be generated through different child-droplet pairs. We first exhaustively enumerate all possible

1: **Start**: input target droplet $D_t$;
2: Create a root node $D_t$;
3: *Parent* $\leftarrow D_t$;
4: **while** *Parent* is not a primary droplet **do**
5: *Pair* $\leftarrow$ *Child_Pair*(*Parent*); *SP* $\leftarrow$ *Select*(*Pair*);
6: Select the pair ($D_{c1}$, $D_{c2}$) with the minimum cost;
7: **if** Child droplet $D_{c2}$ is not a primary droplet **then**
8: *Parent* $\leftarrow D_{c2}$;
9: **end if**
10: **end while**
11: **End**: output generated IDG.

**Fig. 5.6** Pseudocode for generating initial dilution graphs

child-droplet pairs that can generate the parent droplet $D_p$, and the pairs are stored in the set $Pair$ (line 5). Since each iteration (from parent node to child nodes) results in an increase in cost, we impose two additional constraints on child droplets to reduce the number of iterations. For each parent droplet $D_p$, (1) the left child droplet $D_{c1}$ must be a PD, and (2) the concentration of the right child droplet $D_{c2}$ must be smaller than the concentration of $D_p$. Based on above two constraints, a number of feasible child-droplet pair candidates can be selected using the selection function $Select$. The generation cost for each child droplet is calculated based on the generated ECT. The generation cost for a child-droplet pair is the summation of the generation costs for corresponding child droplets. In accordance with the three-priority optimization problem formulated in Sect. 5.3.3, we select the pair with the minimum sample cost. If two pairs have the same sample cost, the pair with the minimum operation cost is selected. If two pairs have the same sample and operation costs, the pair with the minimum number of generated waste droplets is selected. The newly generated child droplet is regarded as a new parent droplet for next iteration. The IDG-construction terminates when all leaf nodes become PDs (lines 7–9).

### 5.4.4 Primary Droplet Sharing

The objective of this step is to further minimize the generation cost for the target droplet. Consider the example shown in Fig. 5.7; the level of accuracy is $\frac{1}{256}$, and the target droplet $D_t$ is a 1X droplet with a concentration of 80. As in Sect. 5.3, we let $C_{i\,(S_i)}$ to represent droplet $D_i$ with concentration $C_i$ and size $S_i$. Accordingly, the target droplet in Fig. 5.7 is labeled as $80_{(1)}$. Two PDs, $128_{(1)}$ and $32_{(1)}$, are used to generate $D_t$. Since there is a waste droplet $128_{(1)}$ associated with the PD $32_{(1)}$, no specific operations are required for generating the other PD, $128_{(1)}$. Therefore, the generation cost for $D_t$ can be further reduced through a PD sharing (PDS) procedure.

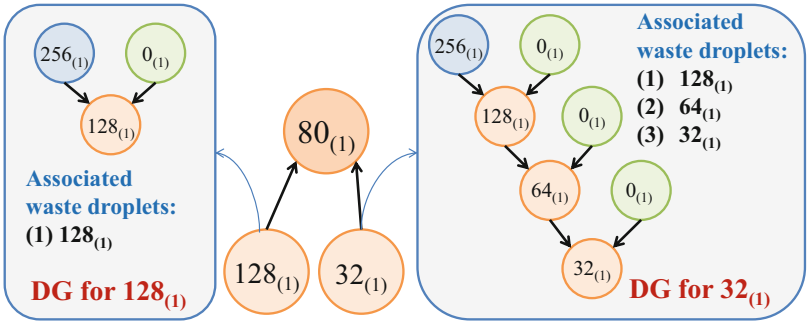



**Fig. 5.7** An example to illustrate PD sharing

1: **Start**: input $S_p$ and $W(D_i)$ for each $D_i$ in $S_p$;
2: $S_{sel} \leftarrow \varnothing$;
3: **if** $\|S_p\| \leq t_{th}$ **then**
4: $S_{sel} \leftarrow BruteForce(S_p)$;
5: **else**
6: **while** $S_p \neq \varnothing$ **do**
7: **for** each droplet $D_i$ in $S_p$ **do**
8: Calculate the benefit $B(D_i)$;
9: **end for**
10: Select the droplet $D_{max}$ with the maximum benefit;
11: $S_{sel} \leftarrow S_{sel} \bigcup \{D_{max}\}$;
12: $S_p \leftarrow S_p - \{D_{max}, W(D_{max})\}$;
13: **end while**
14: **end if**
15: Output $S_{sel}$;
16: **End**

**Fig. 5.8** Pseudocode for primary droplet sharing

The PDS problem can be formulated as follows. Inputs are (1) a set of PDs $S_p$, and (2) a set of waste droplets $W(D_i)$ associated with each PD $D_i$ in $S_p$. For the sake of simplicity, we define the union of (1) a subset $S_{psub}$ of $S_p$, and (2) the set of associated waste droplets from $S_{psub}$ as an *augmented set* ($S^{aug}_{psub}$) of $S_{psub}$. The objective is to select a subset $S_{sel}$ of $S_p$ such that $S^{aug}_{sel}$ can completely cover $S_p$. At the same time, the generation cost of $S_{sel}$, i.e., the summation of generation costs of each PD in $S_{sel}$, should be minimized. The generation cost for a PD is the sum of the corresponding sample, waste and operation cost.

The pseudocode for PDS is presented in Fig. 5.8. We can use an $O(2^N)$ brute-force method, where $N$ is the number of PDs in $S_p$, to exhaustively enumerate all possible subsets of $S_p$ and select the one with the minimum sample cost. If

two subsets are with the sample cost, the one with the minimum operation cost is selected. If two subsets are with the same sample cost and operation cost, the one with the minimum number of generated droplets is selected. However, the brute-force method is not practical for large $N$. Therefore, we propose a hybrid method solving the PDS problem. If $N$ is no larger than a threshold $t_{th}$, the brute-force method is applied to select the optimal $S_{sel}$ (lines 3–4). If $N$ is larger than $t_{th}$, a stepwise O($N$) algorithm is applied (line 5–11). For each iteration in the stepwise algorithm, the associated benefit $B(D_i)$ is calculated for each droplet $D_i$ in $S_p$ (line 7–8). The benefit $B(D_i)$ is calculated using (5.6) based on the intersection $S_{int}$ of the set $S_p$ and $S_{D_i}$={$D_i$, $W(D_i)$}, i.e., $S_{int}$=$S_p \bigcap S_{D_i}$. Parameters $Sample_{max}$, $Waste_{max}$, and $Opt_{max}$ represent the maximum value of sample cost, waste cost, and operation cost for all PDs.

$$B(D_i) = \sum_{D_j \in S_{int}} \left( \frac{Sample(D_j)}{Sample_{max}} + \frac{Waste(D_j)}{Waste_{max}} + \frac{Opt(D_j)}{Opt_{max}} \right) \tag{5.6}$$

