

Optimization Tools for the Design of Reconfigurable Digital Microfluidic Biochips

By

Tao Xu

Department of Electrical and Computer Engineering
Duke University

Date _____

Approved:

Prof. Krishnendu Chakrabarty, Advisor

Prof. Richard B. Fair

Prof. Chris Dwyer

Dr. Vamsee K. Pamula

Prof. Lingchong You

Dissertation submitted in partial fulfillment of
the requirements for the degree of Doctor
of Philosophy in the Department of
Electrical and Computer Engineering in the Graduate School
of Duke University

2008

ABSTRACT

Optimization Tools for the Design of Reconfigurable Digital Microfluidic Biochips

By

Tao Xu

Department of Electrical and Computer Engineering
Duke University

Date _____

Approved:

Prof. Krishnendu Chakrabarty, Advisor

Prof. Richard B. Fair

Prof. Chris Dwyer

Dr. Vamsee K. Pamula

Prof. Lingchong You

An abstract of a dissertation submitted in partial
fulfillment of the requirements for the degree
of Doctor of Philosophy in the Department of
Electrical and Computer Engineering in the Graduate School
of Duke University

2008

Copyright by
Tao Xu
2008

Abstract

Microfluidics-based biochips combine electronics with biochemistry to open new application areas such as point-of-care medical diagnostics, on-chip DNA analysis, automated drug discovery and protein crystallization. Bioassays can be mapped to microfluidic arrays using synthesis tools and they can be executed through the electronic manipulation of sample and reagent droplets. The 2007 International Technology Roadmap for Semiconductors articulates the need for innovations in biochip and microfluidics as part of functional diversification (“Higher Value Systems” and “More than Moore”). This document also highlights “Medical” as being a System Driver for 2009

This thesis envisions an automated design flow for microfluidic biochips, in the same way as design automation revolutionized IC design in the 80s and 90s. Electronic design-automation techniques are leveraged whenever possible, and new design-automation solutions are developed for problems that are unique to digital microfluidics. Biochip users (e.g., chemists, nurses, doctors and clinicians) and the biotech/pharmaceutical industry will adapt more easily to new technology if appropriate design tools and in-system automation methods are made available.

The thesis is focused on a design automation framework that addresses optimization problems related to layout, synthesis, droplet routing, testing, and testing for digital microfluidic biochips. Optimization goal includes the minimization of time-to-response, chip area, and test complexity. The emphasis here is on practical issues such as defects, fabrication cost, physical constraints, and application-driven design. To obtain robust, easy-to-route chip designs, a unified synthesis method has been developed to incorporate droplet routing and defect tolerance in architectural synthesis and physical design. It allows routing-aware architectural-level design choices and defect-tolerant physical design decisions to be made simultaneously.

In order to facilitate the manufacture of low-cost and disposable biochips, design methods that rely on a small number of control pins have also been developed. Three techniques have been introduced for the automated design of such pin-constraint biochips. First, a droplet-trace-based array partitioning method has been combined with an efficient pin assignment technique, referred to as the “Connect-5 algorithm”. The second pin-constrained design method is based on the use of “rows” and “columns” to access electrodes. An efficient droplet manipulation method has been developed for this cross-referencing technique. The method maps the droplet-movement problem to the clique-partitioning problem from graph theory, and it allows simultaneous movement of a large number of droplets on a microfluidic array.

The third pin-constrained design technique is referred to as broadcast-addressing. This method provides high throughput for bioassays and it reduces the number of control pins by identifying and connecting control pins with “compatible” actuation sequences.

Dependability is another important attribute for microfluidic biochips, especially for safety-critical applications such as point-of-care health assessment, air-quality monitoring, and food-safety testing. Therefore, these devices must be adequately tested after manufacture and during bioassay operations. This thesis presents a cost-effective testing method, referred to as “parallel scan-like test”, and a rapid diagnosis method based on test outcomes. The diagnosis outcome can be used for dynamic reconfiguration, such that faults can be easily avoided, thereby enhancing chip yield and defect tolerance. The concept of functional test for digital biochip has also been introduced for the first time in this thesis. Functional test methods address fundamental biochip operations such as droplet dispensing, droplet transportation, mixing, splitting, and capacitive sensing.

To facilitate the application of the above testing methods and to increase their effectiveness, the concept of design-for-testability (DFT) for microfluidic biochips has been introduced in this thesis. A DFT method has been proposed that incorporates a test plan into

the fluidic operations of a target bioassay protocol.

The above optimization tools have been used for the design of a digital microfluidic biochip for protein crystallization, a commonly used technique to understand the structure of proteins. An efficient solution-preparation algorithm has been developed to generate a solution-preparation plan that lists the intermediate mixing steps needed to generate target solutions with the required concentrations. A multi-well high-throughput digital microfluidic biochip prototype for protein crystallization has also been designed.

In summary, this thesis research has led to a set of practical design tools for digital microfluidics. A protein crystallization chip has been designed to highlight the benefits of this automated design flow. It is anticipated that additional biochip applications will also benefit from these optimization methods.

Contents

Abstract	i
List of Tables	xi
List of Figures	xii
Acknowledgements	xx
1. Introduction	1
1.1 Digital Microfluidic Technology	6
1.2 Synthesis, Testing, and Pin-Constrained Design Techniques	9
1.3 Protein Crystallization	13
1.4 Outline of Thesis	16
2. Defect-Tolerant and Routing-Aware Synthesis	20
2.1 Background	20
2.2 Routing-Aware Synthesis.	21
2.2.1 Droplet-Routability Estimation	21
2.2.2 Routing-Time Cost and Assay Completion Time	24
2.3 Defect-Tolerant Synthesis	25
2.3.1 Post-Synthesis Defect Tolerance	26
2.3.2 Pre-Synthesis Defect Tolerance	27
2.4 Simulations Results	30
2.4.1 Routing-Aware Synthesis	31
2.4.2 Post-Synthesis Defect Tolerance	36
2.4.3 Pre-Synthesis Defect Tolerance	37
2.5 Chapter Summary and Conclusions	41
3. Pin-Constrained Biochip Design	42
3.1 Droplet-Trace-Based Array-Partitioning	42

3.1.1	Impact of Droplet Interference and Electrode-Addressing Problem .	42
3.1.2	Array Partitioning and Pin-Assignment Method	47
3.1.3	Pin-Assignment Algorithm	51
3.1.4	Application to Multiplexed Bioassay	54
3.2	Cross-Referencing-Based Droplet Manipulation Method	57
3.2.1	Cross-Referencing Addressing	58
3.2.2	Interference-Free Droplet Manipulation Based on Destination-Cell Categorization	58
3.2.3	Scheduling of Routing for Efficient.	65
3.2.4	Variant of Droplet-Manipulation Method for High-Throughput Power- Oblivious Applications	70
3.2.5	Simulation Results	70
3.3	Broadcast-Addressing Method	79
3.3.1	“Don’t-Cares” in Electrode-Actuation Sequences	79
3.3.2	Optimization Based on Clique Partitioning in Graphs	80
3.3.3	Broadcast Addressing for Multifunctional Biochips	84
3.3.4	Experimental Results	84
3.4	Chapter Summary and Conclusion	91
4.	Testing and Diagnosis	93
4.1	Parallel Scan-like Test	93
4.1.1	Off-line Test and Diagnosis	97
4.1.2	On-line Test and Diagnosis	103
4.2	Diagnosis of Multiple Defects	105
4.2.1	Incorrectly-Classified defects	105
4.2.2	Untestable Sites	106
4.3	Performance Evaluation.	108

4.3.1	Complexity Analysis	109
4.3.2	Probabilistic Analysis	109
4.3.3	Occurrence Probability of Untestable Sites	112
4.4	Application to Fabricated Biochips	113
4.5	Functional Test	116
4.5.1	Dispensing Test	118
4.5.2	Routing Test and Capacitive Sensing Test	119
4.5.3	Mixing and Splitting Test	122
4.5.4	Application to Pin-constrained Chip Design	126
4.6	Simulation Result	131
4.7	Chapter Summary and Conclusions	138
5.	Design for Testability	140
5.1	Testability of a Digital Microfluidic Biochip	141
5.2	Testability-Aware Pin-constrained Chip Design	143
5.2.1	Pin-Assignment Method	144
5.2.2	Euler-Path-Based Functional Test for Irregular Chip Layouts	146
5.3	Simulation Results	148
5.3.1	Multiplexed Assay	149
5.3.2	Polymerase Chain Reaction (PCR)	151
5.4	Chapter Summary and Conclusions	152
6.	Application on Protein Crystallization	154
6.1	Protein Crystallization Chip Design	154
6.1.1	Pin-Constrained Chip Design	156
6.1.2	Shuttle-Passenger-Like Well-Loading Algorithm	161
6.1.3	Defect Tolerance	163
6.1.4	Evaluation of Well-Loading Algorithm and Defect Tolerance Pin-	

Constrained Chip Design	164
6.2 Automated Solution Preparation	167
6.2.1 Efficient Solution-Preparation Planning Algorithm	167
6.2.2 Experimental Results and Comparison	174
6.3 Chapter Summary and Conclusions	175
7. Conclusions and Future Work	177
7.1 Thesis Contributions	177
7.2 Future Work	179
7.2.1 Synthesis Based on Physical Constraints	179
7.2.2 Control-Path Design and Synthesis	182
References	187
Biography	193

List of Tables

Table 2.1: Experimentally-characterized module library for synthesis.	32
Table 2.2: Defect tolerance for defect-oblivious and defect-tolerant method for the protein assay.	40
Table 3.1: Time-span table with detailed scheduling results for the overlapping region.	51
Table 3.2: Bioassay schedule for a full-addressable array.	56
Table 3.3: Random synthetic benchmarks, sample size = 1000.	72
Table 3.4: Droplet manipulation results for grouping-based method without scheduling of droplet routes.	76
Table 3.5: Droplet manipulation results for grouping-based method with scheduling of droplet routes.	77
Table 3.6: A fragment of the activation sequences for multiplexed assay.	86
Table 4.1: List of catastrophic defects for biochips.	95
Table 4.2: Fault models for digital microfluidic biochips.	118
Table 6.1: Target solution list for protein crystallization.	170
Table 6.2: Stock solutions needed to prepare the target solutions in Table 6.1.	171
Table 6.3: Target solutions containing reagent polyethylene glycol 4000.	172
Table 6.4: Preparation plan for target solution MembFac_02.	172

List of Figures

Figure 1.1: Fabricated digital microfluidic arrays: (a) glass substrate [23]; (b) PCB substrate [21]	8
Figure 1.2: An example illustrating system-level synthesis [15].	10
Figure 1.3: A cross-referencing microfluidic device that uses single-layer driving electrodes on both top and bottom plates (adapted from [39]).	13
Figure 1.4: Lysozyme crystals obtained on-chip at 5× and 20× magnifications.	15
Figure 1.5: An overview of thesis research.	17
Figure 2.1: An example of a non-routable interdependent pair.	23
Figure 2.2: Illustration of module distance.	23
Figure 2.3: Pseudocode for the PRSA-based droplet-routing-aware synthesis procedure	25
Figure 2.4: Example of partial reconfiguration.	25
Figure 2.5: Sequencing graph for a protein assay.	32
Figure 2.6: A 3-D model illustrating the synthesis results: (a) routing-oblivious method of [15]; (b) the proposed method.	32
Figure 2.7: (a) A snapshot of a non-routable layout from routing-oblivious synthesis (time instant 297 s) (b) Corresponding layout in routing-aware synthesis (time instant 299 s)..	35
Figure 2.8: Feasibility frontier and feasible design region for [15] and the routing-aware synthesis method.	36
Figure 2.9: Assay completion times (with droplet transportation time included) for [15] and for the proposed routing-aware synthesis method.	36
Figure 2.10: (a) A defective 10×10 microfluidic array (b) Reconfiguration results for post-synthesis defect tolerance.	38
Figure 2.11: Feasibility frontier surface and feasible design region for defect-tolerant and defect-oblivious routing-aware synthesis methods.	41
Figure 3.1: An example to illustrate droplet interference due to the sharing of control pins by	

the electrodes: (a) coordinate locations for the electrodes; (b) pin-assignment for the electrodes.	43
Figure 3.2: An example of an inadvertent operation for a single droplet.	43
Figure 3.3: A 5-pin layout for a 4×4 array.	43
Figure 3.4: (a) Detectors used in bioassay; (b) Routing result and array partitions.	49
Figure 3.5: (a) Routing result and partitioning (b) Time-span table for the droplets.	50
Figure 3.6: Mapping of an array to an undirected graph.	53
Figure 3.7: A single Bagua structure (the tilted square) and its repetition in a square partition.	53
Figure 3.8: Covering a partition by shifting Bagua repetition along rows.	53
Figure 3.9: A demonstration that the “cross constraints” are met.	55
Figure 3.10: A wiring example for the pin assignment obtained using the Connect-5 algorithm. For each partition, two pins can be wired in one layer.	55
Figure 3.11: A 15×15 array used for multiplexed bioassays.	56
Figure 3.12: Partition and pin assignment result of the multiplex bioassay. Blank areas are don’t care regions that can be either left unaddressed or combined with any partition.	56
Figure 3.13: An example to illustrate the problem of electrode interference. H/L stands for high/low voltage pairs to activate the cells, and unselected row/column pins are left floating (F)	60
Figure 3.14: An example of electrode interference within the same row	60
Figure 3.15: Example to illustrate destination-cell-based categorization	62
Figure 3.16: Example to illustrate the concurrent movement of a group of droplets.	62
Figure 3.17: Example of electrode interference due to asynchronous processing of multiple droplet manipulations.	62
Figure 3.18: Mapping destination cell layout to undirected graph.	64
Figure 3.19: An example of a droplet-routing plan.	66

Figure 3.20: Schedule A: (a) Snapshot 1; (b) Snapshot 2; (c) Snapshot 3; (d) Activation sequence (in terms of rows and columns)	67
Figure 3.21: Schedule B: (a) Snapshot 1; (b) Snapshot 2; (c) Snapshot 3. (d) Activation sequence (in terms of rows and columns).	67
Figure 3.22: Pseudo-code of scheduling algorithm for droplet routes.	68
Figure 3.23: A routing example to illustrate scheduling algorithm.	68
Figure 3.24: Sequencing graph model of bioassay example.	73
Figure 3.25: Schedule obtained via architectural-level synthesis.	73
Figure 3.26: Module placement (11 sub-problems) for the multiplexed bioassay.	74
Figure 3.27: Routing plan for Sub-problem 3.	76
Figure 3.28: Comparison of droplet-routing times for 4 sub-problems (#1, #2, #3, #5)	78
Figure 3.29: Comparison of droplet-routing times for the power-efficient-method, the power-oblivious method, and the method from [63].	78
Figure 3.30: Comparison of average # of control pins activated per manipulation step for the power-efficient method, the power-oblivious method, and the method from [63].	78
Figure 3.31: Illustration of a “don’t-care” in electrode activation.	81
Figure 3.32: Example of activation sequence calculation (a) routing and layout information (b) calculated activation sequences.	81
Figure 3.33: Mapping of the activation sequences of Figure 3.32 to an undirected graph.	82
Figure 3.34: Sequencing graph model for the multiplexed bioassay. S_1, S_2 are samples, R_1, R_2 are reagents, $M_1 \sim M_4$ are mixing operations, and $D_1 \sim D_4$ are detection operations	85
Figure 3.35: Broadcast addressing for the multiplexed assay chip.	86
Figure 3.36: Comparison of assay completion time.	86
Figure 3.37: Sequencing graph for the mixing stage of PCR.	88

Figure 3.38: Mapping of the PCR assay on a 15×15 array.	88
Figure 3.39: Schedule for the PCR assay.	88
Figure 3.40: Broadcast addressing for the PCR chip.	89
Figure 3.41: Layout for protein-dilution chip.	90
Figure 3.42: Schedule for the protein dilution assay, Dlt –dilution, Mix – mixing, Opt – optical detection.	90
Figure 3.43: Broadcast-addressing for the protein-dilution chip.	90
Figure 3.44: Pin-assignment layout for multi-functional chip.	91
Figure 4.1: Illustration of single droplet scan-like test using a single droplet.	95
Figure 4.2: Capacitive sensing circuit: (a) outline [67]; (b) detail circuit design; (c) experimental setup.	96
Figure 4.3: Example of target regions and pseudo-sources.	98
Figure 4.4: Example defect-free test outcome for (a) single-droplet scan-like method, and (b) parallel scan-like method.	99
Figure 4.5: Step I (Peripheral testing).	100
Figure 4.6: Step II: (a) parallel scan-like test for even columns; (b) routing of test droplets to sink (c) test droplet routed to odd columns; (d) parallel scan-like test for odd columns; (e) routing of test droplets to sink.	100
Figure 4.7: Step III: (a) parallel scan-like test for odd rows; (b) routing of test droplets to sink (c) test droplet routed to even rows; (d) parallel scan-like test for even rows; (e) routing of test droplets to sink.	101
Figure 4.8: Example of test outcome for a faulty array.	103
Figure 4.9: Diagnosis of a single-electrode defect by “cross-parallel” scan-like test.	103
Figure 4.10: Diagnosis of an electrode-short defect using parallel scan-like test.	104
Figure 4.11: On-line parallel scan-like test for a 6x6 microfluidic array. Shaded cells correspond to modules in use for bioassay operations	104

Figure 4.12: An example of incorrectly-classified defects	107
Figure 4.13: Elimination of incorrectly-classified defects using binary-search-based diagnosis.	107
Figure 4.14: An example of an untestable site.	108
Figure 4.15: Illustration of diagonal parallel scan-like test.	108
Figure 4.16: Comparison of (a) testing complexity, and (b) diagnosis complexity (additional steps) of parallel scan-like test and Euler-circuit based method.	110
Figure 4.17: Illustration of a candidate defect (incorrectly classified).	111
Figure 4.18: Illustration of the conditions that lead to incorrect classification of candidate defects for (a) single-diagonal diagnosis, (b) cross-diagonal diagnosis.	112
Figure 4.19: Simulation results highlighting the likelihood of incorrect classification. .112	
Figure 4.20: Illustration of untestable sites for (a) single-diagonal diagnosis (b) cross-diagonal diagnosis.	114
Figure 4.21: Simulation results highlighting the probability of untestable sites.	114
Figure 4.22: Fabricated biochip for DNA sequencing.	115
Figure 4.23: Column-test step of parallel scan-like test.	115
Figure 4.24: Parallel scan-like diagnosis of single cell defect.	115
Figure 4.25: Illustration of (a) normal dispensing and (b) dispensing failure, for a fabricated microfluidic biochip.	120
Figure 4.27: Test readouts for (a) normal dispensing and (b) dispensing failure.	120
Figure 4.28: (a) An undirected graph for Euler-path-based structural test; (b) The corresponding directed graph for Euler-path-based routing test.	121
Figure 4.29: Test outcomes for the capacitive sensing circuit.	121
Figure 4.30: (a) Pivots and (b) routing path for a 2x4 microfluidic mixer.	121
Figure 4.31: Example of merging test.	122
Figure 4.32: Mixing and splitting test for all the groups of three electrodes that are centered	

on a given electrode.	124
Figure 4.33: Parallel mixing and splitting test for a row of electrodes.	125
Figure 4.34: Parallel mixing and splitting test for a row of electrodes.	126
Figure 4.35: Functional test on an n-phase-bus chip.	127
Figure 4.36: Functional test on cross-referencing-based chip.	129
Figure 4.37: Functional test on array-partitioning-based chip.	129
Figure 4.38: Functional test for broadcast-addressing-based chip (a) layout and droplet routes for the multiplexed-assay chip (b) broadcast-addressing-based pin-assignment result.	130
Figure 4.39: Fabricated biochip for PCR.	132
Figure 4.40: Mixing and splitting test for a fabricated PCR chip (bottom row of Figure 4.39)	132
Figure 4.41: Failure rate for synthesized design without testing and with structural testing.	134
Figure 4.42: Bioassay completion time for synthesized design without test, with structural test, and with functional test for defect occurrence probability of (a) $P[A] = 0.01$. (b) $P[A] = 0.05$	135
Figure 4.43: Pin-constrained chip designs for functional test evaluation: (a) a linear 5-phase-bus chip for dilution; (b) layout and droplet routes for the multiplexed-assay chip; (c) array-partitioning-based array for multiplexed assay; (d) cross-referencing-based array for multiplexed assay	136
Figure 4.44: No. of droplet-manipulation steps for functional test of direct-addressable and pin-constrained chips	137
Figure 4.45: Functional testability of pin-constrained chips	138
Figure 5.1: A fabricated digital microfluidic biochip [24].	142
Figure 5.2: An example of an untestable functional unit on a pin-constrained chip for multiplexed assay.	143
Figure 5.3: Illustration of the influence by adding test operations to the bioassay.	145

Figure 5.4: Mixing and splitting test for a functional unit.	148
Figure 5.5: Parallel mixing and splitting test for a row of electrodes.	148
Figure 5.6: (a) Mixing-and-splitting test of functional units along the Euler path of a chip (b) Testing functional units in groups of two.	148
Figure 5.7: Comparison of test completion time for the Euler-path-based functional test method.	150
Figure 5.8: Pin assignment for the multiplexed assay chip obtained using the testability-aware design method.	150
Figure 5.9: Comparison of test completion time for the Euler-path-based functional test method.	152
Figure 5.10: Pin-assignment result for the PCR assay chip obtained using the test-aware design method.	152
Figure 6.1: Schematic view of a 96-well chip that automatically sets up 96 reagent condition solutions.	155
Figure 6.2: Schematic top-view of four wells and the surrounding electrodes.	155
Figure 6.3: Illustration of wire routing limits on a PCB layer.	157
Figure 6.4: Example of pin-assignment example for a 4-well-plate design.	158
Figure 6.5: Illustration of a 6×6 electrode well unit.	159
Figure 6.6: Pin assignment using 5 pins for the 96-well chip (unit well size = 6×6 electrodes)	159
Figure 6.7: Wiring of a well unit.	160
Figure 6.8: Wiring of a well unit with large vias.	160
Figure 6.9: Loading of three droplets using shuttle-passenger-like method.	162
Figure 6.10: Activation sequence and dispensing time instances for routing droplets to corresponding starting point in Figure 6.9.	162
Figure 6.11: Three categories of defects.	164
Figure 6.12: Illustration of “cross loading” method (a) loading in column direction. (b) loading	

in row direction.	164
Figure 6.13: Critical path for the multi-well chip (for both the direct-access and pin-constrained chips)	166
Figure 6.14: Evaluation of failure rates for pin-constrained chip and independently controlled chip.	166
Figure 6.15: Concentration error limit vs number of mixing-dispensing iterations.	175
Figure 7.1: An example of mismatch problem – reservoir capacity overflow.	180
Figure 7.2: An example of scheduling error (<i>Inc.</i> refers to incubation operation).	181
Figure 7.3: Backtrace algorithm and re-execution subroutine.	185
Figure 7.4: The incorporation of a checkpoint to a sequencing graph (<i>Det.</i> refers to detection operation).	185

Acknowledgements

I would like to acknowledge those who have contributed to the completion of this thesis. First, I would like to thank my advisor, Prof. Krishnendu Chakrabarty for his continued support, guidance, and advice throughout this work. I would also like to thank my industry mentor, Dr. Vamsee K. Pamula from Advanced Liquid Logic. Secondly, I wish to thank the other members of my committee, Prof. Richard B. Fair, Prof. Lingchong You, and Prof. Chris Dwyer. Third, I give thanks to my director at Cisco Systems, Dr. Xinli Gu, and my manager, Mr. Young Lee. A special thanks goes to my family for their encouragement and to Tong Wang for all her loving support. Financial support received from National Science Foundation is greatly appreciated.