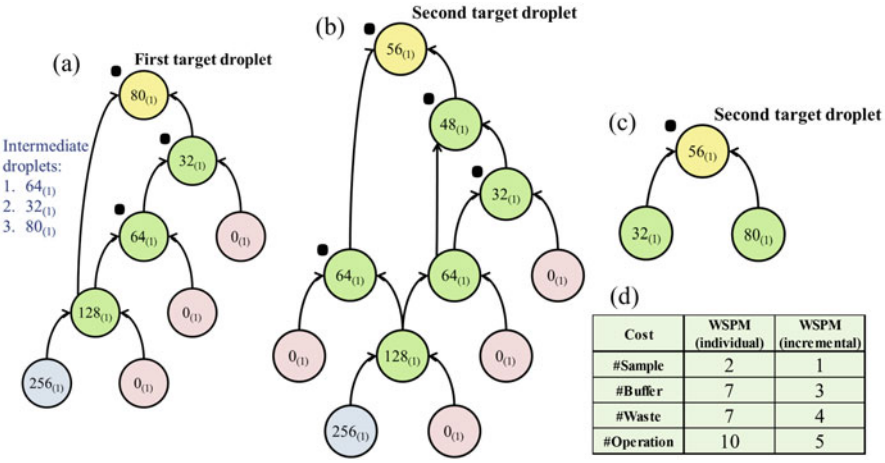
The droplet $D_{max}$ with the maximum benefit value is selected and stored in $S_{sel}$ (line 9–10). At the end of each iteration, $S_p$ is also updated (line 11). The procedure stops when $S_p$ is empty.

### 5.4.5 *Incremental Sample Preparation*

There can be multiple target droplets for one single biochemical application, i.e., protein crystallization requires concentrations of 3%, 10%, and 20% of ammonium sulfate droplets in series [20]. Applying the WSPM for the preparation of multiple target droplets individually will produce a large number of waste droplets and also require a large number of dilution operations, which may result in an increase in sample-preparation time. Therefore, we also consider incremental sample-preparation, i.e., utilizing the generated droplets to sequentially generate new target droplets.

WSPM can be extended to handle the incremental sample-preparation problem. As described in Sect. 5.4.1, the dilution graph and associated generation cost for each primary droplet can be easily obtained using a BFS-based algorithm. Here we define these primary droplets as *real primary droplets* (RPDs), i.e., these primary droplets can always be generated using a series of primary dilution operations. Besides TPDs, we define those existing on-chip droplets (e.g., waste droplets generated by previous dilution operations) as *pseudo primary droplets* (PPDs), i.e., they can be used only once for subsequent sample preparation since the dilution graphs for these droplets are still unknown. Since PPDs have already been generated, the associated generation costs for PPDs are zero.

For the incremental sample preparation, WSPM is first invoked to generate the dilution graph for the first target droplet. All generated intermediate droplets will




| Cost | WSPM (individual) | WSPM (incremental) |
|---|---|---|
| #Sample | 2 | 1 |
| #Buffer | 7 | 3 |
| #Waste | 7 | 4 |
| #Operation | 10 | 5 |

**Fig. 5.9** (**a**) Dilution graph for the first target droplet, (**b**) dilution graph for the second target droplet using individual sample preparation, (**c**) dilution graph for the second target droplet using incremental sample preparation, and (**d**) generation cost comparison. Black dots indicate that there are waste droplets associated with the corresponding operation

be used as PPDs for the generation of the next target droplet. This procedure continues until all target droplets are generated. In this way, intermediate droplets can be efficiently utilized during sample preparation and generation costs example is presented in Fig. 5.9.

In Fig. 5.9, the degree of accuracy is $\frac{1}{256}$ and two targets include a 1X droplet with the concentration of 80 ($80_{(1)}$) and a 1X droplet with the concentration of 56 ($56_{(1)}$). If we generate two target droplets individually using WSPM, the dilution graphs for the first and the second target droplet are presented in Fig. 5.9a, b, respectively. However, during the generation of the first droplet $80_{(1)}$, three intermediate (waste) droplets, $64_{(1)}$, $32_{(1)}$, and $80_{(1)}$, are also generated. These three droplets can be used as PPDs for the generation of the second target droplet $56_{(1)}$. Accordingly, if we apply incremental sample preparation for the second target droplet, the updated dilution graph is presented in Fig. 5.9c. A comparison between the individual and the incremental WSPM is presented in Fig. 5.9d. This example clearly illustrates that incremental sample preparation can effectively reduce the generation costs for multiple target droplets.

## 5.5 Error Recovery in Sample Preparation

In this section, we describe the proposed error-recovery approach and connect the error-recovery approach with the proposed sample-preparation algorithm. The overall flow is presented in Fig. 5.10. As outlined in Fig. 5.10, there are three stages:

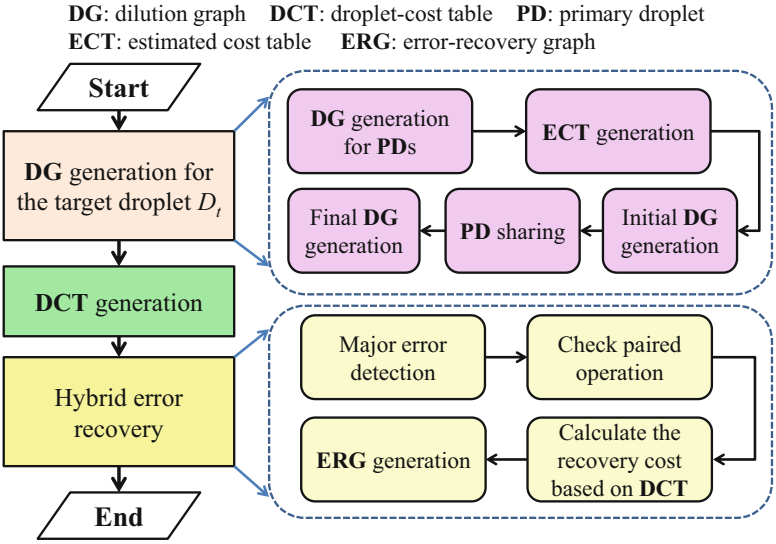



**Fig. 5.10** Overall flow combining WSPM and the proposed error-recovery approach

(1) dilution-graph (DG) generation, (2) droplet-cost table (DCT) construction, and (3) error-recovery graph (ERG) generation. First, the dilution graphs are generated for target droplets. Second, dilution graphs for all possible droplets are generated, and the corresponding generation costs are stored in a contracted ECT. The generation cost for a droplet includes the sample cost, the waste cost, and the operation cost associated with that droplet. Finally, the error-recovery procedure is invoked if an error is detected.

### *5.5.1 Droplet-Cost Table Generation*

Given the level of accuracy $\delta$ and the droplet-size constraint $S_c$, the DG for every droplet can be generated by applying WSPM, the proposed sample-preparation method. Based on generated DGs, the corresponding generation costs can be easily calculated using the method proposed in [7].

After DGs have been generated for every droplet, DGs and the corresponding generation costs are stored in a table referred to as droplet-cost table (DCT). The DCT will be used to calculate recovery cost in the error-recovery procedure proposed in Sect. 5.5.2. Since there are usually multiple ways to recovery from detected errors, the recovery cost for each approach is calculated, and the one with the minimum is selected as the recovery approach.

### 5.5.2 *Hybrid Error Recovery*

Since MEDA biochips integrate a sensing circuit under each microelectrode [21], real-time droplet sensing can be easily achieved on a MEDA biochip [22]. Based on the sensing results, we propose a hybrid error-recovery approach that combines roll-back and roll-forward techniques together to efficiently recovery from detected errors.

Previous error-recovery techniques are mainly based on repeating previous operations [18] or solely using a *roll-back strategy* [13, 19]. More recently, a *roll-back* error-recovery technique is proposed to utilize the erroneous droplets to mutually cancel or reduce the concentration error when the target droplet is reached [9]. However, almost all prior methods utilize the detected feedback at checkpoints from on-chip sensors on conventional DMFBs. Unfortunately, because of the cost constraint, there is usually a limited number of on-chip sensors on a DMFB [13]. Accordingly, real-time sensing cannot be achieved and there can also be undesirable latency when the error is detected. Therefore, prior techniques are expensive in terms of completion time and reagent cost.

#### 5.5.2.1 Error Classification

We define any concentration value in the dilution graph as a reference concentration value. Since real-time droplet sensing can be achieved on MEDA biochips, droplet sizes can be monitored during sample preparation in a real-time manner. The detected droplet sizes can be automatically mapped to the concentration value of each droplet. Based on the difference between the measured concentration value and the reference value, we classify concentration errors into two categories: *minor errors* and *major errors*. If the difference between the measured value and the reference value for one droplet is less than the degree of accuracy $\delta$, then the error is defined as a minor error. Otherwise, the error is a major error. Note that such real-time error classification can only be achieved on MEDA biochips since there are usually limited number of sensors on conventional DMFBs.

A minor error can be ignored since it does not result in perceptible change at the concentration of the target droplet. In contrast, error-recovery procedures are only required and invoked when a major error is detected.

#### 5.5.2.2 Error-Recovery Approach

We first define the concept of a paired droplet. When droplet $D_i$ is diluted with another droplet $D_j$, $D_j$ and $D_i$ are *paired droplets* with respect to each other, i.e., $D_i$ ($D_j$) is a paired droplet of $D_j$ ($D_i$).

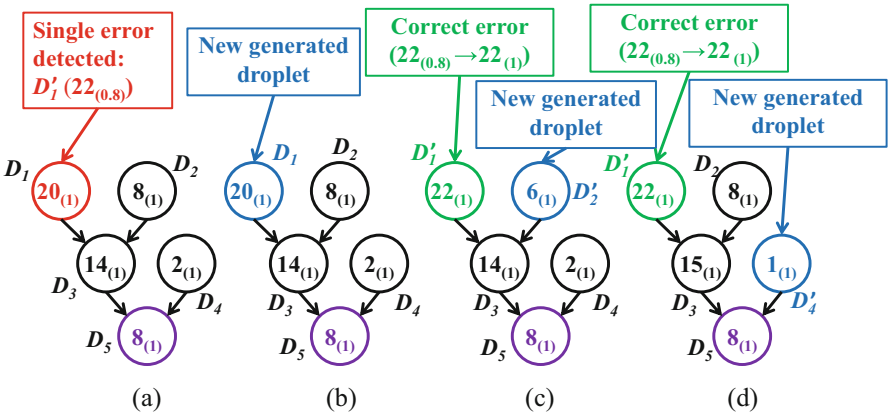



**Fig. 5.11** Illustration of (**a**) the original dilution graph, and updated dilution graph based on (**b**) the first recovery approach, (**c**) the second recovery approach, and (**d**) the third recovery approach for SES

If a major error is detected on $D_i$, we first check whether there is also a major error on $D_i$'s paired droplet $D_j$. If no major error is detected on $D_j$, we refer to this case as a *single error scenario* (SES). Otherwise, the scenario is referred to as a *paired error scenario* (PES). Note that simultaneous error detection on paired droplets can be achieved on MEDA biochips because of the real-time sensing ability. Real-time error detection will not influence the execution of other operations. On conventional DMFBs, droplets need to be serially transported to a checkpoint. The transportation may also interrupt currently executing operations.

Figure 5.11 illustrates the approaches for SES error recovery. In the original dilution graph (see Fig. 5.11a), droplet $D_1$ ($20_{(1)}$) is supposed to dilute with droplet $D_2$ ($8_{(1)}$) to generate $D_3$ ($14_{(1)}$). In Fig. 5.11a, a major error is detected on $D_1$, and the erroneous droplet is detected to be $D_1^{'}$ ($22_{(0.8)}$). Here 0.8 means these is a droplet-size error associated with the droplet, and the detected droplet size is 0.8X. The first recovery approach is to regenerate $D_1$ ($20_{(1)}$) to replace $D_1^{'}$; see Fig. 5.11b. The second approach is to utilize $D_1^{'}$; see Fig. 5.11c. The droplet-size error can be corrected using a droplet-regulation operation described in Sect. 5.2. After applying droplet regulating, $D_1^{'}$ becomes ($22_{(1)}$). A new droplet $D_2^{'}$ ($6_{(1)}$) is generated such that $D_1^{'}$ and $D_2^{'}$ can generate the desired droplet $D_3$. The third recovery approach is to utilize both $D_1^{'}$ and $D_2$; see Fig. 5.11d. Since $D_1^{'}$ and $D_2$ already exist on the chip, they can be regarded as one-time primary droplets. The incremental WSPM described in Sect. 5.4.5 is then invoked to generate a new dilution graph for the target droplet $D_5$ ($8_{(1)}$). The first and second recovery methods can be regarded as roll-back strategies since droplets are re-generated, while the third recovery approach can be regarded as a roll-forward strategy since the dilution graph is updated.