Chapter 1

Introduction

Microfluidics-based biochips, also referred to as lab-on-a-chip, are revolutionizing many areas of biochemistry and biomedical sciences. Typical applications include enzymatic analysis (e.g. lactate assays), DNA sequencing, immunoassays, proteomic analysis, blood chemistry for clinical diagnostics, and environmental toxicity monitoring [1,2,3]. These devices enable the precise control of microliter and nanoliter volumes of biological samples. They combine electronics with biology, and they integrate various bioassay operations, such as sample preparation, analysis, separation, and detection [1,4]. Compared to conventional laboratory experiment procedures, which are usually cumbersome and expensive, these miniaturized and automated biochip devices offer a number of advantages such as higher sensitivity, lower cost due to smaller sample and reagent volumes, higher levels of system integration, and less likelihood of human error.

A popular class of microfluidic biochips is based on continuous fluid flow in permanently-etched microchannels. These devices rely on either micropumps and microvalves; or electrical methods such as electrokinetics to control continuous fluidic flow [4,5]. Some recent continuous-flow biochip products include the Topaz™ system for protein crystallization from Fluidigm Corporation, the LabChip system from Caliper Life Sciences, and the LabCD™ system from Tecan Systems [6,7,8].

An alternative category of microfluidic biochips relies on “digital microfluidics”, which is based on the principle of electrowetting-on-dielectric [9,10,11,12]. Since discrete droplets of nanoliter volumes can be manipulated using a patterned array of electrodes, miniaturized bioassay protocols (in terms of liquid volumes and assay times) can be mapped and executed on a microfluidic chip. Therefore, digital microfluidic biochips require only nanoliter

volumes of samples and reagents. They offer continuous sampling and analysis capabilities for on-line and real-time chemical/biological sensing [13]. These systems also have a desirable property referred to as dynamic reconfigurability, whereby microfluidic modules can be relocated to other places on the electrode array, without affecting functionality, during the concurrent execution of a set of bioassays. Reconfigurability enables microfluidic biochips to be “adaptive” to a wide variety of applications. System reconfiguration can also be used to bypass faulty cells to enable microfluidic arrays to provide reliable assay outcomes in the presence of defects.

Recent years have seen growing interest in automated chip design and optimized mapping of multiple bioassays for concurrent execution on a digital microfluidic platform [14,15,16,17,18]. Therefore, system complexity and integration levels are likely to increase as chips are designed and manufactured for emerging applications. Time-to-market and fault tolerance are also expected to emerge as design considerations. Therefore, there is a need to deliver the same level of design-automation support to the biochip designers and users that the semiconductor industry takes for granted.

As in the case of integrated circuits (ICs), an increase in the density and area of microfluidics-based biochips will also lead to high defect densities, thereby reducing yield, especially for newer technologies and manufacturing process. However, dependability is an important system attribute for biochips. Dependability is essential for safety-critical applications such as point-of care diagnostics, health assessment and screening for infectious diseases, air-quality monitoring, and food-safety tests, as well as for pharmacological procedures for drug design and discovery that require high precision levels. Therefore these chips must be tested adequately not only after fabrication, but also continuously during in-field operation. Due to the underlying mixed-technology and multiple-energy domains, microfluidic biochips exhibit unique failure mechanisms and defects. In fact, the ITRS 2003 document recognized the need for new test methods for heterogeneous device technologies

that underly microelectromechanical systems and sensors, and highlighted it as one of the five difficult test challenges beyond 2009 [19].

The increase in the system complexity and integration levels poses additional challenges for electrode addressing and system control. Most prior work on biochips computer-aided-design (CAD) has assumed a direct-addressing scheme, where each electrode is connected to a dedicated control pin; it can therefore be activated independently. This method provides the maximum freedom for droplet manipulation, but it requires an excessive number of control pins. For example, a total of 10^4 pins are needed to independently control the electrodes in a 100×100 array. Multi-layer electrical connection structures and wire-routing solutions are complicated by the large number of independent control pins in such arrays. Product cost, however, is a major marketability driver due to the one-time-use (disposable) nature of most emerging devices. Thus, the design of pin-constrained digital microfluidic arrays is of considerable importance for the emerging marketplace.

Some of the above issues, especially related to synthesis and testing, have been addressed in [20], which presented the first design automation framework for digital microfluidics. A number of integrated design-automation tools were presented in [20] for chip design and for the chip user. These tools target design optimization, ease of use, as well as chip testing and system maintenance, thereby allowing biochip users to focus on target applications and assay adaptation. However, the design methods in [20] are often based on unrealistic assumptions. Many practical issues such as physical and technology-related constraints, the nature of manufacturing defects, and fabrication cost, are not taken into account. As a result, the chip designs resulting from these methods are often impractical. For example, the synthesis tool in [20] focuses only on compact designs and it is prone to generate synthesis results with no feasible droplet routing pathways. Moreover, most designs resulting from [20] leads to a large number of control pins that require expensive multi-layer PCB technology. Furthermore, the test methods in [20] do not address many realistic defects. As a result, the design tools

presented in [20] are only of a conceptual nature and they cannot be directly used in practice for chip design in practice. Finally, since testability is ignored during chip design in [20], the test methods described in [20] are not always effective for fabricated biochips.

This thesis is focused on application-guided design automation tools that address practical issues such as defects, routability and fabrication cost. The goal is to provide the means for the automated design and use of robust, low-cost, and manufacturable digital microfluidic systems. A unified synthesis tool that incorporates defect-tolerance and droplet-routing has been developed. Effective metrics have been introduced and used to estimate the complexity of routing and system robustness of chip designs. Based on estimation results, the unified synthesis tool uses a parallel-recombinative simulated-annealing (PRSA) algorithm to search for robust and easily-routable chip designs in the candidate design space.

To reduce fabrication cost, pin-constrained design methods has been developed to reduce the number of control pins in microfluidic arrays. The first method is based on droplet-trace-based array partitioning. It uses the concept of “droplet trace”, which is extracted from the scheduling and droplet routing results produced by the synthesis tool. An efficient pin-assignment method, referred to as the “Connect-5 algorithm”, is combined with the array-partitioning technique to address electrode arrays with a limited number of control pins. A second pin-constrained design method is targeted towards a “cross-referencing” chip, which allows the control of an $N \times M$ grid array with only $N+M$ control pins. An efficient droplet manipulation method has been proposed to achieve high throughput on such cross-referencing chips. Finally, a broadcast-addressing-based design method has also been proposed to reduce the number of control pins. This method relies on the grouping of electrodes with compatible actuation sequences, and the addressing of these electrodes using a single control pin.

This thesis also includes fault models for digital microfluidics based on observed defects

in fabricated chips. A parallel scan-like method has been proposed for efficient structural test of digital microfluidic arrays. This method relies on concurrent manipulation of multiple test droplets to traverse the target array. A comprehensive functional test method has been developed to verify the correct operation of functional units. The proposed method provides functional test techniques to address fundamental biochip operations such as droplet dispensing, droplet transportation, mixing, splitting, and capacitive sensing. For each operation, functional testing is carried out using parallel droplet pathways, and it leads to qualified regions where synthesis tools can map the corresponding microfluidic functional modules.

The proposed test methods facilitate defect screening, which is necessary to ensure dependable system operation. However, the effectiveness of these test techniques is limited by the fact that they do not consider testability. To address this problem, design-for-testability (DFT) techniques have been proposed in this thesis for digital microfluidic biochips. A DFT method has been presented to incorporate a test plan into the fluidic operations of a target bioassay protocol. By using the testability-aware bioassay protocol as an input to the biochip design tool, the proposed DFT method ensures a high level of testability. An Euler-path-based functional test method, which allows functional testing for irregular chip layouts, has also been presented.

The above design automation and testing tools have been utilized to design microfluidic biochips for protein crystallization, an important laboratory technique for understanding the structure of proteins. A multi-well high-throughput biochip chip design for protein crystallization has been proposed. The chip design has been optimized using the proposed Connect-5 pin-constrained design method, which achieves a significant reduction of input bandwidth without loss, thereby reducing the fabrication cost. With the help of an efficient well-loading algorithm for parallel manipulation of multiple droplets, the optimized pin-constrained design maintains the same level of operation concurrency as a

direct-addressed design. Finally, defect tolerance techniques have been presented to ensure the functionality of the chip under the condition of defects. The above design automation and optimization tools help deliver an efficient, cost-effective and reliable design of a biochip platform for protein crystallization, which is ready for manufacture, as well as easy to use and maintain after it is fabricated.

The rest of this chapter is organized as follows. Section 1.1 presents an overview of digital microfluidic technology. Section 1.2 discusses synthesis, testing and pin-constrained design techniques. Section 1.3 presents an overview of protein crystallization and design automation tools for protein crystallization chip design. Finally, an outline of the thesis is presented in Section 1.4.

1.1 Digital Microfluidic Technology

Traditional microfluidic technologies are based on the continuous flow of liquid through etched microchannels on a glass or plastic substrate [4, 21]. Pumping is performed either by external pressure sources, integrated mechanical micropumps, or by electrokinetic mechanisms. These systems are often operated in a serial mode where samples and reagents are loaded into one end, and then moved together towards an output at the other end with mixing, sample injection, and separations occurring at (structurally) predetermined points along the path. These systems are adequate for many well-defined and simple applications, but are unsuited for more complex tasks requiring a high degree of flexibility or complicated fluid manipulations. Continuous-flow systems are inherently difficult to integrate because the parameters that govern flow field (e.g. pressure, fluid resistance, electric field strength) vary along the flow-path making the flow at any one location dependent upon the properties of the entire system. As liquids mix and react in the system, their electrical and hydrodynamic properties change, resulting in even more complicated behavior. Consequently, the design and analysis of even moderately complex systems can be very challenging. Furthermore,

since structure and function are so tightly coupled, each system is only appropriate for a narrow class of applications.

A digital microfluidic biochip utilizes the phenomenon of electrowetting to manipulate and move microliter or nanoliter droplets containing biological samples on a two-dimensional electrode array [22]. A unit cell in the array includes a pair of electrodes that acts as two parallel plates. The bottom plate contains a patterned array of individually controlled electrodes, and the top plate is coated with a continuous ground electrode. A droplet rests on a hydrophobic surface over an electrode, as shown in Figure 1.1. It is moved by applying a control voltage to an electrode adjacent to the droplet and, at the same time, deactivating the electrode just under the droplet. This electronic method of wettability control creates interfacial tension gradients that move the droplets to the charged electrode. Using the electrowetting phenomenon, droplets can be moved to any location on a two-dimensional array.

The division of a volume of fluid into discrete, independently-controllable packets or droplets for manipulation, provides several important advantages over continuous-flow. The reduction of microfluidics to a set of basic repeated operations (i.e., “move one unit of fluid one distance unit”) allows a hierarchical and cell-based design approach to be utilized. Large systems may be constructed out of repeated instances of a single well-characterized device in the same way that complex microelectronic circuits may be built upon a single well characterized transistor. Thus, the design and analysis of arbitrarily complex microfluidic systems becomes tractable. The constituent cells may be reorganized at different hierarchical levels, either through hardware or software, to provide new functionality on demand.

By varying the patterns of control voltage activation, many fluid-handling operations such as droplet merging, splitting, mixing, and dispensing can be executed in a similar manner. For example, mixing can be performed by routing two droplets to the same location and then turning them about some pivot points. The digital microfluidic platform offers the

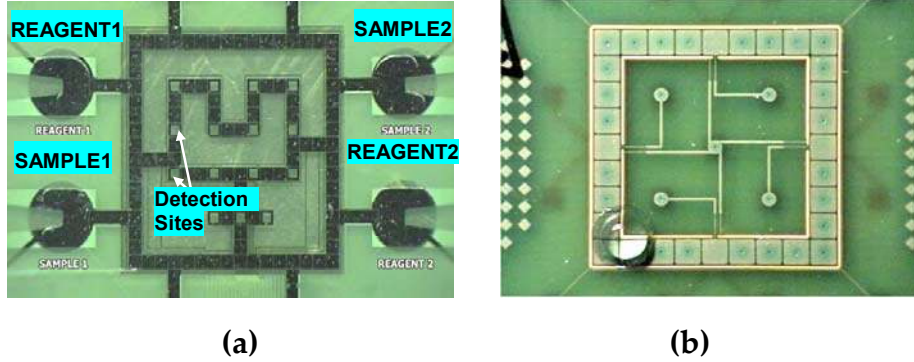


Figure 1.1: Fabricated digital microfluidic arrays: (a) glass substrate [23]; (b) PCB substrate [21].

additional advantage of flexibility, referred to as reconfigurability, since fluidic operations can be performed anywhere on the array. Droplet routes and operation scheduling results are programmed into a microcontroller that drives electrodes in the array. In addition to electrodes, optical detectors such as LEDs and photodiodes are also integrated in digital microfluidic arrays to monitor colorimetric bioassays [23].

To address the need for low-cost, PCB technology has been employed recently to inexpensively mass-fabricate digital microfluidic biochips. Using a copper layer for the electrodes, solder mask as the insulator, and a Teflon AF coating for hydrophobicity, the microfluidic array platform can be fabricated by using an existing PCB manufacturing process [25]. This inexpensive manufacture technique allows us to build disposable PCB-based microfluidic biochips that can be easily plugged into a controller circuit board that can be programmed and powered via a standard USB port. However, multiple metal layers for PCB design for large scale microfluidic biochips may lead to reliability problems and increase fabrication cost. Thus, reducing the number of independent control pins is important for successful commercialization. We can also address individual electrodes separately by employing a serial-to-parallel interface. However, this requires active circuit components on the PCB, e.g., logic elements such as gates and flip-flops, which will lead to increased cost and power consumption.

1.2 Synthesis, Testing, and Pin-Constrained Design Techniques

Recent years have seen growing interest in the automated design and synthesis of microfluidic biochips [14, 15, 16, 17, 18]. One of the first published methods for biochip synthesis decouples high-level synthesis from physical design [11]. It is based on rough estimates for placement costs such as the area of the microfluidic modules. These estimates provide lower bounds on the exact biochip area, since the overheads due to spare cells and cells used for droplet transportation are not known a priori. However, it cannot be accurately predicted if the biochip design meets system specifications, e.g., maximum allowable array area and upper limits on assay completion times, until both high-level synthesis and physical design are carried out. [15] proposed a unified system-level synthesis method for microfluidic biochips based on parallel recombinative simulated annealing (PRSA), which offers a link between these two steps. This method allows users to describe bioassays at a high level of abstraction, and it automatically maps behavioral descriptions to the underlying microfluidic array.

The design flow is illustrated in Figure 1.2. First, the different bioassay operations (e.g. mixing and dilution), and their mutual dependences are represented using a sequencing graph. Next, a combination of simulated annealing and genetic algorithms are used for unified resource binding, operation scheduling, and module placement. A chromosome is used to represent each candidate solution, i.e., a design point. In each chromosome, operations are randomly bound to resources. Based on the binding results, list scheduling is used to determine the start times of operations, i.e., each operation starts with a random latency after its scheduled time. Finally, a module placement is derived based on the resource binding and the schedule of fluidic operations. A weighted sum of area- and time-cost is used to evaluate the quality of the design. The design is improved through a series of genetic evolutions based on PRSA. It generates an optimized schedule of bioassay operations, the binding of assay

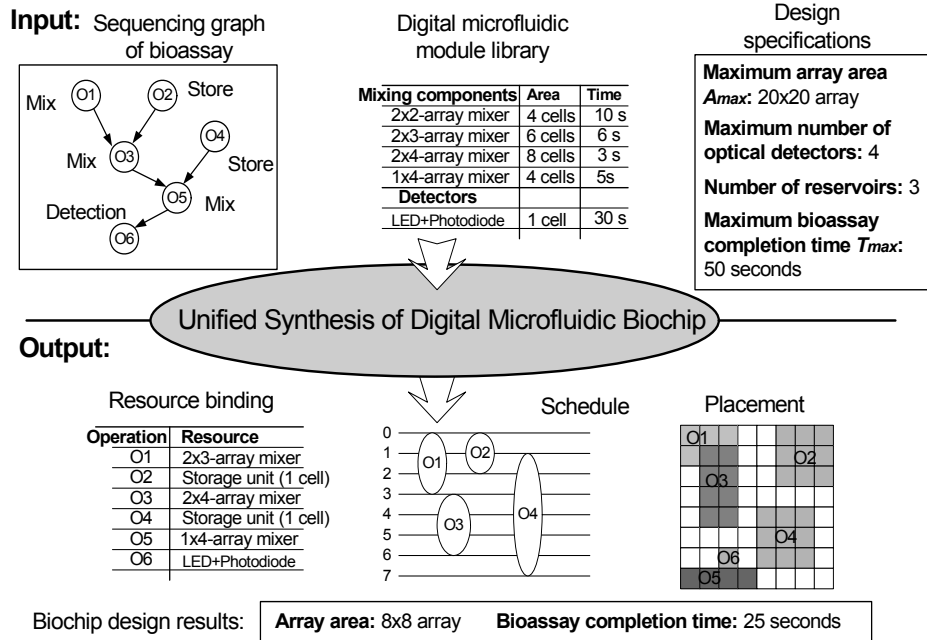


Figure 1.2: An example illustrating system-level synthesis [15].

operations to resources, and a layout of the microfluidic biochip.

The top-down synthesis flow described above unifies architecture level design with physical-level module placement. However, it suffers from two drawbacks. For operation scheduling, it is assumed that the time cost for droplet routing is negligible, which implies that droplet routing has no influence on the operation completion time. While generating physical layouts, the synthesis tool in [15] provides only the layouts of the modules and it leaves droplet routing pathways unspecified. The assumption of negligible droplet transportation times is valid for small microfluidic arrays. However, for large arrays and for biochemical protocols that require several concurrent fluidic operations on-chip, the droplet transportation time is significant and routing complexity is non-trivial.

Recent work on automated biochip design has also included post-synthesis droplet routing [26,27]. These methods can reduce droplet transportation time by finding optimal routing plans for a synthesized biochip. However, the effectiveness of such methods is limited by the synthesis results, i.e., the placement of microfluidic modules often determines the droplet pathways that lead to minimum droplet transportation time. For example, if we

need to route a droplet between two modules that are 10 electrodes away from each other, then it is not possible to reduce the droplet transportation time to less than that needed to move a droplet by a distance equal to 10 electrodes. Since droplet pathways are dynamically reconfigurable, the number of feasible droplet pathways can be very high, leading to considerable computation time for a droplet routing tool.

The testing of microfluidic biochips has recently been investigated [28,29,30]. These test methods add fluid handling aspects to MEMS testing techniques [31]. Test methods have been proposed for both continuous-flow and digital microfluidic biochips. An excellent review is available in [32]. A fault model and a fault simulation method for continuous-flow microfluidic biochips have been proposed in [33]. For digital microfluidic chips, techniques for defect classification, test planning, and test resource optimization have been presented [28]. Defect classification methods are discussed in [28] and corresponding test procedures are described in [29]. Defects have been classified as being either catastrophic or parametric, and techniques have been developed to detect these defects by electrostatically controlling and tracking droplet motion.

The work in [28, 29] facilitates concurrent testing, which allows fault detection and biomedical assays to run simultaneously on a microfluidic system. A drawback of [28] however is that it does not present any automated techniques for optimizing the test application procedure. [34] first proposed a test planning and test resource optimization method. The test planning problem is mapped to the Hamilton cycle problem from graph theory. An alternative method based on Euler paths is proposed in [36]. This method maps a digital microfluidic biochip to an undirected graph and a test droplet is routed along the Euler path derived from the graph to pass through all the cells in the array. Fault diagnosis is carried out using multiple test application steps and adaptive Euler paths.

Another important issue in biochip design is electrode addressing, i.e., the manner in which electrodes are connected to and controlled by input pins. Early design-automation

techniques relied on the availability of a direct-addressing scheme. For large arrays, direct-addressing schemes leads to a large number of control pins, and the associated interconnect routing problem significantly adds to the product cost. Thus, the design of pin-constrained digital microfluidic arrays is of great practical importance for the emerging marketplace.

Pin-constrained design of digital microfluidic biochips was recently proposed in [37]. This method uses array partitioning and careful pin assignment to reduce the number of control pins. However, it requires detailed information about the scheduling of assay operations, microfluidic module placement, and droplet routing pathways. Thus, the array design in such cases is specific to a target biofluidic application. In another method proposed in [38], the number of control pins for a fabricated electrowetting-based biochip is minimized by using a multi-phase bus for the fluidic pathways. Every n th electrode in an n -phase bus is electrically connected, where n is small number (typically $n = 4$). Thus, only n control pins are needed for a transport bus, irrespective of the number of electrodes that it contains. Although the multi-phase bus method is useful for reducing the number of control pins, it is only applicable to a one-dimensional (linear) array.