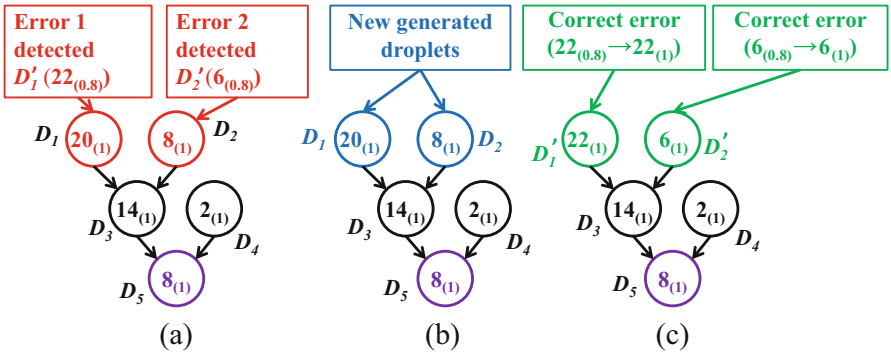



**Fig. 5.12** Illustration of (**a**) the original dilution graph, and updated dilution graph based on (**b**) the first recovery approach, and (**c**) the second recovery approach for PES

The recovery cost associated with each approach can be calculated based on the DCT constructed in Sect. 5.5.1, and the approach with the minimum sample cost is selected as the recovery solution. If two approaches have the same sample cost, the one with the minimum operation cost is selected. If two approaches have the same sample and operation costs, the one with the minimum number of generated droplets is selected.

There are two possible ways to recover from major errors in the case of PES; see Fig. 5.12. Major concentration errors are detected on both $D_1$ ($20_{(1)}$) and $D_2$ ($8_{(1)}$), and the corresponding erroneous droplets are $D_1^{'}$ ($22_{(0.8)}$) and $D_1^{'}$ ($6_{(0.8)}$), respectively. The first approach (roll-back) is to regenerate $D_1$ and $D_2$ to replace the erroneous droplets; see Fig. 5.12b. The second approach (roll-forward) is to utilize erroneous droplets $D_1^{'}$ and $D_2^{'}$; see Fig. 5.12c. After droplet-size errors for droplets $D_1^{'}$ and $D_2^{'}$ are corrected, a new dilution graph can be generated using WSPM for the target droplet $D_5$ ($8_{(1)}$). Similar to SES, recovery costs for these two approaches are also calculated based on DCT, and the recovery approach with the minimum cost is selected.

## 5.6 Experimental Results

In order to evaluate the proposed WSPM, we compare it with five previous sample-preparation methods, viz., BS [3], DMRW [4], IDMA [5], FLOW [6], and REMIA [7], for single target preparation. We also compare individual WSPM and incremental WSPM for multiple target preparation. We finally inject volumetric errors into the dilution graphs for two real-life test cases, and compare the recovery cost with two previous recovery techniques, namely roll-back technique (RBT) [13] and roll-forward technique (RFT) [9].

### 5.6.1 Results on Single-Target Sample Preparation

A total 255 concentration values, ranging from $\frac{1}{256}$ to $\frac{255}{256}$, with a degree of accuracy of $\frac{1}{256}$, are considered in our simulation. All simulations are carried out on an Intel Core i7 platform with a 2.67 GHz CPU and 8 GB of RAM. Simulation results are reported in Tables 5.1, 5.2 and 5.3. Metrics $N_s$, $N_w$, and $N_o$ in these tables represent the number of sample droplets, waste droplets, operations, respectively. Abbreviation STD stands for the size of target droplets.

Recall that conventional DMFBs only support (1:1) mixing model, therefore the maximum droplet size is always 2X (two 1X droplets are mixed to be a 2X droplet) for previous sample-preparation methods ($S_c = 2$ for previous algorithms). Since MEDA can support multiple droplet sizes, $S_c$ can take different values for the proposed WSPM algorithm.

We first examine the simulation results when target droplets are set to be 1X in volume. The droplet-size constraint ($S_c$) is set to be 2X, 3X, and 4X, respectively. The numbers in Table 5.1 are averages over 255 different cases. As shown in Table 5.1, WSPM outperforms all other algorithms for operation-count minimization. Furthermore, larger $S_c$ results in slightly larger sample cost and significantly smaller operation count. Waste droplets are almost the same for different $S_c$ values.

**Table 5.1** Comparisons between different algorithms (STD = 1X)

| | | | | | | WSPM | | |
|---|---|---|---|---|---|---|---|---|
| Metrics | [13] | [14] | [15] | [16] | [17] | $S_c$=2 | $S_c$=3 | $S_c$=4 |
| $N_s$ | 4.49 | 2.98 | 3.06 | 2.22 | 2.31 | 2.48 | 2.82 | 2.97 |
| $N_w$ | 6.03 | 3.96 | 4.12 | 6.68 | 5.01 | 6.50 | 6.36 | 6.44 |
| $N_o$ | 7.03 | 8.91 | 11.32 | 10.27 | 7.70 | 8.03 | 4.97 | 4.07 |

**Table 5.2** Comparisons between different algorithms (STD = 2X)

| | | | | | | WSPM | | |
|---|---|---|---|---|---|---|---|---|
| Metrics | [13] | [14] | [15] | [16] | [17] | $S_c$=2 | $S_c$=3 | $S_c$=4 |
| $N_s$ | 4.49 | 2.98 | 3.06 | 2.22 | 2.31 | 2.48 | 2.82 | 2.97 |
| $N_w$ | 6.03 | 3.96 | 4.12 | 6.68 | 5.01 | 5.49 | 5.36 | 5.35 |
| $N_o$ | 7.03 | 8.91 | 10.27 | 10.27 | 7.70 | 8.03 | 4.97 | 4.07 |

**Table 5.3** Comparisons between different algorithms (STD = 3X)

| | | | | | | WSPM | |
|---|---|---|---|---|---|---|---|
| Metrics | [13] | [14] | [15] | [16] | [17] | $S_c$=3 | $S_c$=4 |
| $N_s$ | 8.98 | 5.96 | 6.12 | 3.33 | 3.47 | 3.14 | 3.12 |
| $N_w$ | 12.06 | 7.92 | 8.24 | 10.02 | 7.49 | 4.85 | 4.50 |
| $N_o$ | 14.06 | 17.82 | 22.64 | 15.40 | 11.55 | 5.21 | 3.95 |

When $S_c$ is 2X, the performance of WSPM is similar to REMIA [7] and FLOW [6]. When $S_c$ is 4X, WSPM reduces the operation count by 42.10%, 54.31%, 47.14%, 46.60%, and 60.08% when compared with BS [3], DMRW [4], REMIA [7], IDMA [5], and FLOW [6], respectively. We next examine the simulation results when target droplets are set to be 2X droplets. As shown in Table 5.2, the WSPM results are similar to previous WSPM results in Table 5.1. The only difference lies in the number of waste droplets, which tends to be one droplet for all droplet-size constraints.

We finally examine the simulation results when target droplets are set to be 3X droplets. Note that the size constraint $S_c$ is at least 3X. This is because the target droplets cannot be generated when $S_c$ is less than 3X. The results in Table 5.3 indicate that WSPM outperforms all previous algorithms in terms of sample minimization, waste minimization, and operation-count minimization. The results also suggest that WSPM is a better solution for target droplets with larger size.

### 5.6.2 *Results on Multiple-Target Sample Preparation*

In order to compare the performance of the individual and the incremental WSPM, a set of $k$ different target droplets are randomly selected for multiple-target sample preparation. The sample-preparation order for different droplets are also randomly determined. Here we set $k$ to be 2, 3, and 4 in our simulation. The degree of accuracy and the droplet-size constraint are set to be $\frac{1}{256}$ and 2, respectively. For each $k$, 100 simulations are conducted and the average results are reported in Table 5.4. In Table 5.4, STD, IDV, and ICR represent the size of target droplets, the individual WSPM, and the incremental WSPM, respectively. Metrics $N_s$, $N_w$, and $N_o$ in these tables represent the number of sample droplets, waste droplets, operations, respectively.

**Table 5.4** Comparisons between the individual WSPM and the incremental WSPM

| | | *k=2* | | *k=3* | | *k=4* | |
|---|---|---|---|---|---|---|---|
| STD | Metrics | IDV | ICR | IDV | ICR | IDV | ICR |
| 1X | $N_s$ | 2.72 | 2.39 | 3.02 | 2.56 | 2.80 | 2.53 |
| | $N_w$ | 6.55 | 5.11 | 6.61 | 5.32 | 6.43 | 5.51 |
| | $N_0$ | 5.14 | 3.87 | v4.92 | 3.97 | 4.90 | 3.77 |
| 2X | $N_s$ | 5.70 | 3.32 | 5.89 | 3.56 | 5.87 | 3.53 |
| | $N_w$ | 11.36 | 9.44 | 11.05 | 9.06 | 11.41 | 9.68 |
| | $N_0$ | 10.44 | 7.68 | 9.98 | 8.29 | 10.29 | 8.12 |
| 3X | $N_s$ | 9.72 | 6.45 | 9.46 | 6.28 | 9.30 | 6.98 |
| | $N_w$ | 15.45 | 11.65 | 15.90 | 11.44 | 16.17 | 11.31 |
| | $N_0$ | 15.96 | 10.55 | 16.56 | 10.70 | 16.38 | 10.65 |

As shown in Table 5.4, incremental WSPM outperforms individual WSPM for all evaluation metrics, i.e., sample cost, waste cost, and operation count, regardless of the value of parameter $k$. Results in Table 5.4 further demonstrates that it is essential to consider incremental sample preparation when there are multiple target droplets.

### 5.6.3 *Results on Error Recovery*

To evaluate error recovery, we conducted simulation experiments on two real-life test cases. According to [9], 95% ($\frac{243}{256}$) ethanol is required for RNA extraction from worms and for nuclear staining of live worms; 70% ($\frac{180}{256}$) ethanol is required as a sample in the glucose tolerance test in mice for determining how quickly exogenous glucose is cleared.

We first conducted 100 splitting operations using a 7 × 7 deionized (DI) water droplet on a fabricated MEDA biochip. For each splitting operation, we detected droplet sizes of the two split droplets and determined the droplet-size variation. The droplet size variation $\delta$ can be calculated using $\delta = \frac{|S_{D1} - S_{D2}|}{\max\{S_{D1}, S_{D2}\}}$, where $S_{D1}$ and $S_{D2}$ represent the detected droplet sizes of two split droplets. Figure 5.13 presents the distribution of droplet-size variations for 100 splitting operations. The experimental results in Fig. 5.13 are used to provide some level of information. The distribution of droplet-size variation may be different for droplets with different sizes. Note that the proposed error-recovery approaches are not influenced by the distribution. If an accurate distribution is required for a certain droplet size, corresponding experiments must be executed using a droplet with the same size.

We then generated the dilution graph for two target droplets: a 1X 95% droplet and a 1X 70% droplet. We randomly injected droplet-size variations into the dilution graphs. The magnitude of the droplet-size variation for each operation is randomly selected based on experimentally derived distribution in Fig. 5.13. We applied the proposed hybrid method, RFT, and RBT for error recovery. The corresponding recovery costs, i.e., sample cost and operation cost, are calculated and presented in Fig. 5.14. The sample cost is counted using the volume of consumed sample droplets, and the operation cost is counted using the number of required operations.

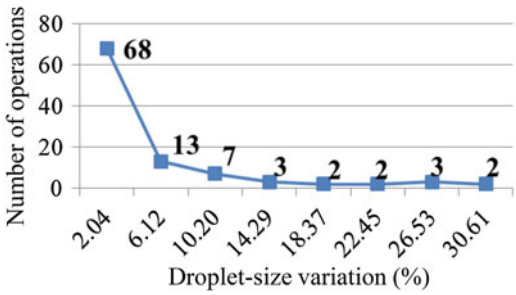



**Fig. 5.13** Experimentally-derived distribution of droplet-size variations from 100 splitting operations on a fabricated MEDA biochip

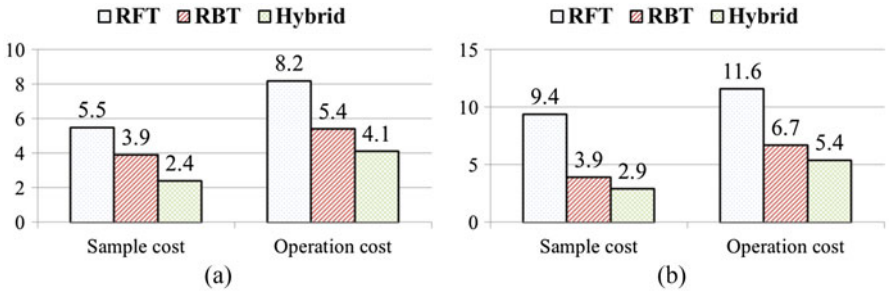



**Fig. 5.14** Error-recovery costs using different methods for (**a**) a 1X 95% droplet, and (**b**) a 1X 70% droplet

As shown in Fig. 5.14, the proposed hybrid method outperforms RFT and RBT in reducing sample cost and operation cost for both target droplets. Therefore, the proposed hybrid method is more effective compared to RFT and RBT for MEDA biochips.

## 5.7 Conclusion

This chapter presents the first sample preparation method with a general mixing/splitting model, referred to as WSPM, for MEDA biochips. Two new MEDA-enabled operations for sample preparation have been proposed and experimentally demonstrated on a fabricated MEDA biochip. In order to tolerate the inherent uncertainty of fluidic operations, we have also proposed a hybrid error-recovery technique integrated with WSPM by exploiting MEDA-specific advances in real-time droplet sensing. Simulation results on both synthetic and real-life test cases highlight the effectiveness of the proposed sample-preparation method.