An alternative method based on a cross-reference driving scheme is presented in [39]. This method allows control of an $N \times M$ grid array with only $N+M$ control pins. The electrode rows are patterned on both the top and bottom plates, and placed orthogonally. In order to drive a droplet along the X-direction, electrode rows on the bottom plate serve as driving electrodes, while electrode rows on the top serve as reference ground electrodes. The roles are reversed for movement along the Y-direction, as shown in Figure 1.3. This cross-reference method facilitates the reduction of control pins. However, due to electrode interference, this design cannot handle the simultaneous movement of more than two droplets. The resulting serialization of droplet movement is a serious drawback for high-throughput applications.

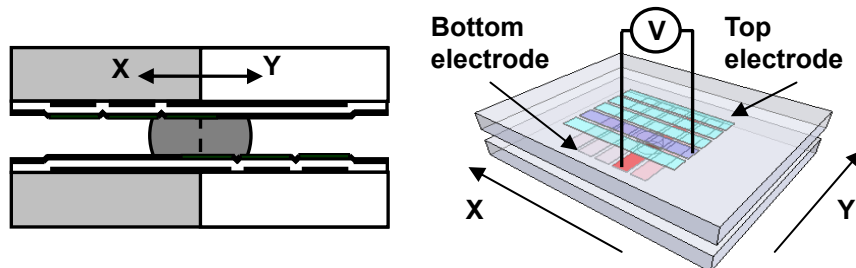


Figure 1.3: A cross-referencing microfluidic device that uses single-layer driving electrodes on both top and bottom plates (adapted from [39]).

The minimization of the assay completion time, i.e., the maximization of throughput, is essential for environmental monitoring applications where sensors can provide early warning. Real-time response is also necessary for surgery and neo-natal clinical diagnostics. Finally, biological samples are sensitive to the environment and to temperature variations, and it is difficult to maintain an optimal clinical or laboratory environment on chip. To ensure the integrity of assay results, it is therefore desirable to minimize the time that samples spend on-chip before assay results are obtained. Increased throughput also improves operational reliability. Long assay durations imply that high actuation voltages need to be maintained on some electrodes, which accelerate insulator degradation and dielectric breakdown, reducing the number of assays that can be performed on a chip during its lifetime. The need to minimize time-to-result and the emphasis on small footprint (because of cost consideration) are the main drivers for the optimization research undertaken in this thesis.

1.3 Protein Crystallization

This thesis considers protein crystallization as a target application for optimized chip design. Proteins play a key role in all biological processes. The specific biological function of a protein is determined by the three-dimensional (3D) arrangement of the constituent amino acids. Therefore, their structure needs to be understood for effective protein engineering, bioseparations, rational drug design, controlled drug delivery, as well as for design of novel enzyme substrates, activators, and inhibitors. A widely used method to study the 3D structure

of proteins is to crystallize the proteins and determine the structure using X-ray diffraction [40].

Studies have been reported in the literature to gain a fundamental understanding of the mechanism of crystallization [41], but, owing to the complexity and the number of parameters involved in the problem, it may take years before the process is understood well enough to have practical value. However, structural biologists need immediate information about the structure of proteins, hence empirical methods are widely employed for crystallization. For example, an empirical approach that is typically used, among others, is a two-dimensional coarse sampling that involves systematic variation of salt concentration versus pH [41].

Protein crystallization is a multi-parametric process that involves the steps of nucleation and growth, where molecules are brought into a thermodynamically unstable and a supersaturated state. In order to “hit” upon the correct parameters for the crystallization of proteins, typically a very large number of experiments (10^3 to 10^4) are required, which leads to the consumption of large protein volumes.

Efforts are underway to reduce the consumption of proteins by miniaturizing the crystallization setup. Screening for protein crystallization includes many repetitive and reproducible pipetting operations. To ease this manual and time-consuming task, several automatic methods have been introduced. In 1990, Chayen et al introduced a micro-batch method where only 1 μL of protein and 1 μL of precipitants were dispensed by programmed Hamilton syringes [42] in each well of a 96-well plate containing paraffin oil. Microbatch crystallization has been recently demonstrated in micropipettes in 1 μL droplets by DeTitta’s group at Hauptman Woodward Institute (HWI), where the precipitant and the protein solutions are loaded manually into a microcentrifuge tube, centrifuged, collected in a micropipette, and then sealed [43]. Despite the efforts at reducing the protein volumes, these processes still consume a significant amount of protein and they are labor-intensive.

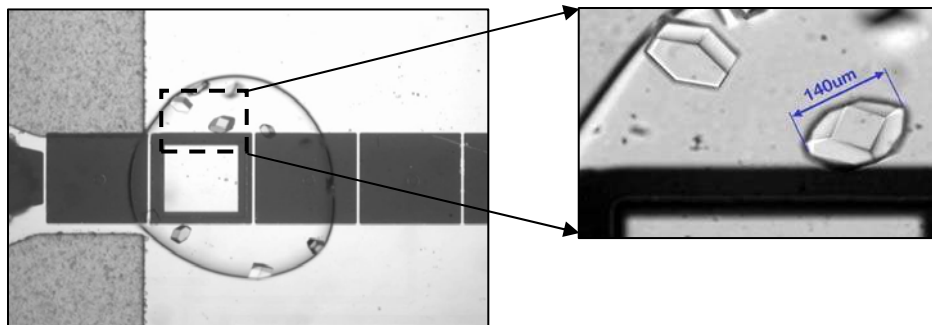


Figure 1.4: Lysozyme crystals obtained on-chip at 5× and 20× magnifications [24].

Robotic automation has emerged as the dominant paradigm in state-of-the-art high-throughput protein crystallization. However, robots are slow, very expensive, and they require high maintenance. Currently, there are only a few automatic crystallization systems that are commercially available. Douglas Instruments' Oryx 8 [44] can perform both microbatch and vapor diffusion methods on protein samples in the range of 0.1–2 μL . Gilson's robotic workstations [45] can also perform both microbatch and vapor diffusion on protein samples about 1 μL . Syrrx, a rational drug design company, manufactures a robotic system [46] for protein crystallization utilizing 20 nL to 1 μL protein samples. State-of-the-art robotic systems at HWI's NIH-funded Center for High Throughput Crystallization have a throughput of 69,000 experiments per day for setting up microbatch crystallization conditions, that is, a 96 well-plate could be setup every 2 minutes. Each screening condition still requires 0.4 μL of protein. These semi-automatic systems do not encompass ideal high-throughput configurations, requiring user intervention for multiple tray processing, as well as suffering from other material processing issues. As most of the work performed with these systems is not on a large scale, the automation of storage and handling of plates was not addressed in these systems [47]. Such industrial systems, even though they are capable of setting up thousands of crystallization screens a day, are prohibitively expensive for academic research labs [48]. Therefore, affordable high-throughput automation functionality of an industrial system is still needed.

Recent work has shown the feasibility of carrying out protein crystallization on digital microfluidic biochip. In [38], Srinivasan et al. presented a fabricated digital microfluidic biochip for protein stamping, which is capable of handling transportation and mixing of protein droplets at high concentrations. The implementation of the basic protein droplet operations clearly highlights the promise of a protein crystallization biochip that relies on digital microfluidics. However, no automated chip design technique has thus far been proposed.

1.4. Thesis Outline

This thesis research addresses a number of optimization problems related to biochip design automation. These optimization problems are motivated by practical considerations. Figure 1.5 shows the various topics studied in this thesis. The remainder of the thesis is organized as follows.

Chapter 2 presents a defect-tolerant, routing-aware, architectural-level synthesis methodology. Section 2.1 provides an overview of related prior work on automated synthesis tools and post-synthesis droplet routing for a digital microfluidic biochip. Section 2.2 introduces a new criterion for evaluating droplet routability for a synthesized design, and incorporates it into the overall synthesis flow. Section 2.3 presents pre-synthesis and post-synthesis defect-tolerance methods, and integrates them with the droplet-routing-aware synthesis flow. In Section 2.4, simulation for the dilution steps of a protein assay is used to evaluate the proposed synthesis method. Finally, conclusions are drawn in Section 2.5.

Chapter 3 presents three methods for pin-constrained biochip design, namely array partitioning, cross referencing, and broadcast addressing. Section 3.1 describes the partitioning and pin-assignment algorithms for pin-constrained design of large microfluidic arrays. The proposed array-partitioning-based method is evaluated using a set of real-life bioassays. Section 3.2 presents an alternative pin-constrained design method based on

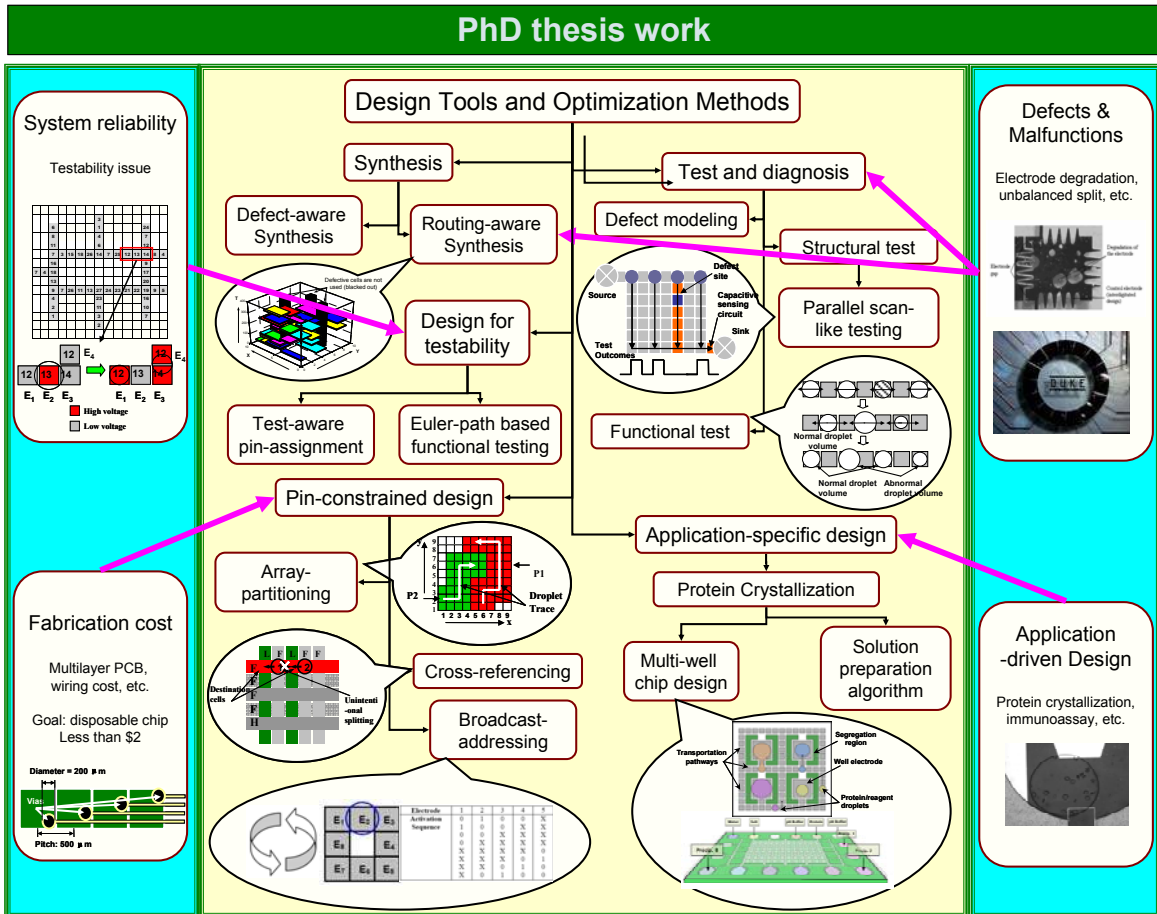


Figure 1.5: An overview of thesis research.

cross-referencing. The cross-referencing-based method is also evaluated using a set of real-life bioassays. The third pin-constrained design method referred to as broadcast addressing is presented in Section 3.3. Section 3.4 analyzes these three methods and concludes the chapter.

Efficient testing and diagnosis methods are proposed in Chapter 4. Section 4.1 relates defects in microfluidic biochips to fault models and observable errors. Next, the proposed parallel “scan-like” test and defect diagnosis scheme for both on-line and off-line testing are introduced. A number of physical defects for microfluidic biochips are listed and fault models are presented. Section 4.3 determines the complexity of the test and diagnosis procedures in terms of the number of droplet manipulation steps required. Section 4.4 presents these results on the application of a fabricated chip. Section 4.5 introduces the concept of functional

testing and proposes effective methods to test the basic operation such as droplet dispensing, droplet transportation, mixing, splitting, and capacitive sensing. In Section 4.6, these functional test techniques are applied to a fabricated chip. Simulation results are also presented. Finally, conclusions are drawn in Section 4.7.

Chapter 5 presents design-for-testability (DFT) for microfluidic biochips. Section 5.1 explains the testability problem. Section 5.2 proposes a testability-aware design method. In Section 5.3, the proposed test-aware design method is applied to a multiplexed bioassay and a PCR assay, and simulation results are presented. Finally, conclusions are drawn in Section 5.4.

Chapter 6 focuses on application-driven design. In Section 6.1, the automation tools described in previous chapters are used for the design of a low-cost, easily manufacturable, high-throughput, and robust chip for protein crystallization. Section 6.2 provides a solution-preparation algorithm that can be used to derive a preparation plan that lists the intermediate mixing steps needed to generate the thousands of target solutions with different sample/reagent concentrations required for protein crystallization. A summary of the chapter is presented in Section 6.3.

Finally, Chapter 7 summarizes the contributions of the thesis and identifies directions for future work.

Chapter 2

Defect-Tolerant and Routing-Aware Synthesis

In this chapter, we present a unified synthesis method that combines defect-tolerant architectural synthesis with droplet-routing-aware physical design [50,51]. *Droplet routability*, defined as the ease with which droplet pathways can be determined, is estimated and integrated in the synthesis flow. The proposed approach allows architectural-level design choices and droplet-routing-aware physical design decisions to be made simultaneously. Pre-synthesis and post-synthesis defect tolerance are also incorporated in the synthesis tool. We use the dilution steps of a protein assay as a case study to evaluate the proposed synthesis method.

2.1 Background

Next-generation biochips are likely to be multifunctional and adaptive “biochemical processing” devices. For example, inexpensive biochips for clinical diagnostics offer high throughput with low sample volumes, and they integrate hematology, pathology, molecular diagnostics, cytology, microbiology, and serology onto the same platform. The emergence of such integrated and multifunctional platforms provides the electronic design automation community with a new application driver and market for research into new algorithms and design tools.

Over the past few years, several automated synthesis methods have recently been proposed for digital microfluidic biochips. These design-automation methods address operation scheduling and module placement for digital microfluidics [14,15,16,17,18]. In Chapter 1, we have reviewed these methods and described a unified synthesis algorithm for microfluidic biochips based on parallel recombinative simulated annealing (PRSA) [15]. The

top-down synthesis flow described in Chapter 1 unifies architecture level design with physical level module placement. This method allows users to describe bioassays at a high level of abstraction, and it automatically maps behavioral descriptions to the underlying microfluidic array.

However, the synthesis flow described in Chapter 1 suffers from two drawbacks. For operation scheduling, it is assumed that the time cost for droplet routing is negligible, which implies that droplet routing has no influence on the operation completion time. While generating physical layouts, the synthesis tool in [15] provides only the layouts of the modules and it leaves droplet routing pathways unspecified. The assumption of negligible droplet transportation times is valid for small microfluidic arrays. However, for large arrays and for biochemical protocols that require several concurrent fluidic operations on-chip, the droplet transportation time is not negligible and routing complexity is non-trivial.

Moreover, due to advances in microfluidic module design (smaller feature sizes, improved materials, etc.), the fluidic operation times are decreasing steadily [49]. However, the droplet transportation times are not decreasing at the same pace. As a result, routing times must be considered during operation scheduling and in the calculation of assay completion times. For the synthesis results derived from the methods proposed in [15], the impact of droplet routing on assay completion time might be significant, and the upper limit on assay completion time might be violated. In such scenarios, the biochip design will no longer correctly implement the desired biochemical procedures. Also, if a synthesized design is not routable, either the chip must be discarded or time-consuming resynthesis must be carried out. To avoid such occurrences, we have to anticipate the availability of routing paths during synthesis. Therefore, droplet routing must be included in the synthesis flow for digital microfluidics.

The other drawback of the synthesis flow described in Chapter 1 is that it is defect-oblivious. It can neither guarantee that the design is robust, i.e., defect-tolerant, nor

does it facilitate reconfiguration techniques that can be used to bypass defects. Therefore, defective chips must be discarded if errors are observed during testing or assay operation. The lack of defect tolerance leads to reduced yield and higher chip cost in an extremely cost-sensitive market. Therefore, defect tolerance needs to be integrated with droplet routing and biochip synthesis.

2.2 Routing-Aware Synthesis

In this section, we describe how we can incorporate droplet routing in the synthesis flow. Droplet-routing methods can be viewed as being either anticipatory, i.e., anticipate the routability (defined qualitatively as the ease of droplet routing) of the synthesized biochip and design the system to be easily routable, or based on post-synthesis routing to find the efficient droplet pathways.

We attempt to provide a guaranteed level of routability for every module pair that needs to be connected to each other. Instead of finding efficient droplet pathways after synthesis, we attempt to achieve high-routability mapping of bioassay protocols to the microfluidic array. We next propose a new method to incorporate droplet routing in the PRSA-based synthesis flow for defect-tolerant microfluidic biochips developed in [50].

2.2.1 Droplet-Routability Estimation

For a synthesized biochip, the droplet-routability of a route between two modules is quantified in terms of the length, measured by the number of electrodes, of the droplet transportation path. Droplet-routability is evaluated in terms of the average length of all the droplet pathways for the complete chip. Also, we have to control the maximum length of droplet paths. Large values for the maximum path length lead to long routing times, e.g., more than 5% of the module operation time, which can have the undesirable consequence of having to halt an assay temporarily until the droplets are routed to their destinations.

Moreover, long routing pathways are likely to be blocked by obstacles, i.e., intermediate modules. For example, in Figure 2.1, all routing pathways from M_1 to M_4 are blocked by M_2 and M_3 , therefore droplet routing is not feasible for this design. Note that guard-ring cells are used to avoid inadvertent mixing and they cannot be used for routing. Synthesized designs with large values for the maximum droplet path length suffer from a high probability of being non-routable. Based on the above considerations, we adopt the maximum droplet path length as a parameter for evaluating routability of a synthesized biochip.

A straightforward technique to derive the routability information is to carry out post-synthesis routing to generate an actual routing plan. However, this approach adds to the computational burden of the synthesis tool. In particular, if a routing plan involving all the droplets on the array is generated for each chromosome in the PRSA-based unified synthesis method, the overall synthesis time will be overwhelming due to the large number of chromosomes and evolution steps in the synthesis flow. Moreover, since we only care about the final synthesis result, we need to reduce the effort spent to generate route plans for the intermediate designs. Therefore, we adopt simple estimates of routability, instead of precisely calculating droplet routes at each step.

The module distance M_{ij} is defined as the length of the shortest path between two interdependent modules M_i and M_j , assuming that there are no obstacles between them. By interdependent module, we refer to module pairs where the operation of one module depends on the operation of the other module.

For example, if optical detection is to be carried out for a mixed droplet, then the optical detector and the mixer are interdependent. Note that in many cases, two interdependent modules may not be able to operate in successive time-steps; for example, a mixed droplet may have to wait for a few cycles since the detector may be busy processing another detection step when the mixing is finished. In such cases, a storage unit is needed, and we consider the storage unit and the detector as interdependent modules. The mixer and the

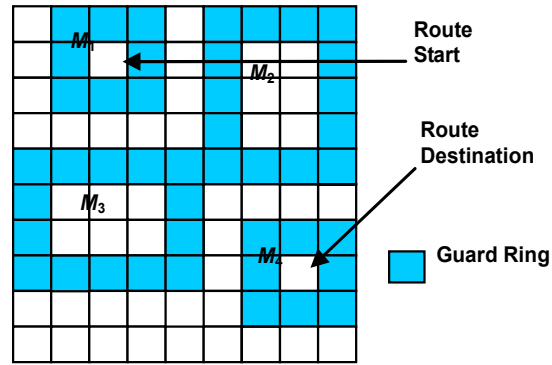


Figure 2.1: An example of a non-routable interdependent pair.

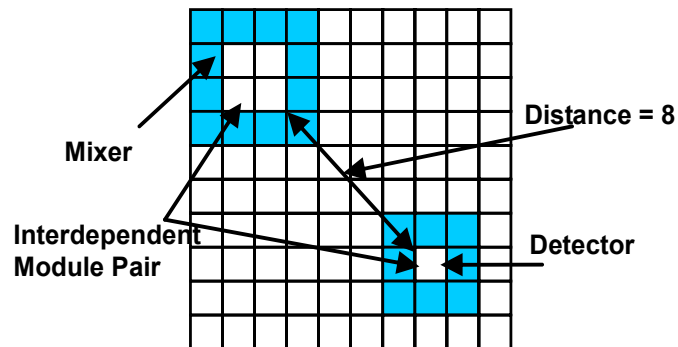


Figure 2.2: Illustration of module distance.

storage unit are also interdependent modules.

Thus droplets are routed only between interdependent modules. The module distance is calculated for each interdependent module pair. Although the module distance M_{ij} may not be exactly the same as the shortest path length, especially if there are obstacles in the form of other modules on the array, M_{ij} is still a good estimate of routability between M_i and M_j .

Note that in some scenarios, the locations of two interdependent modules may overlap on the array. In this case, we set the corresponding module distance to be zero. Since our goal is to guarantee the routability of modules in the synthesized biochip, we adopt the average module distance (over all interdependent modules) as a design metric. Similarly, we adopt maximum module distance to approximate the maximum length of droplet manipulation and use it for routability estimation. For each chromosome considered in the PRSA-based

synthesis flow, we calculate the average and maximum module distance.

Next we incorporate routability in the PRSA-based unified synthesis method. Synthesis results with high routability values are more likely to lead to simple and efficient droplet pathways. To find such designs, we incorporate the above two metrics into the fitness function by a factor that can be fine-tuned according to different design specifications to control the PRSA-based procedure. The pseudocode for the droplet-routing-aware unified synthesis method is shown in Figure 2.3. Candidate designs with low routability are discarded during evolution. Thus, the synthesis procedure guarantees that the routing complexity is reduced for the synthesized biochip, while meeting constraints on array size, bioassay processing time, and defect-tolerance [50].

2.2.2 Routing Time Cost and Assay Completion Time

Next we discuss the impact of routing time cost on bioassay completion time. Here we use the route planning method of [27] to find an efficient route plan for each interdependent pair. The time cost due to the need for droplet transportation is calculated and added to the operation time for the first module in the interdependent module pair. Next the schedule is adjusted based on the modified operation time.

There are two possible scenarios that can arise when the schedule is adjusted. In the first scenario, despite the increased operation time, the fluidic operation can be accommodated in its designated time interval due to the availability of slack or unoccupied time slots in the schedule. In this case, the schedule can simply rely on the available slack or unused time interval for droplet routing. In the second scenario, operations are scheduled so tightly that there is not enough slack available for routing. Here we deal with this problem by adding an extra time slot for routing. As a result, the schedule result is “relaxed” and the completion time is increased. Note that in relaxing the schedule, the ordering of the start times of operations is not changed, therefore, the change in the schedule has no impact on other

PRSA-based droplet-routing-aware synthesis procedure

```
1 Set initial population of chromosome and the initial temperature  $T_\infty$ ;  
2 Implement the synthesis using the information of initial chromosomes:  
  {Phase I: Resource binding; Phase II: Scheduling; Phase III: Placement  
   Phase IV: Routability estimation}  
3 while (Stopping criteria of annealing is not satisfied)  
4   for  $i = 1: N$  /* Inner loop of annealing process */  
5   Find fitness values of chromosomes through construction procedure;  
6   Reproduction; /* Best chromosomes copied to the next generation */  
7   Crossover: {Parameterized uniform crossover is to generate the child  
   chromosome from two randomly-selected parent chromosomes}  
8     if Fitness(child) < Fitness(parents) or  
       rand(0,1) < exp(-[Fitness(child)-Fitness(parents)]/T)  
9       Child chromosome is selected;  
10      else Parent chromosome (the best one) is selected;  
11      end if /* Here a Boltzmann trial is performed */  
12      Mutation; /* New chromosomes are generated randomly */  
13      New population replaces the old generation; end for  
14       $T = \text{rate} \times T$ ; /* update the temperature */ end while  
15 Find the best chromosome from the final population;  
16 Output the results of resource binding, scheduling and placement.
```

Figure 2.3: Pseudocode for the PRSA-based droplet-routing-aware synthesis procedure.

aspects of synthesis, namely resource binding and module placement. The updated assay completion time includes the routing time cost and reflects the actual time needed for executing the biochemical protocol on the synthesized biochip.

2.3 Defect-Tolerant Synthesis

In Section 2.2, we addressed the problem of integrating droplet routing in the synthesis flow. In this section, we focus on enhancing the robustness of the synthesized design. In order to do this, we incorporate defect tolerance as an objective for routing-aware synthesis. Defect tolerance methods can be viewed as being either anticipatory, i.e., anticipate defect occurrences and design the system to be defect-resilient, or based on post-manufacture reconfiguration and re-synthesis. Here we refer to these two types of defect tolerance as pre-synthesis and post-synthesis defect tolerance, respectively.

2.3.1 Post-Synthesis Defect Tolerance

We first focus on post-synthesis defect tolerance. Digital microfluidic biochips are fabricated using standard microfabrication techniques [9]. Due to the underlying mixed technology and multiple energy domains, they exhibit unique failure mechanisms and defects. A manufactured microfluidic array may contain several defective cells. Defects observed include dielectric breakdown, shorts between adjacent electrodes, and electrode degradation [28].

Reconfiguration techniques can be used to bypass faulty cells or faulty optical detectors to tolerate manufacturing defects. Bioassay operations bound to these faulty resources in the original design need to be remapped to other fault-free resources. Due to the strict resource constraints in the fabricated biochip, alterations in the resource binding operation, schedule and placement must be carried out carefully. Our proposed system-level synthesis tool can be easily modified to deal with this issue. To reconfigure a defective biochip, a PRSA-based algorithm along the lines of the one described in Section 2.2 is used. The following additional considerations must be taken into account.

The objective during reconfiguration is to minimize the bioassay completion time while accommodating all microfluidic modules and optical detectors in the fabricated microfluidic array. As resource constraints, the defect-free parts of the microfluidic array and the number of fabricated fault-free non-reconfigurable resources replace the original design specifications. In the placement phase, the locations of the defective cells are no longer available. Note that the locations of non-reconfigurable resources such as integrated optical detectors and reservoirs/ dispensing ports are fixed in the fabricated biochip. Using this enhanced synthesis tool, a set of bioassays can be easily mapped to a biochip with a few defective cells; thus we do not need to discard the defective biochip.

2.3.2 Pre-Synthesis Defect Tolerance

In this subsection, we discuss defect-tolerant design, whereby we attempt to provide guarantees on correct bioassay operation even if the manufacturing process introduces defects. Instead of handling defects after they are detected, we attempt to achieve defect-tolerant mapping of bioassay protocols to the microfluidic array under broad assumptions of defect occurrences.

The synthesis method described in Section 2.2 suffers from two main drawbacks. First, it does not anticipate defect occurrences and it does not consider defect tolerance in the synthesis flow. Instead, it relies on the availability of unused cells in the microfluidic array to avoid defective cells that are located after manufacture. However, such a re-synthesis procedure might not be feasible because of lack of availability of spare cells. Moreover, the impact on assay completion time might be significant, and the upper limit on assay completion time might be violated. In such scenarios, the fabricated biochip must be discarded. A second drawback of defect-oblivious synthesis is that after defects are identified, the complete synthesis process must be repeated. Thus this approach imposes additional computation burden on the design and implementation process.

We next present a new method to incorporate defect tolerance in the unified synthesis flow for microfluidic biochips. A novel partial reconfiguration method is also presented to enhance defect tolerance after the device is manufactured.

Defect Tolerance Index

The defect tolerance of a synthesized biochip can be evaluated in terms of survivability, i.e., the capability to perform bioassays on a microfluidic array with defects. The Defect Tolerance Index (DTI), is defined as the probability that defect tolerance can be achieved via successful partial reconfiguration when the array contains defective cells [36]. Partial reconfiguration refers to the relocation only of the modules that contain defective cells; other

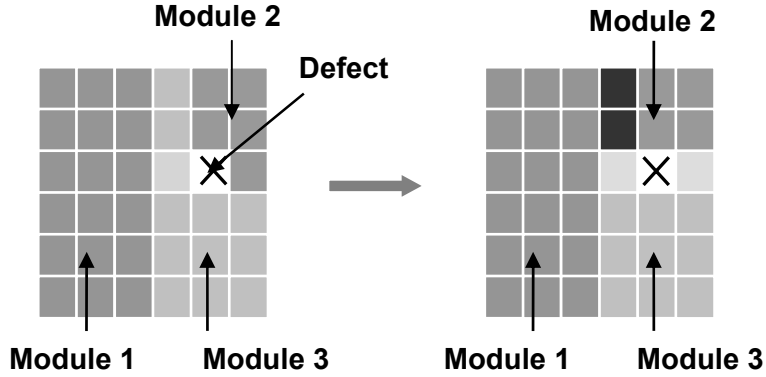


Figure 2.4: Example of partial reconfiguration.

modules are not affected. The relocated modules therefore “survive” through the defects (see Figure 2.4).

Assume that each cell in the microfluidic array has an independent failure probability p . The DTI $D(G)$ value for a layout G can be estimated by multiplying the survival probabilities of all the modules, as follows [50]:

$$D(G) \approx \prod P_s(M_i) = \prod (1 - f_1(M_i) + f_1(M_i) \times f_2(M_i))$$

where M_i , $i = 1 \dots N$, is a microfluidic module (e.g., mixer) contained in a given layout G , and $P_s(M_i)$ is the survival probability of module M . Note that $f_1(M_i)$ is the probability that the module M_i is faulty. It is determined by the equation $f_1(M_i) = 1 - p \cdot A(M_i)$, where $A(M_i)$ is the total number of cells contained in M_i . Finally, $f_2(M_i)$ is the probability that M_i can be successfully reconfigured if it becomes faulty [28].

Now we incorporate DTI into the PRSA-based unified synthesis method. We first define layout vulnerability by $V = 1 - D$. Layouts with low vulnerability are likely to provide high probability of successful partial reconfiguration. To find such designs, we combine vulnerability with time- and area-cost to derive a new fitness function to control the PRSA-based procedure. Candidate designs with low survivability are discarded during evolution. Thus, the synthesis procedure anticipates defect occurrences and selects designs that allow reconfiguration of large number of modules, while meeting constraints on array size and bioassay processing time.

Partial Reconfiguration and Partial Resynthesis

Next we discuss how defects can be bypassed after manufacture. In the defect-oblivious approach described in Section 2.2, defect tolerance is achieved by complete resynthesis, which can be very time consuming. Here we propose an efficient method to achieve defect tolerance without the need for resynthesis. This method is based on the concept of partial reconfiguration, which was introduced in Section 2.3.2. If the number of defective cells is not excessive, most microfluidic modules on the array are not affected and they do not need not to be reconfigured. As discussed in the Section 2.3.2, the incorporation of defect tolerance in the design flow ensures a high probability of partial reconfigurability of the modules, i.e., it is very likely that the defective biochip can be made usable via partial reconfiguration.

For each affected module, we search the array for available defect-free areas for partial reconfiguration. This can be accomplished fast, because the search space is restricted to the layouts in the modules' time duration. Once a module is relocated, the algorithm updates the corresponding layout and starts the search for the next module. Resources binding and scheduling results are not changed. Only the placement of defective modules is modified. Therefore, this method is much faster compared to a complete resynthesis procedure.

In some cases, there may not be a sufficient number of defect-free cells to carryout partial reconfiguration for some defective modules. We therefore introduce a new method called partial resynthesis. The key idea here is to truncate the bioassay and carry out resynthesis only for the modules that start later than the earliest-in-use defective module. Although this partial resynthesis procedure may take as much time as complete resynthesis in the worst case, i.e., if the first in-use module is defective and cannot be relocated, it is faster on average than the complete resynthesis procedure.

Using these two methods, the complexity of doing post-manufacture processing for defect tolerance can be greatly reduced compared to resynthesis. The time needed to complete a set of bioassays is also significantly decreased.

2.4 Simulation Results

In this section, we evaluate the defect-tolerant droplet-routing-aware synthesis method by using it to design a biochip for a real-life protein assay.

Recently, the feasibility of performing a colorimetric protein assay on a digital microfluidic biochip has been successfully demonstrated [38]. Based on the Bradford reaction [9], the protocol for a generic droplet-based colorimetric protein assay is as follows. First, a droplet of the sample, such as serum or some other physiological fluid containing protein, is generated and dispensed into the biochip. Buffer droplets, such as 1M NaOH solution, are then introduced to dilute the sample to obtain a desired dilution factor (DF). This on-chip dilution is performed using multiple hierarchies of binary mixing/splitting phases, referred to as the interpolating serial dilution method [9]. The mixing of a sample droplet of protein concentration C and a unit buffer droplet results in a droplet with twice the unit volume, and concentration $C/2$. The splitting of this large droplet results in two unit-volume droplets of concentration $C/2$ each. Continuing this step in a recursive manner using diluted droplets as samples, an exponential dilution factor of $DF = 2^N$ can be obtained in N steps. After dilution, droplets of reagents, such as Coomassie brilliant blue G-250 dye, are dispensed into the chip, and they mix with the diluted sample droplets. Next the mixed droplet is transported to a transparent electrode, where an optical detector (e.g., a LED-photodiode setup) is integrated. The protein concentration can be measured from the absorbance of the products of this colorimetric reaction using a rate kinetic method [38]. Finally, after the assay is completed, all droplets are transported from the array to the waste reservoir. A sequencing graph model can be developed from the above protocol for a protein assay ($DF = 128$), as shown in Figure 2.5. There are a total of 103 nodes in one-to-one correspondence with the set of operations in a protein assay, where DsS , DsB_i ($i = 1, \dots, 39$), and DsR_i ($i = 1, \dots, 8$) represents the generation and dispensing of sample, buffer and reagent droplets, respectively.

In addition, Dlt_i ($i = 1, \dots, 39$) denotes the binary dilution (including mixing/splitting) operations, Mix_i ($i = 1, \dots, 8$) represents the mixing of diluted sample droplets, and reagent droplets; Opt_i ($i = 1, \dots, 8$) denotes the optical detection of the droplets. Until the fourth step of a serial dilution, all diluted sample droplets are retained in the microfluidic array. After that stage, for each binary dilution step, only one diluted sample droplet is retained after splitting, while the other droplet is moved to the waste reservoir.

The basic operations for the protein assay have been implemented on a digital microfluidic biochip [9]. Experiments indicate that the dispensing operation takes 7 seconds [9]. The operation times of various mixers have been found to be different [9]. A binary dilution operation can also be easily implemented by mixing of sample droplet followed by droplet splitting. Absorbance of the assay product can be measured using an integrated LED-photodiode setup. Experiments indicate this absorbance measurement takes 30 seconds.

The microfluidic module library for a protein assay is shown in Table 2.1. A total of 122 interdependent module pairs must be routed for this protocol. Clearly, the large number of droplet transportation operations in this protocol makes it difficult for a biochemist user or a post-synthesis design tool to determine transportation paths. We also need to specify some design parameters for the biochip to be synthesized. Different design specifications can be determined based on user needs and manufacturing constraints.

2.4.1 Results for Routing-Aware Synthesis

We first evaluate the proposed routing-aware synthesis method described in Section 2.2. We apply it to an example in which we set the maximum microfluidic array size to be 100 cells, and the maximum allowable completion time for the protein assay to be 400 seconds. We assume that there is only one on-chip reservoirs/dispensing port available for sample fluids, but two such ports for buffer fluids, two for reagent fluids, and one for waste fluids. Finally, we assume that at most four optical detectors can be integrated into this biochip.

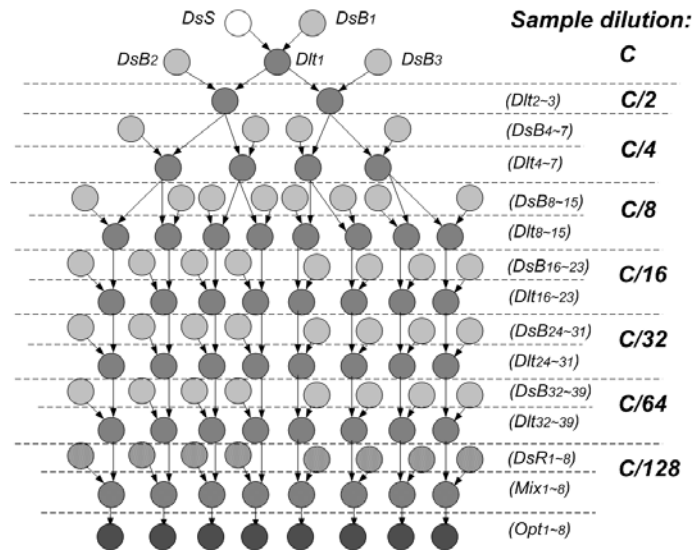


Figure 2.5: Sequencing graph for a protein assay.

Table 2.1: Experimentally-characterized module library for synthesis.

Operation	Resource	Time (s)
DsS;DsB; DsR	On-chip reservoir/dispensing port	7
Dlt	2x2-array dilutor	12
	2x3-array dilutor	8
	2x4-array dilutor	5
	4-electrode linear array dilutor	7
Mix	2x2-array mixer	10
	2x3-array mixer	6
	2x4-array mixer	3
	4-electrode linear array mixer	5
Opt	LED+Photodiode	30
Storage	Single cell	N/A

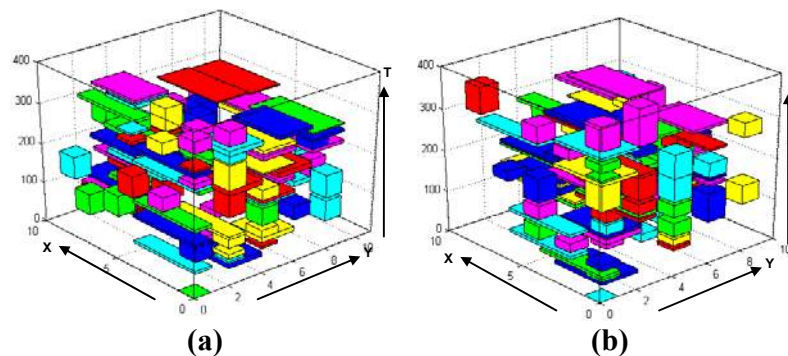


Figure 2.6: A 3-D model illustrating the synthesis results: (a) routing-oblivious method of [15]; (b) the proposed method.

We first use the routability-oblivious PRSA-based algorithm from [15] to find a desirable solution for the protein assay that satisfies design specifications. The solution thus obtained yields a biochip design with a 10×10 microfluidic array, an assay completion time of 377 seconds, a maximum module distance of 14 electrodes, and an average distance of 3 electrodes. Next we use the droplet-routing-aware synthesis method using the procedure of Figure 2.3. The procedure yields a biochip design with a 10×10 microfluidic array, a completion time of 378 seconds, a maximum module distance of 7 electrodes, and an average distance of 1 electrode. The computation time for the routability-oblivious and routing-aware methods for the protein assay are 4 minutes and 5 minutes, respectively on a 3.00 GHz dual-core Intel Xeon server with 4 G of RAM.

We illustrate the synthesis results, i.e., assay operation schedule and module placement, using a 3-D box model shown in Figure 2.6. Each microfluidic module is represented as a 3-D box, the base of which denotes the rectangular area of the module and the height denotes the time-span of the corresponding assay operation. The projection of a 3-D box on the X-Y plane represents the placement of this module on the microfluidic array, while the projection on the Z-axis (time axis) represents the schedule of the assay operation. Note that the synthesis results determine the locations of integrated optical detector. Transparent electrodes for optical detection are used in the microfluidic array.

Although the two designs have comparable area- and time- cost, the routing-aware synthesis method leads to a 50% reduction in the average and maximum module distance. This indicates a significant improvement of routability and reduction of the time-cost for carrying out droplet-routing. To verify this improvement, we applied the post-synthesis routing method of [27] to find efficient droplet pathways routing for both layouts. We find that while routing-aware synthesis easily leads to a feasible routing plan, the layout for the routing-oblivious result is not routable, i.e., no pathway is available for certain droplet manipulations Figure 2.7(a) shows a snapshot of the layout for the routing-oblivious result

taken at time instant 297 s. In this snapshot, a droplet is to be routed from the storage unit S_1 to dilutor D_3 , which is located 7 electrodes away in the routing-oblivious layout. However, as shown in Figure 2.7 (a), there is no pathway available for routing due to the compact layout and large module distance. In contrast, in the layout derived from the routing-aware synthesis procedure, since the average module distance is minimized, D_3 is placed next to S_1 and the droplet pathway can be trivially determined.

Thus we can see that without violating constraints on time- and area-cost, the routing-aware method carefully arranges interdependent modules to be close to each other. Therefore, it ensures that droplet pathways can be determined with a high probability. On the other hand, the routing-oblivious method only aims at meeting satisfying constraints on time- and area- cost. As a result, the interdependent modules are likely to be segregated by other modules when routing-oblivious synthesis is employed; a consequence of this is that routing solutions cannot be obtained. Without a careful arrangement of modules, routing-oblivious synthesis can find feasible routes only if the area constraint is fairly loose, thereby enough chip area is available to create droplet pathways. As a result, time- and area-cost are compromised and the design specifications might not be met.

We examine this issue as follows. We first synthesize the protein assay under a set of design specifications using both the routing-oblivious and routing-aware synthesis methods. limits ($T = \{T_1, T_2, T_3, \dots, T_n\}$) and a set of area-cost limit ($A = \{A_1, A_2, A_3, \dots, A_n\}$). Therefore, each synthesized chip G_{ij} corresponds to a point (T_i, A_j) . For each synthesized chip, we check if it is routable. A point (T_i, A_j) is referred to as a feasibility boundary point if there are no other points (T_m, A_n) such that G_{ij} is routable and $T_m < T_i, A_n < A_j$. A feasibility frontier is defined by connecting all the feasibility boundary points. Therefore, the feasible design region is defined by the area above the feasibility frontier. Here we set $T = \{320, 340, 360, \dots, 440\}$ and $A = \{60, 70, 80, \dots, 180\}$ and carry out both the routing-oblivious and routing-aware

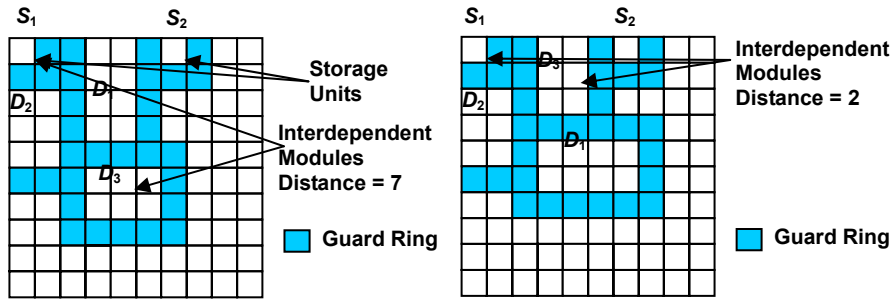


Figure 2.7: (a) A snapshot of a non-routable layout from routing-oblivious synthesis (time instant 297 s) (b) Corresponding layout in routing-aware synthesis (time instant 299 s).

synthesis (the unit of T is seconds while the unit of A is measured in terms of the number of electrodes). The feasibility frontier is derived for both methods, as plotted in Figure 2.8. Note that in finding the feasibility frontier, we fix a time limit and search for the minimum chip area for which a routable synthesis result is available.

As shown in Figure 2.8, routing-aware synthesis leads to a lower feasibility frontier and a larger feasible design region. For tight time limits, e.g., 320s, the routing-aware method achieves a routable synthesis result with less than 140 electrodes, while the routing-oblivious method fails when the area limit is lower than 170 electrodes. On the other hand, for a fixed array size, e.g., 110 electrodes, routing-aware synthesis leads to a much lower assay completion time (less than 360s) than the routing-oblivious method (between 380s and 400s). The improvement becomes more significant when routing time is considered and added to the assay completion time. In addition to assay-time reduction, routing-aware synthesis allows us to reduce chip area, and thereby the product cost for disposable and reusable biochips.