## References

1. Fair, R., Srinivasan, V., Ren, H., Paik, P., Pamula, V., & Pollack, M. (2003). Electrowetting-based on-chip sample processing for integrated microfluidics. In *Proceedings of IEEE international electron devices meeting (IEDM)* (pp. 32–35).
2. Gascoyne, P. R., & Vykoukal, J. V. (2004). Dielectrophoresis-based sample handling in general-purpose programmable diagnostic instruments. *Proceedings of the IEEE, 92*, 22–42.
3. Thies, W., Urbanski, J. P., Thorsen, T., & Amarasinghe, S. (2008). Abstraction layers for scalable microfluidic biocomputing. *Natural Computing, 7*(2), 255–275.
4. Roy, S., Bhattacharya, B. B., & Chakrabarty, K. (2010). Optimization of dilution and mixing of biochemical samples using digital microfluidic biochips. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), 29*(11), 1696–1708.
5. Roy, S., Bhattacharya, B., & Chakrabarty, K. (2011). Waste-aware dilution and mixing of biochemical samples with digital microfluidic biochips. In *Proceedings of IEEE/ACM design, automation and test conference in Europe (DATE)* (pp. 1–6).

6. Dinh, T. A., Yamashita, S., & Ho, T.-Y. (2014). A network-flow-based optimal sample preparation algorithm for digital microfluidic biochips. In *Proceedings of IEEE/ACM Asia and South Pacific design automation conference (ASP-DAC)* (pp. 225–230).
7. Liu, C.-H., Chiang, T.-W., & Huang, J.-D. (2015). Reactant minimization in sample preparation on digital microfluidic biochips. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), 34*, 1429–1440.
8. Chiang, T.-W., Liu, C.-H., & Huang, J.-D. (2013). Graph-based optimal reactant minimization for sample preparation on digital microfluidic biochips. In *Proceedings of IEEE international symposium on VLSI design, automation, and test (VLSI-DAT)* (pp. 1–4). Piscataway: IEEE.
9. Poddar, S., Ghoshal, S., Chakrabarty, K., & Bhattacharya, B. B. (2016). Error-correcting sample preparation with cyberphysical digital microfluidic lab-on-chip. *ACM Transactions on Design Automation of Electronic Systems (TODAES), 22*, 2:1–2:29.
10. Huang, C.-M., Liu, C.-H., & Huang, J.-D. (2015). Volume-oriented sample preparation for reactant minimization on flow-based microfluidic biochips with multi-segment mixers. In *Proceedings of IEEE Design, Automation & Test in Europe Conference & Exhibition (DATE), 2015* (pp. 1114–1119).
11. Bhattacharjee, S., Poddar, S., Roy, S., Huang, J.-D., & Bhattacharya, B. B. (2017). Dilution and mixing algorithms for flow-based microfluidic biochips. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), 36*(4), 614–627.
12. Bera, N., Majumder, S., & Bhattacharya, B. B. (2016). Simulation-based method for optimum microfluidic sample dilution using weighted mix-split of droplets. *IET Computers & Digital Techniques, 10*(3), 119–127.
13. Luo, Y., Chakrabarty, K., & Ho, T.-Y. (2013). Error recovery in cyberphysical digital microfluidic biochips. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), 32*(1), 59–72.
14. Li, Z., Lai, K. Y.-T., Yu, P.-H., Chakrabarty, K., Ho, T.-Y., & Lee, C.-Y. (2016). Built-in self-test for micro-electrode-dot-array digital microfluidic biochips. In *Proceedings of IEEE international test conference (ITC)* (pp. 1–10).
15. Ren, H., Srinivasan, V., & Fair, R. B. (2003). Design and testing of an interpolating mixing architecture for electrowetting-based droplet-on-chip chemical dilution. In *Proceedings of IEEE international conference on solid-state sensors, actuators and microsystems (TRANSDUCERS)* (pp. 619–622).
16. Wang, G., Teng, D., Lai, Y.-T., Lu, Y.-W., Ho, Y., & Lee, C.-Y. (2013). Field-programmable lab-on-a-chip based on microelectrode dot array architecture. *IET Nanobiotechnology, 8*(3), 163–171.
17. Hsieh, Y.-L., Ho, T.-Y., & Chakrabarty, K. (2014). Biochip synthesis and dynamic error recovery for sample preparation using digital microfluidics. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), 33*, 183–196.
18. Alistar, M., Pop, P., & Madsen, J. (2012). Online synthesis for error recovery in digital microfluidic biochips with operation variability. In *Proceedings of IEEE design, test, integration and packaging of MEMS/MOEMS (DTIP)* (pp. 53–58).
19. Jaress, C., Brisk, P., & Grissom, D. (2015). Rapid online fault recovery for cyber-physical digital microfluidic biochips. In *Proceedings of IEEE VLSI test symposium (VTS)* (pp. 1–6).
20. Smith, P. E., Krohn, R. I., Hermanson, G., Mallia, A., Gartner, F., Provenzano, M., Fujimoto, E., Goeke, N., Olson, B., & Klenk, D. (1958). Measurement of protein using bicinchoninic acid. *Analytical Biochemistry, 150*(1), 76–85.
21. Lai, K. Y.-T., Shiu, M.-F., Lu, Y.-W., Ho, Y.-C., Kao, Y.-C., Yang, Y.-T., Wang, G., Liu, K.-M., Chang, H.-C., & Lee, C.-Y. (2015). A field-programmable lab-on-a-chip with built-in self-test circuit and low-power sensor-fusion solution in 0.35 μm standard CMOS process. In *Proceedings of IEEE Asian Solid-State Circuits Conference (A-SSCC)* (pp. 1–4).
22. Li, Z., Lai, K. Y.-T., Yu, P.-H., Chakrabarty, K., Pajic, M., Ho, T.-Y., & Lee, C.-Y. (2016). Error recovery in a micro-electrode-dot-array digital microfluidic biochip. In *Proceedings of IEEE/ACM international conference on computer-aided design (ICCAD)* (pp. 105:1–105:8).

# Chapter 6
# Conclusions and New Directions


## 6.1 Book Summary

In this book, we have presented a set of design automation techniques for MEDA biochips. Contributions include automated synthesis of bioassays, development of error-recovery techniques, optimization of on-chip sample preparation procedures, and development of test methods for MEDA biochips. In contrast to previous conceptual methods for conventional DMFBs, the proposed techniques address practical issues that arise in the design, utilization, and maintenance of MEDA biochips. MEDA-specific advantages, e.g., real-time droplet sensing and droplet shape morphing, have been fully investigated in this book. The proposed techniques have also been validated using experiments on fabricated chips. By bridging the gap between theory and realistic applications on fabricated chips, these techniques provide efficient, powerful, and practical design-automation solutions for MEDA biochips.

Chapter 2 described a high-level synthesis method targeting reservoir placement, operation scheduling, module placement, and routing of droplets of various sizes in a MEDA biochip. We have presented a velocity model for droplets with different sizes and shapes on a MEDA biochip. The model was validated using a fabricated MEDA biochip. We have also developed a size-aware droplet router, which is able to route droplets with different sizes and shapes on a MEDA biochip. Routing constraints (including both static and dynamic constraints) specific to MEDA biochips have been formulated and experimentally validated. In order to effectively exploit the advantages offered by MEDA, MEDA-specific fluidic operations have been considered in the proposed synthesis approach.