Next, we carry out post-synthesis routing for all the routable synthesis results corresponding to the feasible layouts and use schedule relaxation as defined in Section 2.2 to derive the adjusted completion time. We add the droplet transportation time to the assay completion time in each case. The results are shown in Figure 2.9. These results show that in addition to providing a greater range of feasible design points, routing-aware

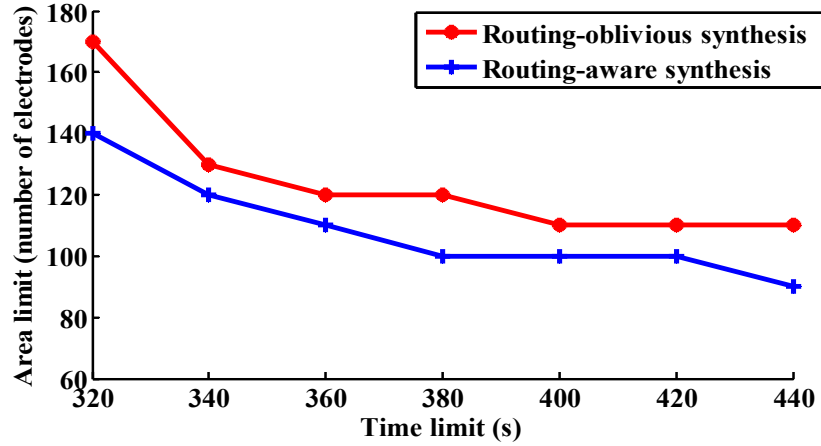


Figure 2.8: Feasibility frontier and feasible design region for [15] and the routing-aware synthesis method.

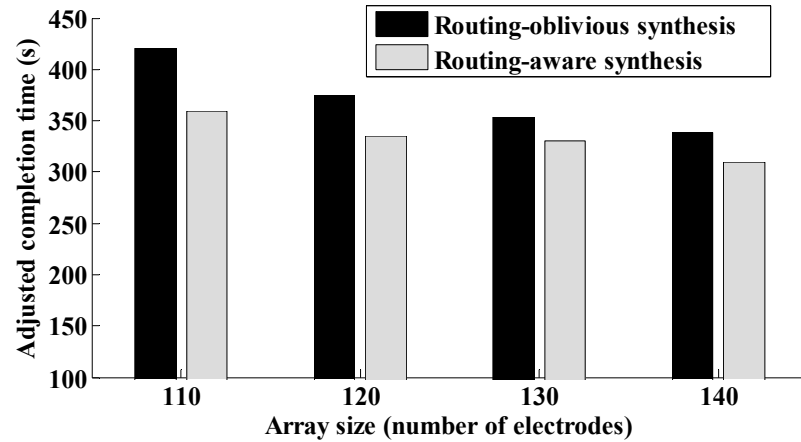


Figure 2.9: Assay completion times (with droplet transportation time included) for [15] and for the proposed routing-aware synthesis method.

synthesis leads to lower assay completion times.

2.4.2 Results for Post-Synthesis Defect Tolerance

Next we investigate defect tolerance using the above example. Assume that the above biochip has been fabricated. Suppose that due to particle contamination, 3 cells in the 10×10 microfluidic array in Figure 2.6 are rendered defective, as shown in Figure 2.10(a). In order to ensure that the protein assay can still be carried out on this biochip, we need to bypass these faulty cells during assay operation. Moreover, due to defective cells, some non-reconfigurable resources may no longer be available. In this example, we assume one

that optical detector is rendered defective after manufacturing. Thus the operations assigned to this detector have to be remapped to other detectors. The modified synthesis method proposed in Section 2.3.1 is used here to carry out the reconfiguration to tolerate these manufacturing defects. The reconfiguration results are shown in Figure 2.10(b). This new design, allows the protein assay to operate on this defective biochip with an increase of only 7% in the completion time, i.e., the completion time is now 387 seconds.

2.4.3 Results for Pre-Synthesis Defect Tolerance

We next evaluate the pre-synthesis defect tolerance that is achieved for the protein assay using the enhanced routing-aware method described in Section 2.3.2. We first use this method to find a desirable solution that satisfies design specifications. The solution thus obtained yields a biochip design with a 10×10 microfluidic array and an assay completion time of 390 seconds. Next, we compare it with a design synthesized using the routing-aware defect-oblivious method described in Section 2.2. The metric for comparison is the amount of defect tolerance exhibited by each design. For the defect-oblivious case, we use the design shown in Figure 2.6(b), which is a 10×10 microfluidic array with an assay completion time of 377 seconds.

We evaluate the defect tolerance of the two synthesized designs by injecting random defects. A design is deemed to be robust if the injected defects can be bypassed by partial reconfiguration. Defects can be classified based on their impact on bioassay functionality.

The first category includes defects that affect only the unused cells in the array. As the biochip functionality is not compromised, these defects are referred to as *benign*. The second category refers to defects that cause significant “fragmentation” of the array, whereby it is no longer possible to relocate a microfluidic module to another part of the array due to lack of availability of defect-free cells. These defects are referred to as *catastrophic*. The third category includes defects that are neither benign nor catastrophic. The microfluidic array can

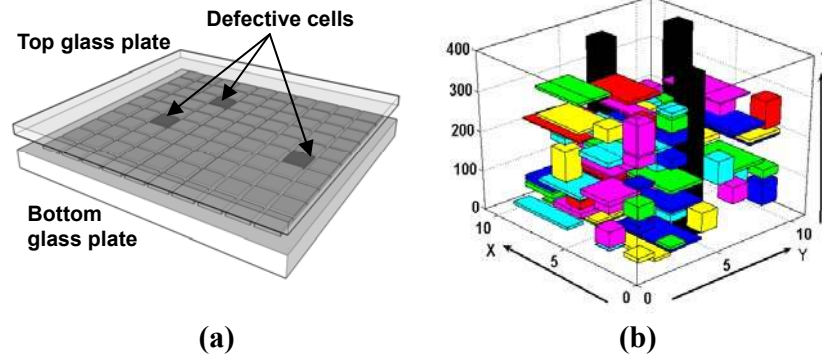


Figure 2.10: (a) A defective 10×10 microfluidic array (b) Reconfiguration results for post-synthesis defect tolerance.

be reconfigured for such defects; hence we refer to these defects as *repairable*.

A biochip that contains only benign defects is placed in Group I. A biochip that contains catastrophic defects is placed in Group II. Finally, a biochip that contains only repairable and benign defects is placed in Group III. Let N_t be the total number of biochips in a representative sample, and let N_i be the number of biochips in Group i , $1 < i < 3$. Clearly $N_1 + N_2 + N_3 = N_t$. We next define two ratios related to the defect tolerance capability of the synthesized biochip: (i) robustness index $r = (N_1 + N_3) / N_t$; (ii) failure index $f = N_2 / N_t$. The goal of defect-aware synthesis is to maximize r and minimize f .

Resynthesis must be carried out for biochips in Group II, i.e., for biochips that suffer from catastrophic defects. Let the bioassay completion time before (after) resynthesis be T_1 (T_2). We define the time degradation td as follow: $td = (T_2 - T_1) / T_1$. Another goal of defect-aware synthesis is to minimize td .

We take 100 simulated samples of a microfluidic biochip synthesized for the protein assay with defect tolerance as a criterion and without defect tolerance, i.e., in a defect-oblivious manner. In each case, we randomly inject defects by assuming that each unit cell is defective with probability p ($p = 0.01, 0.05, 0.1$ in our experiments). We then determine the ratios r, f , and td for both methods. The results are shown in Table 2.2.

As shown in Table. 2.2, although the defect-tolerant design leads to slightly higher assay

times, this design leads to a DTI value of 0.8918, which implies that almost 90% of the modules can be reconfigured if they are affected by defects. This is a considerable improvement over the DTI value of 0.0144 obtained using the defect-oblivious method. This improvement is also apparent from the comparison of the failure ratio (f), the robustness index (r) and the time degradation (td) from Table 2.2.

For all three value of p , defect-aware synthesis results in a higher value of r and a considerably lower f . Moreover, the defect-aware biochip design also provides a much lower value of td , which implies that for resynthesized biochips, the performance is compromised much less. Since the original time-cost for the two methods are comparable, the difference in td is therefore even more significant. Moreover, td falls more sharply for smaller values of p for the defect-aware design. Therefore, for low defect probabilities, as is often the case for mature manufacturing processes, the proposed defect tolerant synthesis method allows resynthesis in the case of catastrophic defects with lower time-cost increase. This feature is often required for many biochip applications.

In summary, the incorporation of pre-synthesis defect tolerance into the routing-aware synthesis tool leads to a significant improvement in the robustness of the synthesized design. It also allows the search for an optimal design under multiple design specifications including completion time, chip area, routability, and system dependability.

For the protein assay example, we next run the defect-tolerant routing-aware and defect-oblivious routing-aware algorithms under a set of combinations of weights in the fitness function. For each combination of weights, if the derived synthesis result is not routable, the algorithm is repeated until a routable design is found. Next we carry out random defect injection to each design and obtain its failure rate f as defined in Section 2.3. We map each design G to a 3-D point (T_G, A_G, F_G) where T_G, A_G, F_G are completion time, chip area and failure rate of the design respectively. Similar to the definition in Section 2.4.1, a point (T_G, A_G, F_G) is referred to as a feasibility boundary point if there are no other points $(T_m,$

Table 2.2: Defect tolerance for defect-oblivious and defect-tolerant method for the protein assay.

	DTI value	Assay time (s)	Array area	r	f	td
Defect-oblivious	0.0144	377	10×10	0.12	0.88	1.45
Defect-tolerant	0.8918	390	10×10	0.83	0.17	1.21

(a) $p = 0.1$

	DTI value	Assay time (s)	Array area	r	f	td
Defect-oblivious	0.0144	377	10×10	0.22	0.78	1.29
Defect-tolerant	0.8918	390	10×10	0.83	0.17	1.09

(b) $p = 0.05$

	DTI value	Assay Time (s)	Array Area	r	f	td
Defect-oblivious	0.0144	377	10×10	0.30	0.70	1.21
Defect-tolerant	0.8918	390	10×10	0.94	0.07	1.04

(c) $p = 0.01$

A_m, F_m) such that $T_m < T_G$, $A_m < A_G$ and $F_m < F_G$. A feasibility frontier surface is obtained by connecting all the feasibility boundary points, as shown in Figure 2.11. The feasible design region corresponds to the space above the feasible surface. Any design specification, whose corresponding is point located in this region, can be met. Otherwise no feasible design exists for this specification.

As shown in Figure 2.11, defect-tolerant routing-aware synthesis leads to a lower feasibility frontier surface and a larger feasible design space than the defect-oblivious method. For tight time and area limits, e.g., 400s, 110 electrodes, the defect-tolerant method achieves a routable synthesis result with failure rate less than 0.5, while the defect-oblivious method requires a failure rate of at most 0.1. On the other hand, for a predetermined failure rate limit, e.g., 0.3, defect-aware synthesis leads to a much lower assay completion time (less than 390s) and smaller chip area (less than 100 electrodes) than the defect-oblivious method, which requires a completion time of 450s and a minimum chip area of 150 electrodes. The improvement becomes more significant for limits on the failure rate. Defect-aware synthesis allows us to reduce chip area and assay completion time, and thereby the product cost for reliable biochip platforms.

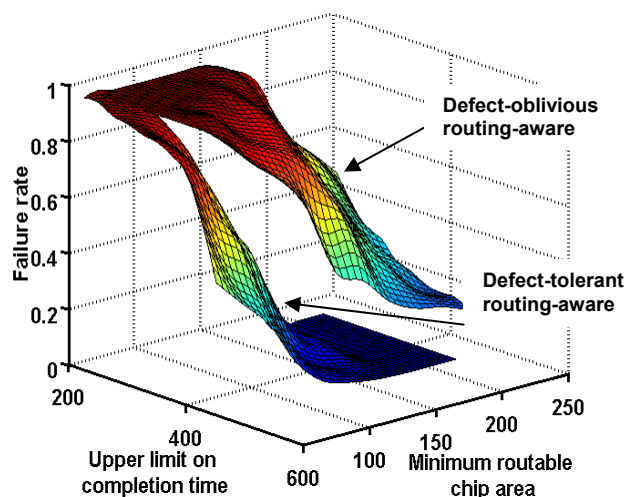


Figure 2.11: Feasibility frontier surface and feasible design region for defect-tolerant and defect-oblivious routing-aware synthesis methods.

2.5 Chapter Summary and Conclusions

We have presented a droplet-routing-aware automated synthesis tool for microfluidic biochips. Droplet routability, defined as the ease with which droplet pathways can be determined, has been estimated and integrated in the synthesis flow. The proposed method increases the likelihood that feasible droplet pathways can be for area-constrained biochip layouts. We have demonstrated the advantages of this approach using a large-scale protein assay based on the Bradford reaction. To increase system dependability, two defect-tolerance schemes, i.e., pre-synthesis and post-synthesis, have been incorporated into the routing-aware design method. Simulation results have highlighted the increase in defect tolerance achieved in each case. The design techniques presented in this chapter relieves the chip user from the burden of post-synthesis droplet routing, and they facilitate the automated design of biochips with guaranteed high levels of defect tolerance. In this way, the biochip user can concentrate on the development of nano- and micro-scale bioassays, leaving cumbersome implementation details to the synthesis tools.

Chapter 3

Pin-Constrained Chip Design

In this chapter, we present three design-automation techniques for pin-constrained biochips. In contrast to the direct-addressing scheme that has mostly been studied thus far in the literature [14,15,16,17,18,51,52,53], we assign a small number of independent control pins to a large number of electrodes in the biochip, thereby reducing design complexity and product cost.

3.1 Droplet-Trace-Based Array-Partitioning Method

We first present an array-partitioning-based pin-constrained design method [57]. The design procedure relies on a droplet-trace-based array-partitioning scheme and an efficient pin-assignment technique, referred to as the “Connect-5 algorithm”. The proposed method is evaluated using a set of multiplexed bioassays.

3.1.1 Impact of Droplet Interference and Electrode-Addressing Problem

In this subsection, we formulate the pin-constraint problem for a two-dimensional electrode array. The goal is to use a limited number of independent control pins to control the electrodes of a digital microfluidic array. However, the sharing of control pins leads to the problem of droplet interference, which is defined as the inadvertent activation of multiple electrodes on which droplets are incident at any time instant. Droplet interference results in unintentional droplet operations caused by the simultaneous activation or deactivation of electrodes that are controlled by the same pin.

Impact of Droplet Interference

A pin-constrained layout may result in unintentional droplet movement when multiple

droplets are present in the array. Figure 3.1 shows a 4×4-array in which the 16 electrodes are controlled by only 9 input pins. The pin numbers are indicated in the figure. Droplet interference occurs if we attempt to move droplet D_i while keeping droplet D_j at its current location. Suppose D_i is at coordinate location (0,0) and D_j is at coordinate location (3,2). To move D_i to (1,0), we need to activate electrode (1,0) and deactivate (0,0). This implies that a high voltage must be applied to Pin 8 while a low voltage must be applied to Pin 1. Note however that a high voltage on Pin 8 also activates electrode (3,3). This results in the inadvertent stretching of droplet D_j across electrodes (3,2) and (3,3).

The sharing of control pins can also affect a single droplet. An example is shown in Figure 3.2. To move droplets D_i one electrode to the left requires Pin 8 to be activated. However, the electrode on the right of the droplet is also connected to Pin 8; it is therefore also activated. As a result, D_i is pulled from both sides and it undergoes inadvertent splitting. The above example shows that the sharing of control pins can lead to unintentional operations such as droplet splitting and inadvertent movement due to droplet interference. This problem therefore must be avoided in any practical pin-assignment layout.

Minimum Number of Pins for a Single Droplet

Given a two-dimensional microfluidic array, the problem of determining the minimum number of independent control pins, k , necessary to have full control of a single droplet without interference can be reduced to the well-known graph-coloring problem [56]. Full control implies that a droplet can be moved to any cell on the array through an appropriate electrode activation sequence. While the problem of finding the chromatic number of a graph is NP-Complete [57], it is trivial to observe that for rectangular arrays of size greater than 3×3, the largest number of directly adjacent neighbors to any cell is four. Hence, if k denotes the number of independent control pins, we ensure that $k \geq 5$ such that each cell and all of its directly adjacent neighbors can be assigned different

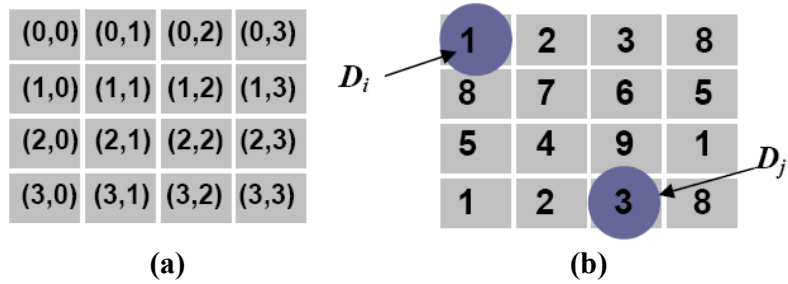


Figure 3.1: An example to illustrate droplet interference due to the sharing of control pins by the electrodes: (a) coordinate locations for the electrodes; (b) pin-assignment for the electrodes.

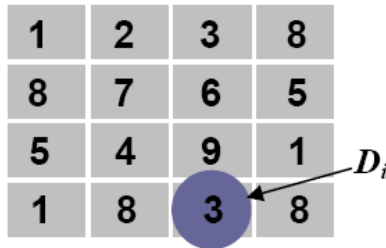


Figure 3.2: An example of an inadvertent operation for a single droplet.



Figure 3.3: A 5-pin layout for a 4x4 array.

colors. A possible pin layout using 5 pins for a 4x4 array is shown as an example in Figure 3.3.

Pin-Assignment Problem for Two Droplets

We next examine the interference problem for two droplets. For more than two droplets, the interference problem can be reduced to the two-droplet problem by examining all possible pairs of droplets. In general, any sequence of movements for multiple droplets can occur in parallel. We analyze interference between two droplets for a single clock cycle, during which

time a droplet can only move to a directly adjacent cell. Any path can be decomposed into unit movements, and we say that the two paths are compatible if and only if all of their individual steps do not interfere.

In some situations, we would like both droplets to move to another cell in the next clock cycle. If this is not possible without interference, then a contingency plan is to have one droplet undergo a stall cycle (i.e., stay on its current cell). There are other possibilities such as an evasive move or backtracking to avoid interference, but these lead to more substantial changes in the scheduled droplet paths and are therefore not considered here.

Let us denote two droplets by D_i and D_j , with the position of droplet D_i at time t given by $P_i(t)$. Let $N_i(t)$ be the set of directly adjacent neighbors of droplet D_i . The operator $\bar{k}(\bullet)$ is the set of pins that control the set of cells given by \bullet . Then the problem of two droplets moving concurrently can be formally stated as: D_i moves from $P_i(t)$ to $P_i(t+1)$, and D_j moves from $P_j(t)$ to $P_j(t+1)$.

We are interested in the overlap of pins between sets of cells for the interference constraints, rather than the spatial locations of the cells. The latter are important for the fluidic constraints discussed in [27]. For the purpose of defining interference behavior, the system is completely determined by the positional states of the two droplets at times t and $t+1$. For droplet D_i , the positional states are characterized by the quartet $(P_i(t), P_i(t+1), N_i(t), N_i(t+1))$ and for D_j , the quartet $(P_j(t), P_j(t+1), N_j(t), N_j(t+1))$. We consider $\bar{k}(\bullet)$ of all unordered pairs involving the following sets: $P_i(t), P_i(t+1), N_i(t), N_i(t+1), P_j(t), P_j(t+1), N_j(t), N_j(t+1)$. Since $\binom{8}{2} = 28$, there are 28 such pairs that need to be mutually exclusive to prevent interference between the droplets. Pairs of cell sets that must be mutually exclusive for non-interference must be contained in this pool of unordered pairs because these eight sets include all cells currently occupied by droplets and all of their neighboring cells. A total of 22 pairs can be quickly removed from consideration, leaving only six pairs of sets that need to be closely examined to determine if their mutual exclusion is required for non-interference.

For simplicity, the pin operator $\bar{k}(\bullet)$ is left implicit in the following discussion. Mutual exclusion always refers to the *pins* controlling the cells, and not the cells themselves. We can analytically confirm that the control pins for these six pairs must be mutually exclusive to prevent interference.

Pair 1 $\{P_i(t), N_j(t)\}$: if $N_j(t)$ contains a cell that shares the same pin as $P_i(t)$, then D_j may be between $P_j(t)$ and $P_j(t+1)$ at time t . If this is not the case, D_j will not be able to move to $P_j(t+1)$ because it will no longer overlap with $P_j(t+1)$.

Pair 2 $\{P_i(t+1), N_j(t)\}$: if $N_j(t)$ contains a cell that shares the same pin as $P_i(t+1)$, D_j will not move properly at time $t+1$ unless $P_j(t+1)$ is the cell in $N_j(t)$ that shares the same pin as $P_i(t+1)$.

Pair 3 $\{P_i(t+1), N_j(t+1)\}$: if $N_j(t+1)$ contains a cell that shares the same pin as $P_i(t+1)$, then D_j will drift after moving to $P_j(t+1)$ so that it ends up between $P_j(t+1)$ and a cell that is an element of $N_j(t+1)$.

Pair 4 $\{N_i(t), P_j(t)\}$: if $N_i(t)$ contains a cell that shares the same pin as $P_j(t)$, then at time t , D_i will drift between $P_i(t)$ and the cell of interest in $N_i(t)$. D_i may no longer overlap with $P_j(t+1)$; if so, it will not be able to move to $P_j(t+1)$.

Pair 5 $\{N_i(t), P_j(t+1)\}$: if $N_i(t)$ contains a cell that shares the same pin as $P_j(t+1)$, D_i will not move properly at time $t+1$ unless $P_i(t+1)$ is the cell in $N_i(t)$ that shares the same pin as $P_j(t+1)$.

Pair 6 $\{N_i(t+1), P_j(t+1)\}$: if $N_i(t+1)$ contains a cell that shares the same pin as $P_j(t+1)$ then at time $t+1$, D_i will drift between $P_i(t+1)$ and the cell of interest in $N_i(t+1)$.

It can be easily shown that the control pins for these six pairs must be mutually exclusive to prevent interference between droplets D_i and D_j . These lead to the following necessary and sufficient interference constraints:

- (1) $\bar{k}(\{P_i(t)\}) \cap \bar{k}(N_j(t)) = \phi$
- (2) $\bar{k}(\{P_i(t+1)\}) \cap \bar{k}(N_j(t+1)) = \phi$
- (3) $\bar{k}(N_i(t)) \cap \bar{k}(\{P_j(t)\}) = \phi$
- (4) $\bar{k}(N_i(t+1)) \cap \bar{k}(\{P_j(t+1)\}) = \phi$
- (5) $\bar{k}(\{P_i(t+1)\}) \cap \bar{k}(N_j(t) - \{P_j(t+1)\}) = \phi$
- (6) $\bar{k}(\{P_j(t+1)\}) \cap \bar{k}(N_i(t) - \{P_i(t+1)\}) = \phi$

Moving one droplet and stalling the other is a special case that can be used when concurrent movement of two droplets leads to violation of the interference constraints. In this situation, $P_j = P_j(t) = P_j(t+1)$ and the interference constraints reduce to:

- (1) $\bar{k}(\{P_i(t)\}) \cap \bar{k}(N_j) = \phi$
- (2) $\bar{k}(\{P_i(t+1)\}) \cap \bar{k}(N_j) = \phi$
- (3) $\bar{k}(N_i(t)) \cap \bar{k}(\{P_j\}) = \phi$
- (4) $\bar{k}(N_i(t+1)) \cap \bar{k}(\{P_j\}) = \phi$
- (5) $\bar{k}(\{P_i(t+1)\}) \cap \bar{k}(N_j - \{P_j\}) = \phi$
- (6) $\bar{k}(\{P_j(t+1)\}) \cap \bar{k}(N_i - \{P_i\}) = \phi$

The above constraints must be satisfied during droplet routing [27]. Therefore, they are enumerated here for the sake of completeness.

3.1.2 Array Partitioning and Pin-Assignment Methods

In this subsection, we propose a pin-constrained design method for digital microfluidic biochips based on array partitioning. The key idea is to “virtually” partition the array into regions. Similar partitioning techniques have been used for VLSI circuits and for microarray biochips [58,59,60].

Mutually-exclusive sets of pins are utilized for different partitions. Therefore, if we can partition the array so that droplets are in different partitions, interference between them can be avoided. Partitions can be viewed as subarrays that can contain at most one droplet at any given time. Hence, the partitioning criterion here is to ensure at most one droplet is included in each partition. However, partitions with no droplets (at any point in time) should be avoided because no droplet manipulation is done in this region with the additional set of pins assigned to it. Hence it is best to ensure that each partition has exactly one droplet in it.

Based on this requirement, we find that the droplet trace, defined as the set of cells

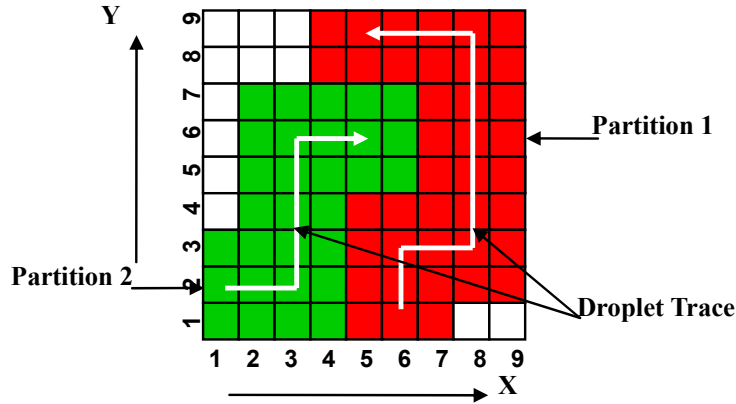
traversed by a single droplet, serves as a good tool for generating the array partitions. Since we view pin assignment as the last step in system synthesis, information about module placement and droplet routing is available *a priori*. The droplet trace can be easily extracted from the droplet routing information and the placement of the modules to which it is routed. A trace extraction example is shown in Figure 3.4, where two droplets are to be manipulated on the microfluidic array. Both of these are required to be detected by an optical sensor three times in a specific bioassay. The placement of these detectors is shown in Figure 3.4(a). The droplet routes, i.e., the path taken by droplets, are shown by the arrows in Figure 3.4(b). The connected arrows illustrate the traces of the two droplets. For each droplet, we create a partition composed of all the cells on its trace as well as the cells adjacent to the trace. The adjacent cells are included to form a “guard ring” along the trace to avoid inadvertent mixing and movement. The guard rings are a consequence of the fluidic constraint described in [27].

Note that in Figure 3.4(b), there are two “white” regions that belong to neither partition. They are referred to as “don’t-care” regions because they are similar to the “don’t-care” terms in logic synthesis; they can either be assigned to any partition or they can together form an additional partition if multi-droplet-operation modules, e.g. mixers, can be positioned in them.

In order to reduce the number of partitions, we introduce a time-division pin-sharing method. The basic idea is to merge partitions that have no overlapping time spans, where a time span for a partition is defined as the period of time during which it contains a droplet. The time spans for all the partitions can be easily calculated from the operation schedule, module placement and droplet routing results [15]; the overlaps can then be readily determined. Partitions with non-overlapping time spans are merged to form a larger partition. This check-merge procedure continues until all partition pairs overlap in their time spans. By reducing the number of partitions, we can reduce the number of control pins needed for the array. Note that droplet traces may have spatial overlap, i.e., they may intersect at one or

	Detector1(x,y)	Detector2(x,y)	Detector3(x,y)
Droplet 1	(8, 3)	(8, 9)	(5, 9)
Droplet 2	(3, 2)	(3, 6)	(5, 6)

(a)



(b)

Figure 3.4: (a) Detectors used in bioassay; (b) Routing result and array partitions.

more unit cells on the array. In this case, the requirement of one droplet per partition is not met and droplet interference may occur. This problem is handled by simply modifying the partitioning result.

We next study the case where droplets traces intersect on the array. This implies that partitions derived by the proposed method overlap in some regions. Sets of pins from an “overlapping” partition cannot be used in the overlapped region since the reuse of the pins may lead to droplet interference. One solution to this problem is to make the overlapping region a new partition, referred to as the overlapping partition, and use direct-addressing for it. Again, time-division pin-sharing (TDPS) can be used to reduce the number of pins since pin sets of the other (non-overlapping) partitions can be candidates for direct-addressing in the overlapping partition.

An example of this approach is shown in Figure 3.5. The droplet traces are first derived from the droplet routing information. Partitions 1, 2, 3, and 4 are assigned accordingly. Partition 2 and Partition 3 overlap with each other as shown. Thus a new Partition 23 is

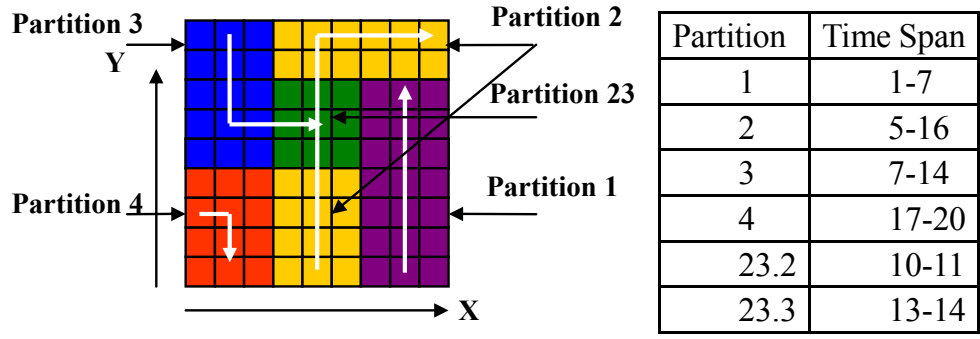


Figure 3.5: (a) Routing result and partitioning (b) Time-span table for the droplets.

created. From the scheduling result in Figure 3.5(b), the time span for Partition 23 is found to be 10-14s. Next the time spans for Partitions 1 and 4 are checked and it is seen that their time spans do not overlap with that for Partition 23. Hence the two sets of pins (a total of $2 \times 5 = 10$ pins) in Partitions 1 and 4 can be used to directly address the nine electrodes in Partition 23.

Partitions that share pins with the overlapping partition are empty while droplets are manipulated in the overlapping partition. Therefore, the sharing of pins in these cases does not lead to droplet interference. By introducing the concept of TDPS, we can significantly reduce the number of pins required for independent addressing in overlap partitions. The concept of TDPS can also be applied in the spatial dimension to the operations inside the overlapping region to further reduce the number of control pins.

Once a spatially overlapping region is found, we determine if there are temporally overlapping droplets in this region. Depending on the outcome of this procedure, a spatial overlap region can be then divided into two groups — a spatially overlapping but temporally non-overlapping (SOTN) region, and a spatially overlapping as well as temporal overlapping (SOTO) region. For SOTO regions, direct- addressing is used. For SOTN regions, even though droplets traces cross each other, different droplets are sequenced in time

Table 3.1: Time-span table with detailed scheduling results for the overlapping region.

Partition	Time Span
1	1-7
2	5-12
3	7-23
4	17-20
23	10-14

(one after the other), i.e., at any point in time, there is at most one droplet inside the region. In this case, a pin set with the minimum size ($k = 5$) for single droplet manipulation is assigned to this SOTN region.

Again, we use the above example of Figure 3.5 for illustration. Table 3.1 shows the schedule information needed for carrying out the temporal check for the overlapping region. Partitions 23.2 and 23.3 represent the manipulation of Droplet 2 and Droplet 3 in Partition 23 respectively. Table 1 shows that the time spans for these partitions do not overlap, thus five pins (in contrast to the nine pins needed for direct-addressing) are adequate for the overlapping partition.

3.1.3 Pin-Assignment Algorithm

In this subsection, we address the problem of how to map control pins to the electrodes in a partition. An efficient algorithm that can be easily implemented using a 3-layer-PCB is presented. The algorithm is based on a strategy of the Connect-5 (Gomoku) board game [61], thus it is referred to as the Connect-5 algorithm.

The sets of pins assigned to the partitions belong to two groups according to their cardinality, i.e., the minimum for single droplet manipulation ($k = 5$) or the number of pins required for direct-addressing. Here we focus on the pin assignment problem for the first case, since pin assignment for direct-addressing is straightforward (there exists a simple one-to-one mapping between pins and electrodes).

Our goal is to ensure that any five adjacent unit cells (i.e., a central cell and its four

neighbors) that form a “cross” are assigned distinct pins. We refer to the above constraint as the “cross constraint”. The pin assignment problem under cross constraints can be mapped to the well-known vertex coloring problem in graph theory [56]. The problem here is to obtain a 5-coloring of the graph derived from a partition, as shown in Figure 3.6. The unit cells in the partition are mapped to vertices and any two cells that belong to a “cross” are connected by an edge. The graph corresponding to a partition is referred to as the partition graph.

The graph coloring problem, which involves the determination of the chromatic number $\chi(G)$ for a graph G , is known to be NP-complete [57]. However, if $\chi(G)$ or the number of colors to be used is known, as is the case here, there exist an efficient algorithm for graph coloring. Moreover, the regular structure of the partitions can be used to solve the problem more efficiently using tiling. This approach allows us to use a regular distribution of pins, a layout feature that is not directly obtained from graph coloring. The tile (or template) used here is referred to as “Bagua”, a Chinese game strategy for the Connect-5 board game [61]. A Bagua is a tilted square, as shown in Figure 3.7. By repeatedly placing Bagua structures next to each other until the partition boundaries are reached, a Bagua repetition is derived as shown in Figure 3.7. The tiling using Bagua repetitions forms the basis for the Connect-5 algorithm.

Five copies of Bagua repetitions are sufficient to cover a partition of any size. This is because of the following property of a Bagua repetition: vertices connected to the same (shared) pin appear after exactly five cells in the same row or column of the partition. The partition can be covered with Bagua repetitions by simply taking a Bagua repetition and shifting it one cell along an arbitrary direction, e.g., upwards, then assigning it to another control pin and repeating this step four times, as shown in Figure 3.8. Note that, although the shifting direction is arbitrarily selected at the start of the tiling process, once chosen it must be consistent over the four shifting steps.

As shown in Figure 3.8, the pin assignment that results from the shifting of Bagua repetition satisfies a cyclic property, i.e., each row is a cyclic repetition of an ordered sequence,

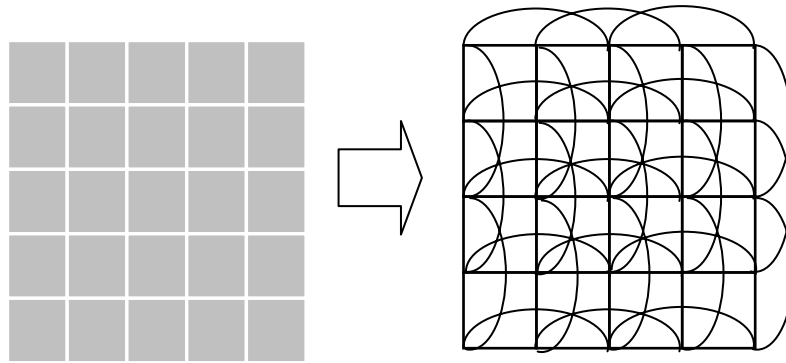


Figure 3.6: Mapping of an array to an undirected graph.

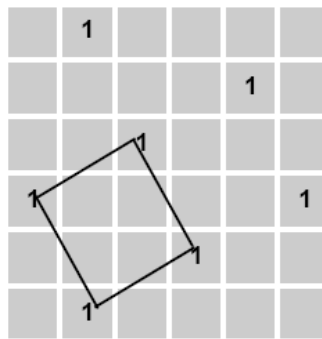


Figure 3.7: A single Bagua structure (the tilted square) and its repetition in a square partition.

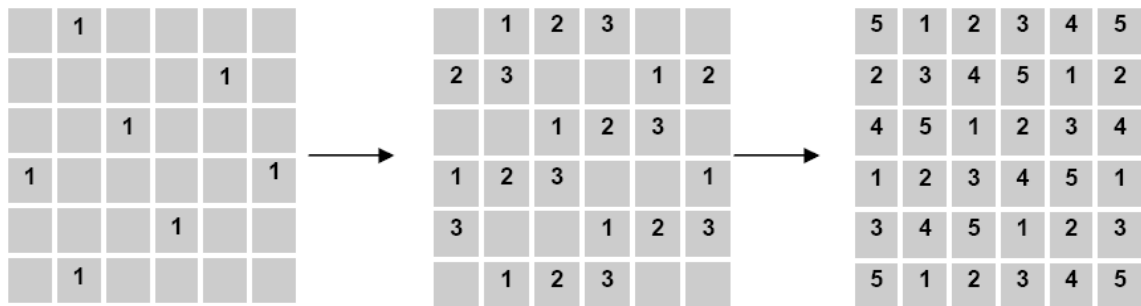


Figure 3.8: Covering a partition by shifting Bagua repetition along rows.

and it is also a shifted copy (shift by two cells) of the previous row. This cyclic property provides an easy way to implement the Connect-5 algorithm.

To start, the first row of a partition is selected. Pins are assigned in a fixed cyclic order until the boundary of the partition is reached. Then in the next row, the same order is used but with a 2-cell-shift to the left/right. The procedure continues until all cells in the partition have

been assigned pins. Recall that the shifting direction, once chosen, must remain fixed during the assignment procedure for a given partition.

Next we show that control pins assigned to the electrodes in a partition using this method allow free movement of a single droplet, i.e., the “cross constraint” is met. To demonstrate this, we consider the cell which is hatched in Figure 3.9. If the cell is assigned Pin 1, we cannot assign the same pin to the unit cells that are shaded. Otherwise, we will violate the cross constraint in some cases. It can be found that all the unit cells in the Bagua tile and its repetitions stay out of the forbidden area. Thus for each pin assigned to cells in a Bagua repetition, the cross constraint is not violated. Since this is true for any Bagua repetitions and any partition can be tiled by five copies of Bagua repetitions, the “cross constraint” is automatically met for every cell in our pin assignment method.

Compared to the graph coloring approach, the Connect-5 algorithm offers the important advantage that it allows wiring to be done easily on a 3-layer PCB; see Figure 3.10. The graph coloring approach does not lend itself to this simple pin layout because of the likelihood of irregular vertex coloring.

Connect-5 algorithm succeeds in avoiding droplet interference while moving a single droplet inside the partition. Recall that in the droplet-trace-based array partitioning method, partitions contains only one droplet each. Therefore, the Connect-5 algorithm can be integrated into the droplet-trace-based array partitioning method to generate droplet-interference-free layouts with a minimum number of pins.

3.1.4 Application to Multiplexed Bioassay

To evaluate the array partitioning and pin assignment method for pin-constrained microfluidic biochips, we use a real-life experiment of a multiplexed biochemical assay consisting of a glucose assay and a lactate assay based on colorimetric enzymatic reactions. These assays have been demonstrated recently [38]. The digital microfluidic biochip contains a 15×15 microfluidic array, as shown in Figure 3.11. The schedule for the set of bioassays, if a

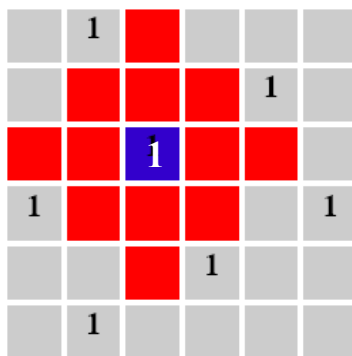


Figure 3.9: A demonstration that the “cross constraints” are met.

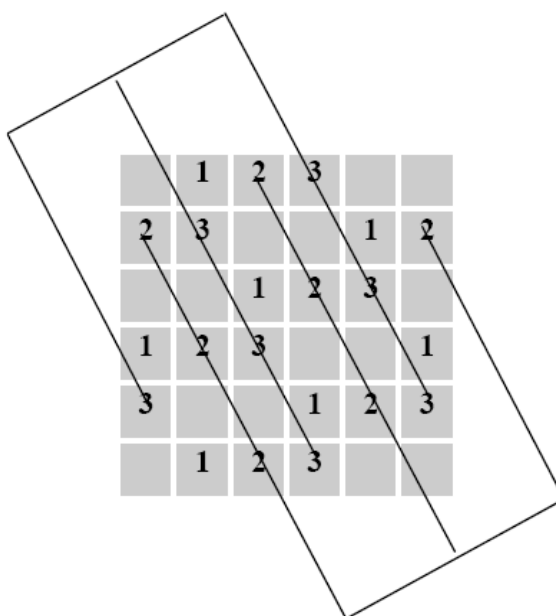


Figure 3.10: A wiring example for the pin assignment obtained using the Connect-5 algorithm. For each partition, two pins can be wired in one layer.

microfluidic array with 225 control pins is available, is listed in Table 2; one iteration of the multiplexed assays takes 25.8 seconds [38]. The movement of droplets is controlled using a 50 V actuation voltage with a switching frequency of 16 Hz. A depiction of the droplet paths for multiplexed glucose and lactase assays is shown in Figure 3.11.

We next apply the droplet-trace-based array partitioning method to the multiplexed bioassay example. Initially, six partitions are created for the four droplet traces of Reagents 1, 2 and Samples 1, 2, and the two traces corresponding to the mixed samples going to Detector

Table 3.2: Bioassay schedule for a full-addressable array.

Step/Time (seconds)	Elapsed	Operation
Step 1 / 0		Sample 2 and reagent 2 start to move towards the mixer.
Step 2 / 0.8		Sample 2 and reagent 2 begin to mix together and turn around in the 2x3-array mixer.
Step 3 / 6.0		Sample 1 and reagent 1 start to move towards the mixer. Sample 2 and reagent 2 continue the mixing.
Step 4 / 6.8		Sample 2 and reagent 2 finish the mixing and product 2 leaves the mixer to optical detection location 2. Sample 1 and reagent 1 begin to mix in the 2x3-array mixer.
Step 5 / 12.8		Sample 1 and reagent 1 finish the mixing and product 1 leaves the mixer to the optical detection location 1. Product 2 continues the absorbance detection.
Step 6 / 19.8		Product 2 finishes optical detection and leaves the array to the waste reservoir. Product 1 continues the absorbance detection.
Step 7 / 25.8		Product 1 finishes optical detection and leaves the array to the waste reservoir. One procedure of the multiplexed bioassays ends.

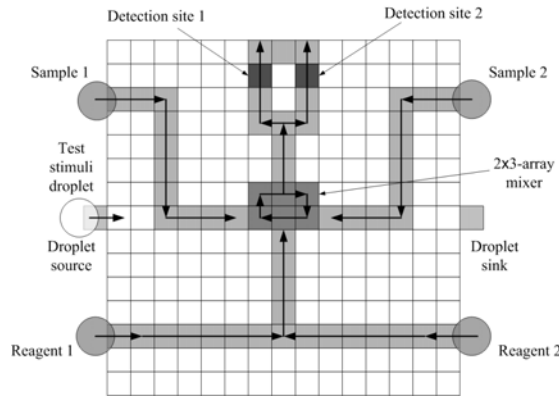


Figure 3.11: A 15x15 array used for multiplexed bioassays.

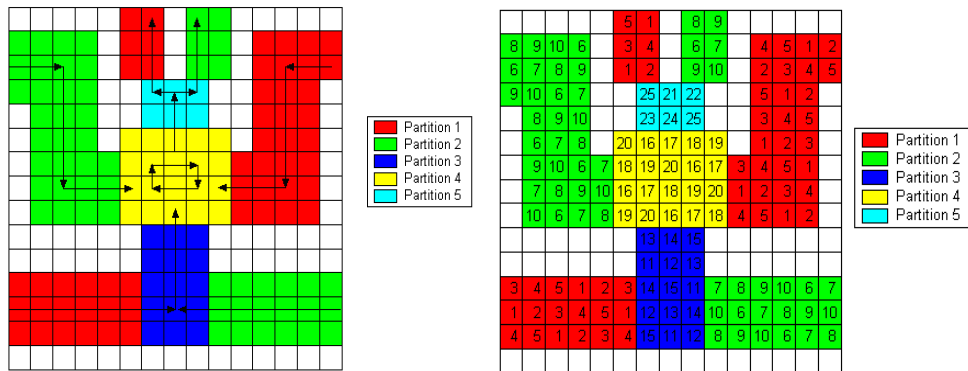


Figure 3.12: Partition and pin assignment result of the multiplex bioassay. Blank areas are don't care regions that can be either left unaddressed or combined with any partition.