Chapter 3 presented a new probabilistic-timed-automata (PTA)-based strategy for error recovery in MEDA biochips. We have proposed a classification of the outcomes of operations into different categories. Instead of utilizing the same error-recovery strategy for all types of errors, different PTA-based error-recovery strategies are proposed for different types of local errors. An on-line synthesis



Z. Li et al., *Micro-Electrode-Dot-Array Digital Microfluidic Biochips*,
https://doi.org/10.1007/978-3-030-02964-7_6

approach and a control flow have also been proposed to connect the local recovery procedures to global error recovery. Moreover, an integer linear programming (ILP)-based method has been utilized to determine 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.

Chapter 4 proposed both structural and functional test methods for MEDA biochips. Various defects and malfunctions that are typical for MEDA biochips have been presented and related to developed fault models. Efficient test techniques were then proposed for structural test. A capacitive voltage divider has been proposed and integrated into each micro cell to facilitate structural test. Cost-effective functional test techniques have also been proposed to test fundamental microfluidic operations on a target MEDA biochip. These techniques can identify "qualified regions" (i.e., groups of microelectrodes that pass functional test). Therefore, the synthesis tool can map modules only to these qualified regions and the reliability of the synthesized biochip is significantly increased.

Chapter 5 introduced a droplet size-aware and error-correcting sample preparation method, referred to as the weighted sample-preparation method (WSPM), for MEDA biochips. In contrast to previous methods, the proposed approach considers droplet sizes and incorporates various mixing models in sample preparation. It utilizes MEDA-enabled microfluidic operations, and fully exploits the feature of real-time droplet sensing on MEDA biochips. Corresponding MEDA-enabled microfluidic operations are validated on a fabricated chip. We have been able to significantly reduce the number of dilution steps, hence the time needed for sample preparation.

## 6.2 Future Research Directions

The contents of this book open up a number of new research directions related to the automated design and utilization of MEDA biochips. Compared with conventional DMFBs, MEDA biochips can provide a higher degree of "intelligence" with the help of real-time sensing feedback. Therefore, more adaptive design techniques can be investigated to fully utilize the benefits provided by MEDA architecture. Some potential new directions are summarized below.

### *6.2.1 Adaptive On-Line Synthesis for MEDA Biochips*

While a bioassay is executed on a DMFB, operations do not always match their specified behavior. For example, droplets can encounter changes in their volume during the execution of bioassays. Droplet evaporation is one potential reason for

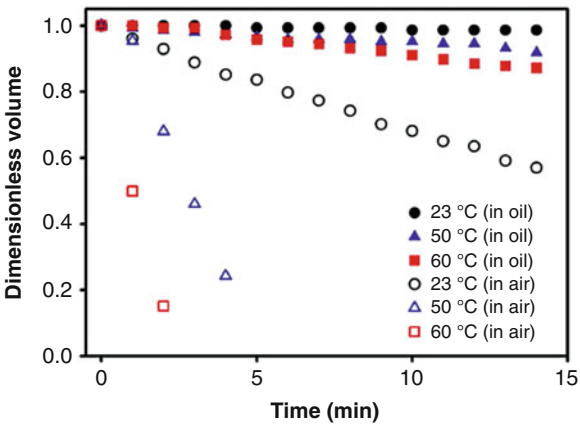



**Fig. 6.1** Variations of droplet volume in ambient air and silicone oil at different temperatures [1]

the volume change. Figure 6.1 presents the volume change with time at different temperatures. Moreover, mixing and splitting operations can also result in the change of droplet volumes; this can happen because of parametric faults, such as an electrode coating fault or unequal actuation voltages.

Furthermore, due to the inherent variability and randomness of biochemical processes, the problem of completion-time uncertainties in fluidic operations remains even after a careful characterization of the bioassay. Therefore, the module library that defines the completion times of operations as constants is oversimplified and cannot be utilized to precisely model the fluidic operations.

Hence, off-line synthesis techniques usually cannot guarantee a successful execution of the target bioassay on a MEDA biochip. Therefore, it is necessary to address the biochemical operation variability during the synthesis. As we have seen in this book, MEDA biochips provide a powerful droplet sensing scheme, which can be used to track the progress of each ongoing operation on a chip. Based on the droplet sensing on MEDA biochips, we are able to develop an adaptive on-line synthesis approach that can adjust the executing of operations in a real-time manner.

### *6.2.2 Dynamic Droplet Routing on a MEDA Biochip*

Droplet routing on a MEDA biochip determines droplet routes from their initial positions to their desired positions, such that unintentional droplet mixing is avoided and on-going fluidic operations are bypassed. The objective is to minimize the droplet routing time. There has been considerable amount of work on the droplet routing for conventional DMFBs [2–6]. Some off-line droplet-routing technique [7, 8] have also been proposed for MEDA biochips.

Once an offline routing solution is obtained, the solution will be stored in a memory component (e.g., ROM in the control board). The functional electrodes will then be configured and activated to transport droplets during runtime. Since defects in MEDA biochips can be latent, droplets can get stuck during droplet transportation even if chip is carefully tested before use. Therefore, it is necessary to develop techniques that can dynamically change droplet routes when an unexpected situation arises.

Because of droplet sensing on MEDA biochips, we can precisely detect the location of each on-chip droplet during its transportation. Once a stuck droplet is detected, dynamic router needs to be invoked and generate an alternative droplet route in order to transport the droplet to its destination. Note that fluidic constraints and the constraints imposed by executing operations are needed to be satisfied.

### *6.2.3 Application-Based Chip-Level Synthesis for MEDA Biochips*

Conventional DMFBs are typically designed using custom methods, i.e., discrete microfluidic components are first designed and then integrated. Unlike conventional DMFBs, which use a fixed electrode layout and fixed-sized electrodes to perform fluidic operations, MEDA biochips are based on the concept of a sea-of-microelectrode array, i.e., microelectrodes can be dynamically configured to form functional electrodes and fluidic modules.

Just as field programmable gate array (FPGA) needs to be configured before it can deliver the target functionality, MEDA biochips also need to be configured before they can be used for different bio-applications. An example is presented in Fig. 6.2. Before configuration, a blank MEDA biochip is shown in Fig. 6.2a.

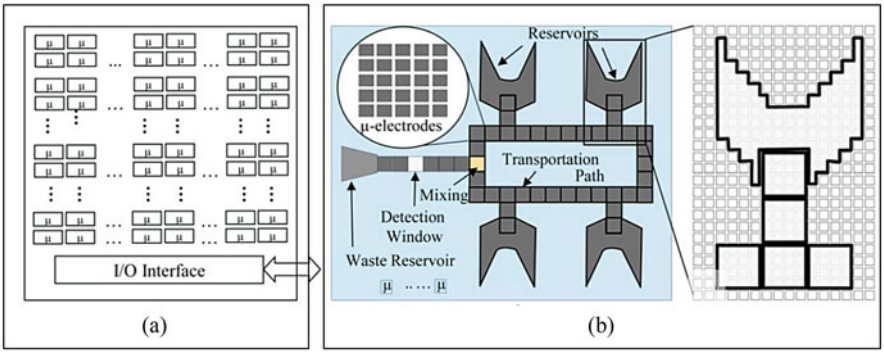



**Fig. 6.2** (**a**) A blank MEDA biochip before programming, and (**b**) the configured MEDA biochip after programming [9]

Figure 6.2b presents the MEDA biochip after configuration. The configured MEDA biochip can be for polymerase chain reaction (PCR) through electrode programming.