1 and Detector 2. Three more partitions are created for the three trace-overlapping regions respectively. Next, time-span overlap is checked for the three spatially overlapping partitions (Partitions 3, 4 and 5). Since there is no temporal overlap of droplets being manipulated in both Partition 3 and Partition 5, only five pins are needed for each of them. Partition 4 is recognized as a mixer, thus only five pins are needed for it. In the next step, time span overlap is checked for all partition pairs. The six partitions corresponding to four droplets traces and two detector paths merge into two partitions (Partition 1 and Partition 2). Finally, the Connect-5 algorithm is applied. The partitions and the pin-assignment results are shown in Figure 3.12.

We therefore see that array partitioning and pin assignment are effective in reducing the input bandwidth, while maintaining the same throughput that is obtained for a directly-addressable array. Five partitions are sufficient for preventing interference between multiple droplets on the array, as shown in Figure 3.12. Since only five control pins are necessary for full control of a single droplet within each partition, only 25 out of the possible 225 control pins are necessary, i.e., only 11.11% of the total number of electrodes. This represents a significant reduction in input bandwidth without sacrificing throughput.

3. 2 Cross-Referencing-Based Droplet Manipulation Method

In this section, we represent an alternative implementation of pin-constrained biochips. We propose a droplet manipulation method based on a “cross-referencing” addressing method that uses “row” and “columns” to access electrodes [62]. By mapping the droplet movement problem to the clique partitioning problem from graph theory, the proposed method allows simultaneous movement of a large number of droplets on a microfluidic array. This in turn facilitates high-throughput applications on a pin-constrained biochip.

3.2.1 Cross-Referencing Addressing

Pin-constrained biochip based on “cross-referencing” addressing method is first proposed in [39]. This method allows control of an $N \times M$ grid array with only $N+M$ control pins. The electrode rows are patterned on both the top and bottom plates, and placed orthogonally. In order to activate a cell on the grid array, the electrode row and column that the cell belongs to are selected, as shown in Figure 1.3. This cross-reference method facilitates the reduction of control pins. However, due to electrode interference, this design cannot handle the simultaneous movement of more than two droplets. The resulting serialization of droplet movement is a serious drawback for high-throughput applications.

As discussed in Section 1.2, the minimization of the assay completion time, i.e., the maximization of throughput, is essential for applications such as environmental monitoring, surgery and neo-natal clinical diagnostics.

3.2.2 Power-Efficient Interference-Free Droplet Manipulation Based on Destination-Cell Categorization

In this subsection, we focus on the problem of manipulating multiple droplets on based digital microfluidic biochips that use cross-referencing to address the electrodes.

Electrode Interference

For the concurrent manipulation of multiple droplets on a cross-referencing-based biochip, multiple row and column pins must be selected to activate the destination cells, i.e., cells to which the droplets are supposed to move. However, the selected row and column pins may also result in the activation of cells other than the intended droplet destinations. An example is shown in Figure 3.13. The goal here is to route Droplets 1, 2, 3 simultaneously to their destination cells. Droplet 4 is supposed to remain in its current location. However, two additional cells are activated unintentionally when the activation voltage is applied to the row

and column pins corresponding to the destination cells. As a result, Droplet 4 is unintentionally moved one cell up (along the Y-direction).

Fluidic Constraints

Droplet manipulations must also conform to rules referred to as the fluidic constraints. These constraints are given by a set of inequalities, as shown in Section 3.1.1.

Destination-Cell Categorization

As shown in Figure 3.13, the concurrent manipulation of multiple droplets must be carried out without introducing any electrode interference. For simplicity, here we only focus on the implementation of a set of multiple droplet manipulations that can be carried out concurrently (in a single routing step) on a direct-addressing-based chip, without violating any fluidic constraints. We refer to such a set of droplet manipulations as a *droplet-manipulation snapshot*.

We propose a solution based on destination-cell categorization. Note that the problem highlighted in Figure 3.13 can be avoided if the destination cells of the droplets being moved simultaneously reside on the same column or row. However, electrode interference may still occur within the same column or row, as shown in Figure 3.14. Suppose Droplet 1 and Droplet 2 are both moved one cell to the left at the same time. Even though no additional cells are activated unintentionally, Droplet 1 undergoes unintentional splitting in this situation. Fortunately, further scrutiny reveals that the situation in Figure 3.14 is only a false alarm. The intended multiple droplet manipulation violates the constraint $|P_i(t+1) - P_j(t)| \geq 2$. Such manipulations cannot be carried out concurrently even on a direct-addressing-based chip. Thus they will never appear in a single droplet-manipulation snapshot. Therefore, it is safe to carry out concurrent manipulation of multiple droplets whose destination cells are accessed by the same column or row.

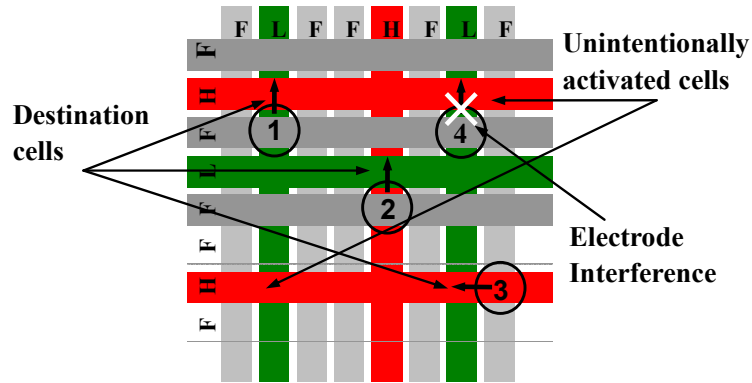


Figure 3.13: An example to illustrate the problem of electrode interference. H/L stands for high/low voltage pairs to activate the cells, and unselected row/column pins are left floating (F).

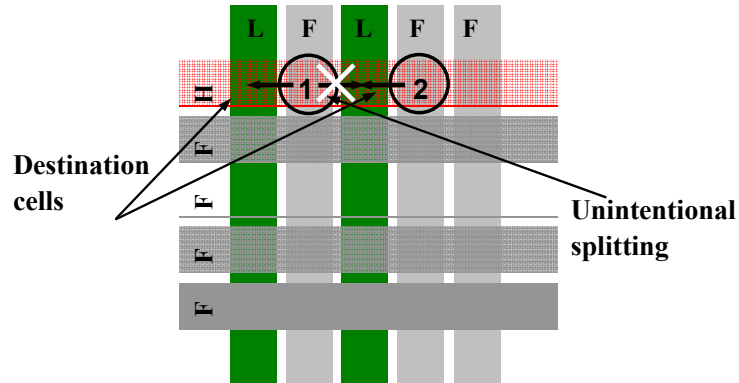


Figure 3.14: An example of electrode interference within the same row.

On the basis of the above observations, we consider the droplets that can be moved simultaneously as part of the bioassay, and place them in different groups. A group consists of droplets whose destination cells share the same column or row. An example is shown in Figure 3.15. A total of nine droplets are needed to be moved on a 10×10 array. As discussed above, we group the droplet movements according to their destination cells. For example, Droplets 4 and 9 from a group since the destination cells in both cases resides on Row 2. Similarly Droplets 1, 2, and 3 are placed in the same group since they are all moving to Column 3. Following this grouping process, we finally get four groups of droplets, i.e., {4,9}, {1,2,3}, {5,6}, {7,8}. In this way, the manipulation of multiple droplets is ordered in time;

droplets in the same group can be moved simultaneously without electrode interference, but the movements for the different groups must be sequential. For example, droplet movements for the group {4,9} in Figure 3.15 can be carried simultaneously, as shown in Figure 3.16. Droplet movements are carried out one group after another until all the droplet movements are completed.

Note that the ordering of droplet movements based only on the above grouping strategy can cause electrode interference and inadvertent mixing. An example is shown in Figure 3.17. The movement of Droplet 2 alone to the left by activating Column 3 will not influence Droplet 1. Similarly, the movement of Droplet 1 alone to the right by activating Column 2 will not influence Droplet 2. However, if these two droplets are moved concurrently, as determined by the grouping procedure, by the activation of (Column 2, Row 2) and (Column 3, Row 2), they mix at (3,2). However, manipulations of this type violate the fluidic constraint given by $|P_i(t+1)-P_j(t+1)| \geq 2$. Thus they cannot exist in a single droplet-manipulation snapshot. Therefore, it is safe to carry out the droplet manipulations in a single manipulation snapshot with an arbitrary ordering.

Although the grouping of droplets based on destination cells reduces the number of droplets that can be simultaneously moved, this approach provides more concurrency than the baseline method of moving one droplet at a time. Compared to direct-addressing, an order of magnitude reduction in the number of control pins is obtained. Simulation results in Section 3.2.5 show that there is only a small increase in the bioassay processing time compared to direct-addressing. The above droplet-manipulation method is focused on minimizing power consumption because, in each step, only droplet manipulations that involve a single column or row are carried out. Additional droplet movements are typically possible, but concurrency is traded-off for power in this method. An extension to allow higher concurrency is described in Section 3.2.4.

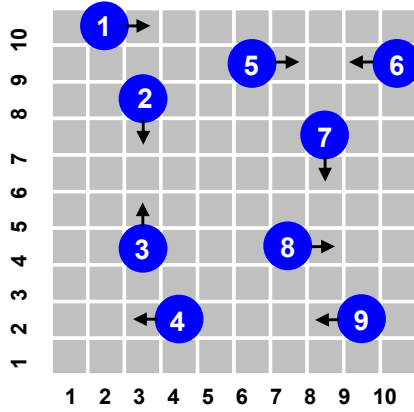


Figure 3.15: Example to illustrate destination-cell-based categorization.

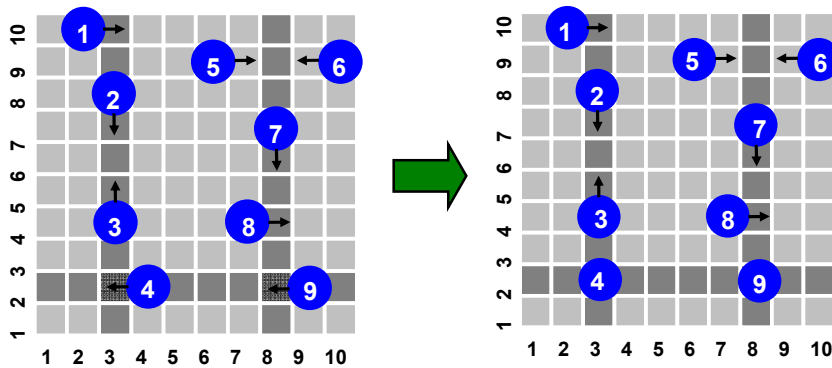


Figure 3.16: Example to illustrate the concurrent movement of a group of droplets.

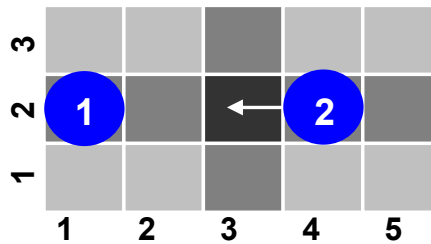


Figure 3.17: Example of electrode interference due to asynchronous processing of multiple droplet manipulations.

Graph-Theoretic Model and Clique Partitioning

We have thus far introduced the basic idea of multiple droplet manipulations based on destination-cell categorization, and shown that the droplets in each group can be moved simultaneously. Assuming that each step takes constant processing time, the total completion

time for a set of droplet movement operations is determined by the number of groups derived from the categorization of destination cells. Note however that the grouping need not be unique. For instance, in the example of Figure 3.15, we can form four groups, i.e., $\{4,9\}$, $\{1,2,3\}$, $\{5,6\}$ and $\{7,8\}$. However, $\{1,2,3,4\}, \{5,6\}, \{7,8,9\}$ is also a valid grouping of the droplets. The latter grouping is preferable because three groups allow more concurrency, and therefore lower bioassay completion time.

The problem of finding the minimum number of groups can be directly mapped to the clique partitioning problem from graph theory [56]. To illustrate this mapping, we use the droplet manipulation problem defined in Figure 3.15. Based on the destinations of the droplets, an undirected graph, referred to as the droplet movement graph, is constructed for each time-step; see Figure 3.18. Each node in the droplet movement graph represents a droplet. An edge in the graph between a pair of nodes indicates that the destination cells for the two droplets either share a row or a column. For example, Nodes 1 and 2, which represent the Droplet 1 and Droplet 2, respectively, are connected by an edge because the destination cells for these droplets are accessed using Column 3 in the array. Similarly, Nodes 4 and 9 are connected by an edge because the corresponding destination cells are addressed using the same row.

A clique in a graph is defined as a complete subgraph, i.e., any two nodes in this subgraph are connected by an edge [56]. Clique partitioning refers to the problem of dividing the nodes into overlapping subsets such that the subgraph induced by each subset of nodes is a clique. A minimal clique partition is one that covers the nodes in the graph with a minimum number of non-overlapping cliques. The grouping of droplets as discussed above is equivalent to the clique partitioning problem. The categorization of destination cells using the grouping of droplets is equivalent to the problem of determining a minimal clique partition. Cliques of different sizes for a given droplet movement graph are shown in Figure 3.18. A minimal clique partition here is given by $\{1,2,3,4\}, \{5,6\}, \{7,8,9\}$, which corresponds to the groups

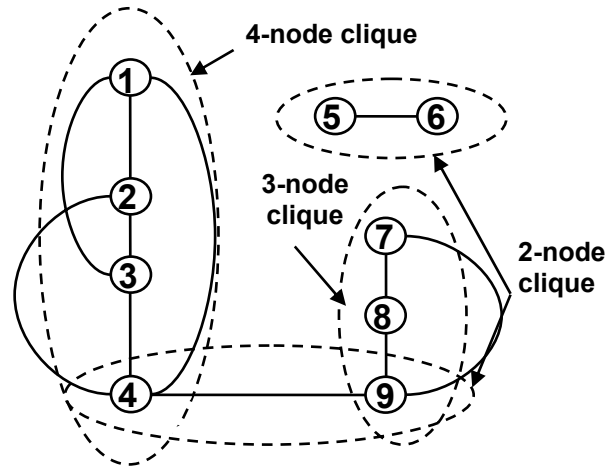


Figure 3.18: Mapping destination cell layout to undirected graph.

derived above. Even though the general clique partitioning problem is known to be NP-hard [57], a number of heuristics are available in the literature to solve it in an efficient manner.

Algorithm for Droplet Grouping

Next we describe a greedy algorithm to determine a (minimal) clique partition for the droplet-movement graph (DMG). The algorithm determines cliques for the DMG in an iterative manner.

The largest clique is first determined and then nodes and edges corresponding to this clique are deleted from the graph. Next, the clique searching procedure is applied to the reduced graph. The algorithm terminates when all the nodes in the DMG have been deleted, i.e., an empty graph is obtained. The computational complexity of this algorithm is linear in the number of rows/columns. Recall that the cliques can only be formed among nodes sharing the same row or column. Therefore, the largest clique can be determined by scanning the columns and rows of the array. Thus a maximum of only $N+M$ iterations are needed for the droplet-movement graph derived from an $N \times M$ array.

Note that even though in each step of the above algorithm, the largest clique and the associated destination cells are deleted, the absence of the corresponding destination cells does not lead to any added complexity for droplet movement. This is because the droplet

movements involving these destination cells are incorporated in the clique determined at this step. Therefore, when the algorithm terminates with an empty graph, all droplet movements have been processed without any electrode interference.

The steps of the complete procedure to determine the order of droplet movements can be stated as follows:

1. Obtain the required droplet movements (from a synthesis tool such as [15]), and organize these movements in the form of snapshots corresponding to different time-steps.
2. Compare consecutive snapshots to determine the destination cells for the droplets.
3. Scan each row and each column to find the row/column with the largest set of destination cells. The destination cells thus determined forms a group of droplets that can be simultaneously moved. If no row/column contains more than one destination cells, set the flag END to 1.
4. If $END = 1$, process the remaining movements in multiple steps, but with two droplets at each step. Else carry out the droplet movements indicated by Step 3.
5. Check if all the movements in the snapshot have been processed. If the check yields a negative outcome, repeat Step 3.
6. Check whether all the snapshots are processed. If not, get the next snapshot and repeat Step 2, else terminate the procedure.

3.2.3 Scheduling of Routing for Efficient Grouping

The column- and row-scan methods described above enable the simultaneous manipulation of multiple droplets on the cross-referencing chip. However, the efficiency of this approach depends on the pre-alignment of the destination cells corresponding to the droplet movements in the target droplet-routing snapshot. The better aligned the destination cells are, i.e., they share the same column/row, the larger is the number of droplets that can be moved simultaneously. Therefore, to increase efficiency, it is important to generate routing snapshots

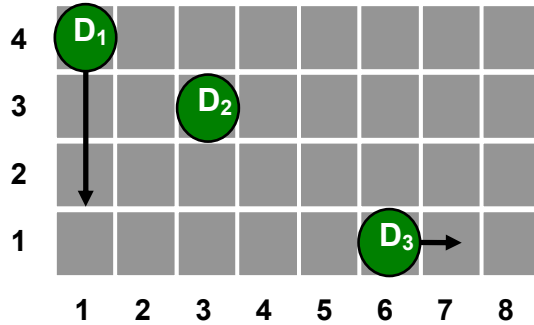


Figure 3.19: An example of a droplet-routing plan.

with well-aligned destination cells.

Note that routing snapshots are obtained from the schedule of droplet movements corresponding to the droplet-routing plan. Typically, several schedules are feasible for a given droplet-routing plan. For example, the routing plan shown in Figure 3.19, derived from routing methods such as [27], can be scheduled, i.e., implemented, in two ways; see Figure 3.20 and Figure 3.21.

Using the grouping algorithm described in Section 3.2.2, the three snapshots in Figure 3.20 take five manipulation steps in total (three steps for Snapshot 1, one step for Snapshot 2, and one step for Snapshot 3). However, the three snapshots in Figure 3.21 only require three manipulation steps (one step for each). Therefore, the key to obtaining a well-aligned snapshot is the schedule of droplet routes. Compared to the schedule in Figure 3.20, the schedule in Figure 3.21 carefully orders the droplet movements. Droplets movements are scheduled in such a way that each snapshot brings the droplets closer to each other, therefore their destination cells are more likely to share the same column/row. By this means, the alignment of the droplets for each snapshot is improved, which explains the increase in routing efficiency.

Based on the above observation, we present an efficient scheduling algorithm to generate well-aligned snapshots. The pseudo-code for this algorithm is sketched in Figure 3.22.

We next use an example to illustrate the algorithm. Figure 3.23 shows a routing plan with

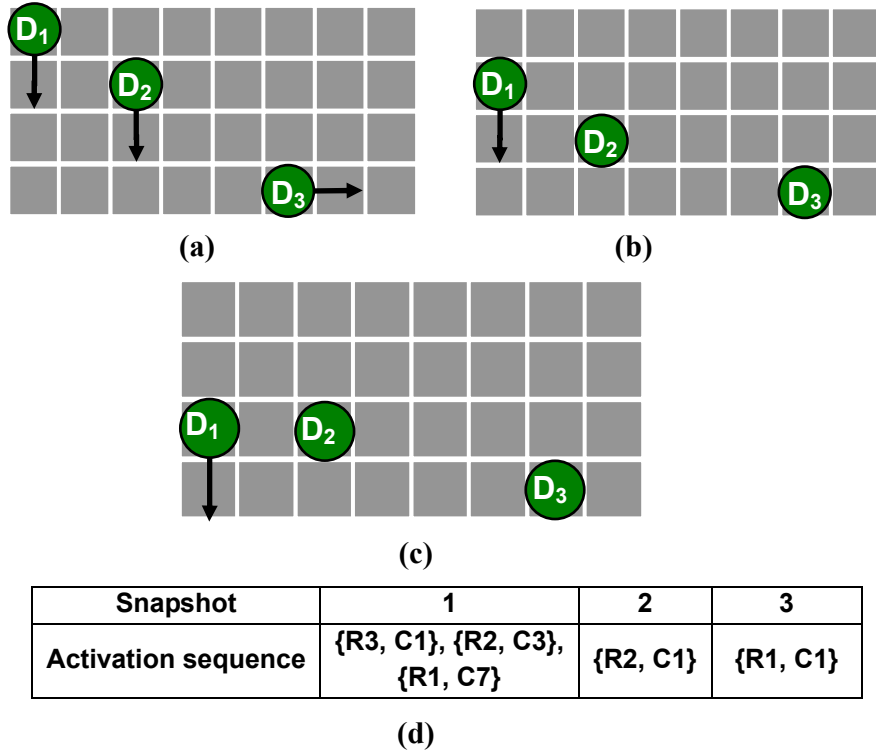


Figure 3.20: Schedule A: (a) Snapshot 1; (b) Snapshot 2; (c) Snapshot 3; (d) Activation sequence (in terms of rows and columns).

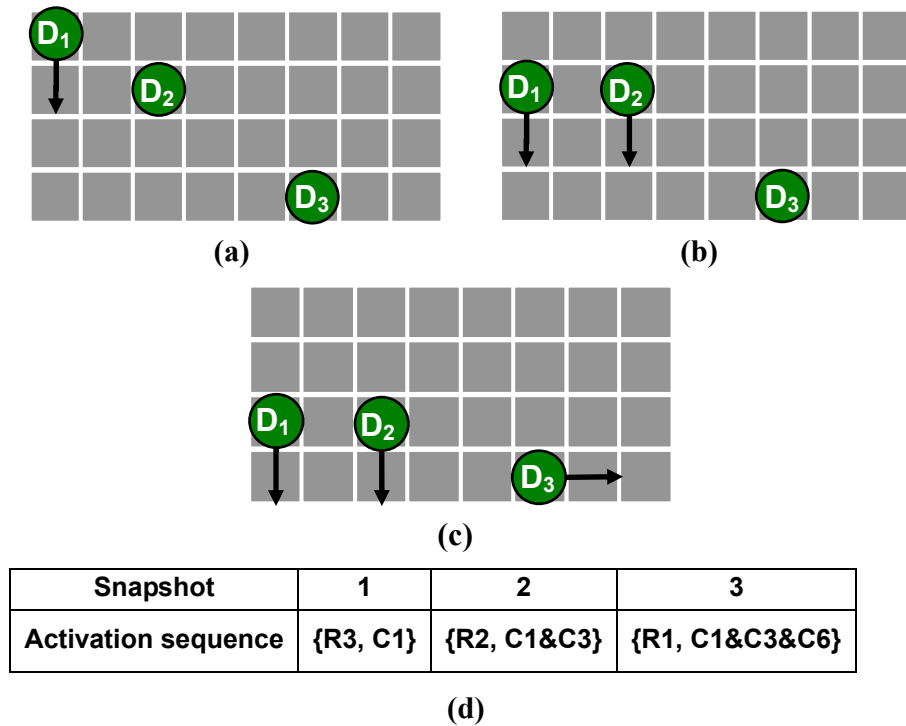


Figure 3.21: Schedule B: (a) Snapshot 1; (b) Snapshot 2; (c) Snapshot 3. (d) Activation sequence (in terms of rows and columns).