Therefore, a chip-level synthesis technique is required to configure a "blank" chip into a functional biochip before utilizing high-level synthesis techniques to map the bioassay on the chip.

### *6.2.4 Fault-Tolerant Design of MEDA Biochips*

Since MEDA biochips can be used for safety-critical biomedical applications, the reliability issues associated with MEDA biochips are particularly important. Because of the reconfigurability and the ability to provide real-time sensing feedbacks, MEDA biochips present interesting opportunities for the development of fault-tolerance techniques. Fault-tolerant designs usually consist of two main components: fault detection and repair.

The first step of a fault-tolerant scheme is fault detection. The purpose of fault detection is to alert the supervising process that actions are needed to correct detected errors and to identify faulty components. Since MEDA biochips can provide real-time sensing feedback and monitor operations, on-line errors can be easily detected during the execution of bioassays. At the same time, faulty microelectrodes can be discarded for the subsequent operations.

After an error is detected and located, the chip needs to be repaired. Since a MEDA biochip is highly reconfigurable, microcells can be arbitrarily grouped to form fluidic modules. Therefore, it is easy to re-execute the faulty operation on a new configured fluidic module. However, how to efficiently utilize sparse resources and how to minimize the interference on executing operations are open problems. Moreover, roll-forward techniques [10] can also be utilized for fault tolerance on MEDA biochips

### *6.2.5 Verification of Synthesis Results for MEDA Biochips*

After a bioassay is mapped to a MEDA biochip, the synthesis results need to be carefully checked before the chip can be utilized. This necessitates the development of automated verification tool to verify the synthesis results against the design constraints. The constraints include (1) fluidic constraints (e.g., static and dynamic constraints descried in Chap. 2), (2) constraints imposed by the input protocol (e.g., one operation must be completed before the start of another operation), or (3) constraints imposed by the target chip architecture (e.g., waste droplets must be collected within the waste reservoir).

Formal verification strategies can be utilized to achieve this objective. An initial automated verification framework has been presented in [11], and the proposed

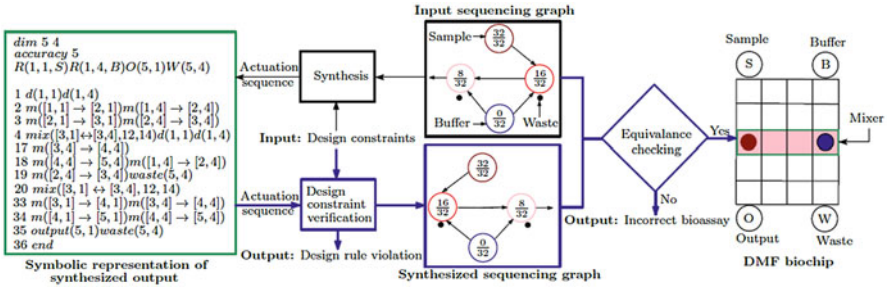



**Fig. 6.3** Illustration of the verification flow proposed in [11]

verification flow is shown in Fig. 6.3. There are two steps in the proposed verification flow. The first step is to formulate the analysis problem as an instance of a satisfiability problem (SAT)-based problem over Boolean variables and to check for design constraints; The second step is to utilize a general method to verify the correctness of the synthesized results. Since MEDA can provide more advanced operations, e.g., diagonal droplet splitting and droplet shape morphing, there are more design constraints to be considered. Moreover, since MEDA can integrate a large number of microelectrodes in a single chip, a SAT-based approach may no longer be feasible for MEDA biochips. Accordingly, new verification techniques are required for MEDA biochips.

## References

1. Hong, J., Kim, Y. K., Won, D.-J., Kim, J., & Lee, S. J. (2015). Three-dimensional digital microfluidic manipulation of droplets in oil medium. *Scientific Reports, 5*, 1–11.
2. Su, F., Hwang, W., & Chakrabarty, K. (2006). Droplet routing in the synthesis of digital microfluidic biochips. In *Proceedings of IEEE/ACM Design, Automation and Test Conference in Europe, 1*, pp. 1–6.
3. Huang, T.-W., Yeh, S.-Y., & Ho, T.-Y. (2011). A network-flow based pin-count aware routing algorithm for broadcast-addressing EWOD chips. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 30*(12), 1786–1799.
4. Zhao, Y., & Chakrabarty, K. (2012). Cross-contamination avoidance for droplet routing in digital microfluidic biochips. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 31*(6), 817–830.
5. Huang, T.-W., & Ho, T.-Y. (2011). A two-stage integer linear programming-based droplet routing algorithm for pin-constrained digital microfluidic biochips. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 30*, 215–228.
6. Chakraborty, S., Chakraborty, S., Das, C., & Dasgupta, P. (2016). Efficient two phase heuristic routing technique for digital microfluidic biochip. *IET Computers and Digital Techniques, 10*, 233–242.
7. Chen, Z., Teng, D. H.-Y., Wang, G. C.-J., & Fan, S.-K. (2011). Droplet routing in high-level synthesis of configurable digital microfluidic biochips based on microelectrode dot array architecture. *Biochip Journal, 5*(4), 343–352.

8. Keszocze, O., Li, Z., Grimmer, A., Wille, R., Chakrabarty, K., & Drechsler, R. (2017). Exact routing for micro-electrode-dot-array digital microfluidic biochips. In *Proceedings of IEEE/ACM Asia and South Pacific Design Automation Conference*, pp. 708–713.
9. Wang, G., Teng, D., Lai, Y.-T., Lu, Y.-W., Ho, Y., & Lee, C.-Y. (2013). Field-programmable lab-on-a-chip based on microelectrode dot array architecture. *IET Nanobiotechnology, 8*(3), 163–171.
10. Pradhan, D. K., & Vaidya, N. H. (1994). Roll-forward checkpointing scheme: A novel fault-tolerant architecture. *IEEE Transactions on Computers, 43*(10), 1163–1174.
11. Bhattacharjee, S., Banerjee, A., Chakrabarty, K., & Bhattacharya, B. B. (2014). Correctness checking of bio-chemical protocol realizations on a digital microfluidic biochip. In *Proceedings of IEEE International Conference on VLSI Design*, pp. 504–509.

# Index





Z. Li et al., *Micro-Electrode-Dot-Array Digital Microfluidic Biochips*,
https://doi.org/10.1007/978-3-030-02964-7