Scheduling algorithm for droplet routing

- 1 Start from the droplet closest to the boundary of the array, whose initial moving direction facing the inner of the array. Mark its initial moving direction as the *starting direction*. We chose such a droplet as starting point since it is most likely to be far away from all the other droplets.
 - 2 Move this droplet along its pathway until it shares the same column/row with the starting point of another droplet moving in the same direction. Move the two droplets in parallel. By this means, the farthest droplet is “pulled” closer to other droplets, thereby improving the alignment and increasing the probability of concurrent manipulation.
 - 3 Repeat Step 2 and keep adding new droplets for parallel movement until a droplet in the set of droplets moving in parallel confronts either a turn or another droplet in its pathway.
 - 4 Count the number of droplet movements oriented in each of the three directions other than the reverse of the starting direction at the current column or row, and store these numbers in a candidate moving set S .
 - 5 Choose the direction of the next droplet manipulation corresponding to the largest number of unfinished droplet movements in S .
 - 6 Repeat Steps 2-5 until no droplet can be moved in the three directions other than the reverse of the starting direction. The reverse direction of the starting direction is excluded in this step because allowing droplet moving in this direction may cause dead loop. Therefore, we handle such movement separately in Step 8.
 - 7 Repeat Steps 1-6 starting from the reverse of the starting direction. This step schedules the leftover droplet movements from Step 6.
 - 8 Add the routing schedule from Step 7 to the end of the schedule obtained from Steps 1-6 in the first iteration to obtain the schedule for the entire routing snapshot.
-

Figure 3.22: Pseudo-code of scheduling algorithm for droplet routes.

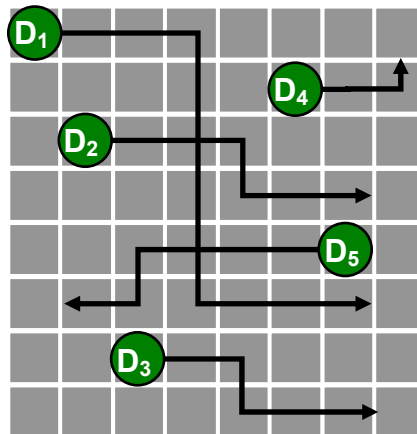


Figure 3.23: A routing example to illustrate scheduling algorithm.

five droplet pathways. Initially the droplets reside at the starting points shown in Figure 3.23. The algorithm first moves D_1 one electrode to the right. Next droplets D_1 and D_2 are moved one electrode to the right in parallel. D_3 is added and moved one electrode to the right with D_1 and D_2 . At this time, D_1 meets a turning point in its pathway. The algorithm stops and counts the number of droplet movements in each direction, i.e., two rightwards, one downwards. Therefore, the algorithm stores these numbers in S and keeps on moving D_2 and D_3 until they reach their respective turning points. Next the algorithm moves them one electrode downwards and one electrode to the right when they meet D_4 . Then D_2 , D_3 and D_4 are moved one electrode to the right.

Next the algorithm goes back to move the leftover droplets D_1 and D_4 . Finally, the algorithm generates a routing schedule for D_5 and then integrates it into the schedule for droplets $D_2 \sim D_4$.

Next we evaluate the computational complexity of the algorithm using step by step analysis. We assume that there are M droplets on an array of $N \times N$ electrodes. Droplet i has a pathway of length L_i . As shown in Figure 3.22, Step 1 takes $O(M)$ time to determine the starting droplet. Step 2 and Step 3 check if there are multiple droplets that can be moved concurrently after each manipulation step. Each checking operation takes $O(M)$ time. Step 4 calculates the number of droplet movements in different directions and stores this information in S . This step takes $O(M)$ time in the worst case. Step 5 also takes $O(M)$ time. Note that Steps 2-5 can be repeated. However, the number of repetitions is bounded by the sum of the lengths of all the droplet pathways, i.e., in the worst case, Steps 2-5 are repeated for each single droplet movement. Therefore, the time taken by Steps 2-5 is simply $(M + M + M) \sum_{i=1}^M L_i = O(M \sum_{i=1}^M L_i)$. Note that $L_i \leq 2N$, where $1 \leq i \leq M$. Thus, $\sum_{i=1}^M L_i \leq 2MN$. We therefore conclude that the worst-case time complexity for Steps 2-5 is $O(NM^2)$. The time complexity of Step 1 is $O(N)$. For Steps 7-8, we also have $O(NM^2)$, therefore, the overall time complexity of the scheduling algorithm is $O(NM^2)$.

3.2.4 Variant of Droplet-Manipulation Method for High-Throughput Power-Oblivious Applications

For applications where power consumption is not critical, the method proposed in the previous section can be modified to achieve even higher throughput.

Note that when we use the low-power manipulation method of Section 3.2.3 to implement the “aligned” droplet manipulations, in each step, only droplet manipulations corresponding to a single row or column are carried out. Note however that there may be other droplet manipulations that can also be implemented without introducing electrode interference. For example, for the droplet manipulation pattern shown in Figure 3.15, manipulation of D_7 and D_8 can also be carried out at the same time when we concurrently move D_4 and D_9 .

The implementation of these “compatible” droplet manipulations will result in higher power consumption, with the associated benefit of higher throughput. Based on this observation, we propose a modified droplet-manipulation method that relies on the method from [63] to carry out the droplet manipulation for the routing plans generated from the proposed routing-scheduling method. Note that the straightforward application of [63] to the droplet routes from derived from the routing-scheduling algorithm leads to undesirable consequences. The alignment of the droplet movements will be broken, thereby leading to lower throughput. Therefore, we limit the use of [63] for handling droplet movements in Step 7 of the routing scheduling method, i.e., the ones that correspond to the reverse of the starting direction. Instead of being carried out using additional steps, these droplet manipulations are carried out concurrently with the ones from Step 1-6. This approach results in higher throughput and reduced assay completion time.

3.2.5 Simulation Results

In this subsection, we use random synthetic benchmarks and a set of multiplexed bioassays to evaluate the proposed method.

Random Synthetic Benchmarks

We first use random synthetic benchmarks to evaluate the effectiveness of the grouping-based droplet-movement approach. Digital microfluidic arrays of size $N \times N$, ($N = 25, 50, 75$) are considered here. For each array, we consider 1000 simulated droplet-movement plans. Each droplet-movement plan is defined by a starting snapshot and destination snapshot. The starting snapshot is generated by injecting a droplet in the array with probability k , referred to as the droplet injection probability (DIP). A check is incorporated in the generation process to avoid the violation of fluidic constraints. Results derived from this process can be viewed as snapshots of droplets moving around the chip. Each droplet-movement plan is provided as input to the grouping-based method and the number of steps required for droplet movement is calculated. One-at-a-time droplet movement is also considered as a baseline and the results are recorded for the purpose of comparison.

To evaluate the proposed method, we introduce the parameter “number-of-steps-ratio” (NSR), defined by the equation $NSR = N_p/N_o$, where N_p (N_o) is the number of movement steps for the grouping-based method (one-at-a-time baseline method). Small values of NSR are clearly desirable. We calculate the NSR values for different array sizes and the results; see Table 3.3.

As shown Table 3.3, regardless of DIP value, the NSR decreases with array size. This shows that the grouping-based method is more efficient for concurrent droplet manipulation on large-scale digital microfluidic arrays. For a given array size, the proposed method achieves lower NSR values for higher values of DIP. Thus we see that compared to the one-at-a-time scheme, droplets can be manipulated more efficiently for high-throughput biochips with higher concurrency in biochip operations.

A Multiplexed Bioassay Example

Next, we evaluate the proposed scheduling and grouping-based droplet-manipulation

Table 3.3: Random synthetic benchmarks, sample size = 1000.

DIP	Array Size	NSR
0.1	25×25	0.31
0.1	50×50	0.24
0.1	75×75	0.19
0.15	25×25	0.28
0.15	50×50	0.20
0.15	75×75	0.14

methods by using them to implement the routing plan for a set of real-life bioassays, namely multiplexed *in-vitro* diagnostics on human physiological fluids.

As a typical example of multiplexed and concurrent assays, three types of human physiological fluids, urine, serum and plasma are sampled and dispensed into the digital microfluidic biochip, and glucose and lactate measurements are performed for each type of physiological fluid. The assay protocol, based on Trinder’s reaction, can be modeled by a sequencing graph, as shown in Figure 3.24. We assume that the schedule for assay operations and resource binding have been obtained via architectural level synthesis (e.g., through the modified list-scheduling algorithm [16]), as shown in Figure 3.25. Note that one time unit in this schedule is set to 2 seconds. Moreover, assume that a module placement on a 16×16 microfluidic array has also been given *a priori*, as shown in Figure 3.26.

To find suitable droplet routes for this biochip, the routing problem is decomposed into eleven sub-problems, as highlighted in Figure 3.26. For each sub-problem, a droplet routing plan is generated using the routing algorithm in [27]. Next we apply the proposed grouping-based manipulation method to implement the routing plan. For comparison, the grouping-based manipulation method is first applied without the scheduling algorithm for aligning droplets. The snapshots are generated by assuming that from the starting point, each droplet moves one electrode along its pathway in every snapshot. Note that fluidic constraints might be violated if two droplets simultaneously move to the next cells. In this case, we can force one of them to stay in the current location at that snapshot, thereby overriding the constraint violation. We calculate the number of manipulations cycles needed using the

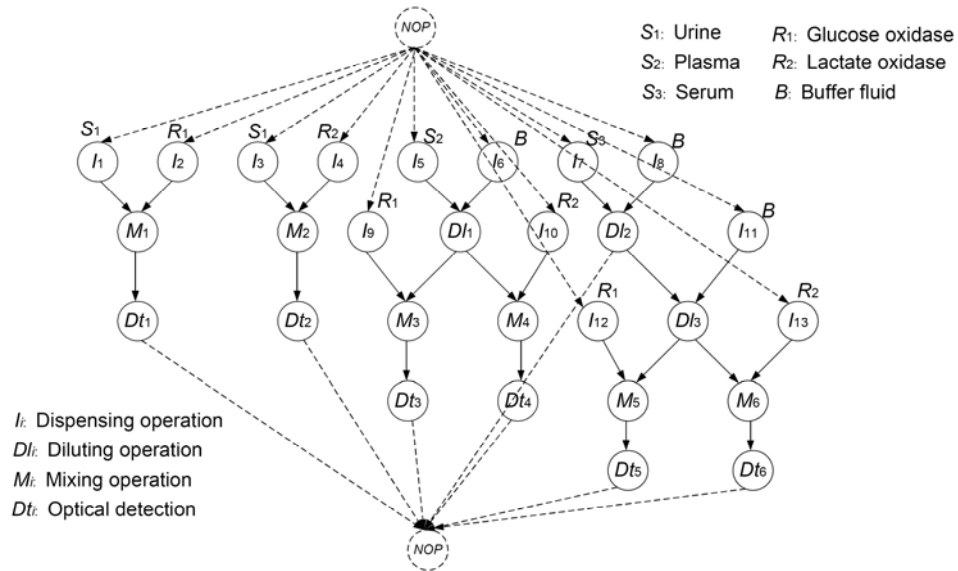


Figure 3.24: Sequencing graph model of bioassay example.

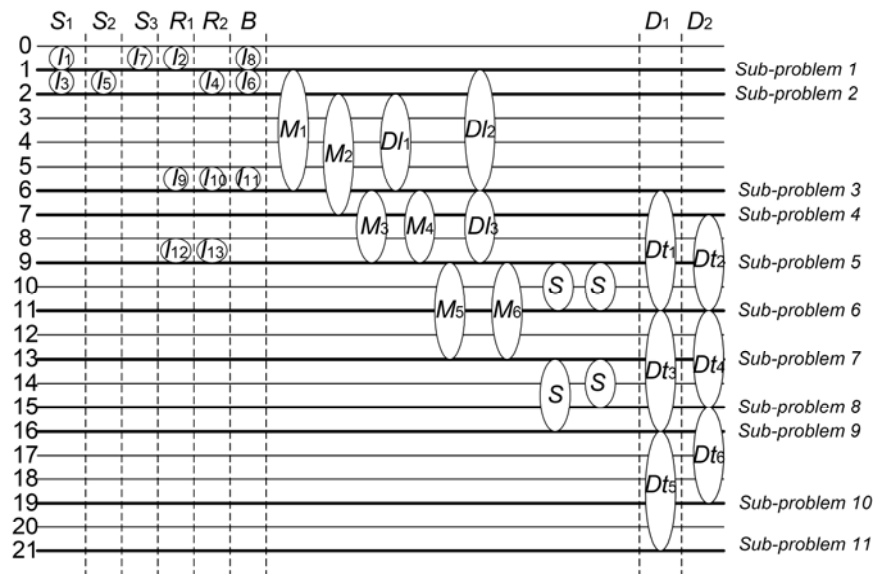


Figure 3.25: Schedule obtained via architectural-level synthesis.

grouping-based method. Next, we apply the manipulation method combined with the scheduling algorithm. Again, we record the number of droplet-manipulation manipulation cycles required. The computation time for the routing scheduling and the manipulation method for the entire assay is 173 seconds, on a Intel Core Duo 2 GHz PC with 2G of RAM.

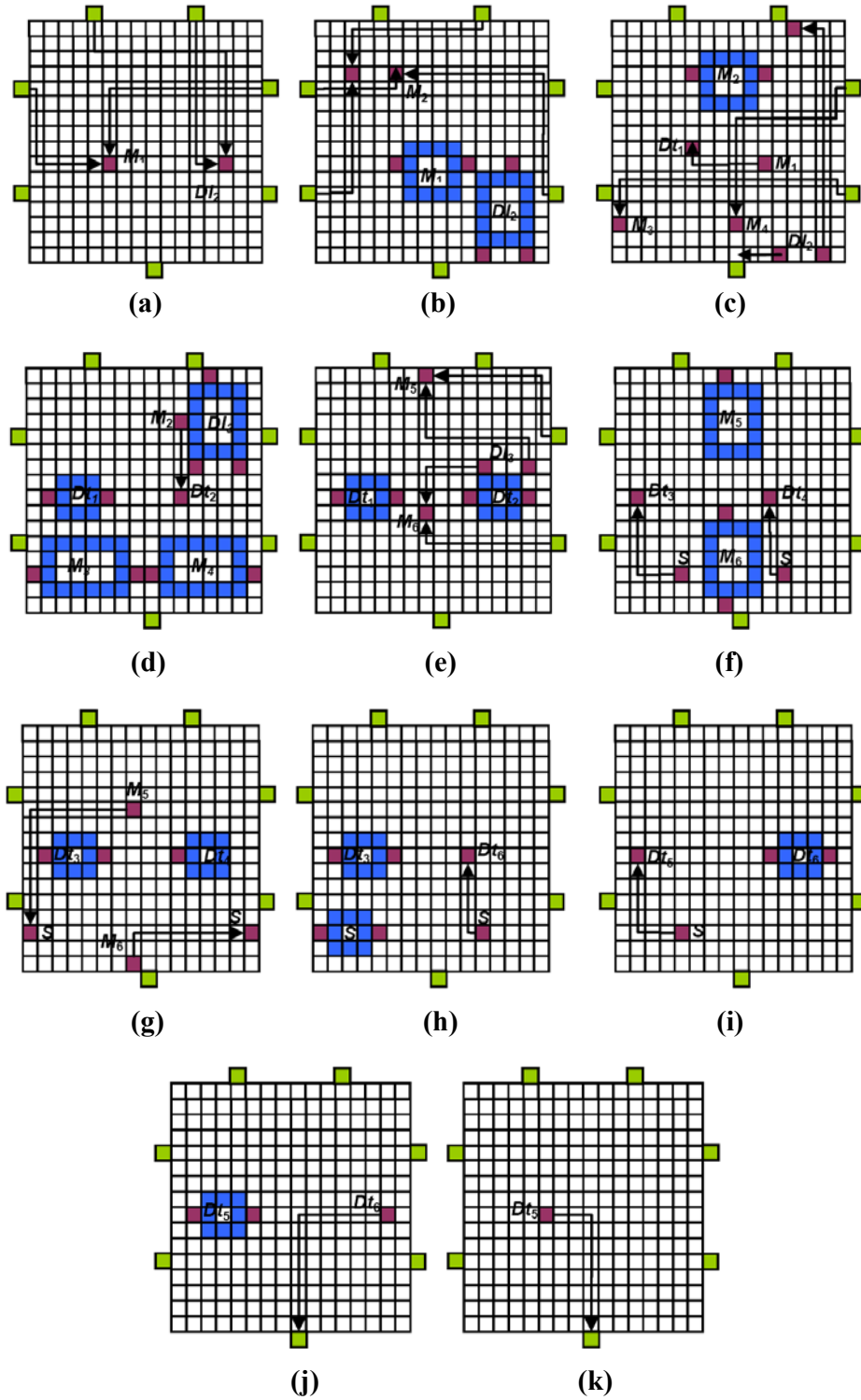


Figure 3.26: Module placement (11 sub-problems) for the multiplexed bioassay.

Here we use Sub-problem 3 for illustration. As shown in Figure 3.27, the routing plan for Sub-problem 3 contains five routing pathways (routes). If no grouping method is used, droplet movements are carried out one per cycle. The total number of cycles needed equals the number of electrode on all the pathways, i.e., 60. However, using the grouping-based manipulation method, the number of cycles is reduced to 53 cycles. Note that no scheduling of droplet routes is used at this time to align droplets. The details of the manipulations in each cycle are listed in Table 3.4, where the arrows indicate that the target droplet is moved one electrode toward the corresponding direction. An entry “x” indicates that the droplet stays in the current location in that snapshot.

In Table 3.4, two droplets are moved simultaneously in several manipulation cycles. This increases the currency of droplet movements. However, due to the severe misalignment of the snapshots, the number of such concurrent manipulation cycles is rather limited. Therefore, the reduction of routing time is quite modest, less than 12%.

Next, we apply a combination of the proposed cross-based scheduling method and grouping-based droplet-based manipulation to the routing plan. The results are shown in Table 3.5. Compared to the results shown in Table 3.4, droplet-movement concurrency is significantly improved. Half of the cycles now contain the concurrent manipulation of more than two droplets. In some cycles, even four droplets are moved simultaneously. As a result, only 34 cycles are required for this routing plan, which is only 57% of the time required for one-at-a-time droplet, and 64% of the time required if droplet grouping is carried out without route scheduling.

Note that the completion time obtained using the proposed droplet-manipulation method is slightly more than that for the direct-addressing method (34 cycles versus 17 cycles). However, the proposed method requires only 32 (16+16) control pins while 256 (16×16) pins are required for the direct-addressing method.

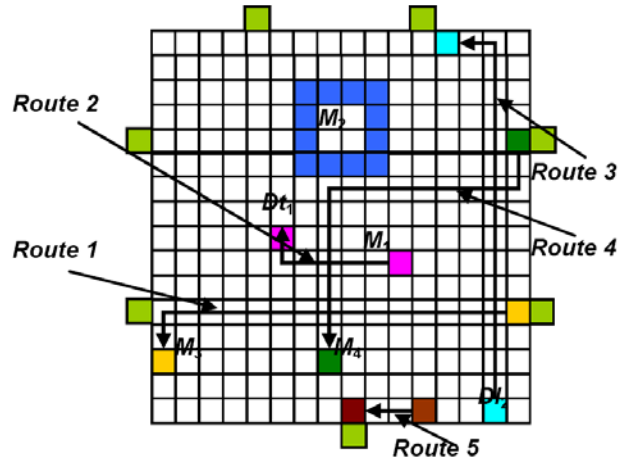


Figure 3.27: Routing plan for Sub-problem 3.

Table 3.4: Droplet manipulation results for grouping-based method without scheduling of droplet routes.

Snap shot	Cycle number	Manipulation <D ₁ , D ₂ , D ₃ , D ₄ , D ₅ >	Snap shot	Cycle number	Manipulation <D ₁ , D ₂ , D ₃ , D ₄ , D ₅ >	
1	1	<←, x, ↑, x, x>	8	28	<←, x, x, x, x>	
	2	<x, ←, x, x, x>		29	<x, x, ↑, x, x>	
	3	<x, x, x, ↓, x>		30	<x, x, x, ←, x>	
	4	<x, x, x, x, ←>		31	<x, x, ↑, ←, x>	
2	5	<←, x, x, x, x>	10	32	<←, x, x, x, x>	
	6	<x, ←, x, x, x>		33	<←, x, x, x, x>	
	7	<x, x, ↑, x, x>		34	<x, x, ↑, x, x>	
	8	<x, x, x, ↓, x>		35	<x, x, x, ←, x>	
	9	<x, x, x, x, ←>		36	<←, x, x, x, x>	
3	10	<x, x, ↑, ←, x>	12	37	<x, x, ↑, x, x>	
	11	<←, x, x, x, x>		38	<x, x, x, ↓, x>	
	12	<x, ←, x, x, x>		39	<←, x, x, x, x>	
	13	<x, x, x, x, ←>		40	<x, x, ↑, x, x>	
4	14	<←, x, ↑, x, x>	13	41	<x, x, x, ↓, x>	
	15	<x, ←, x, x, x>		42	<←, x, x, x, x>	
	16	<x, x, x, ←, x>		43	<x, x, ↑, x, x>	
5	17	<←, x, x, x, x>	14	44	<x, x, x, ↓, x>	
	18	<x, ←, x, x, x>		45	<←, x, x, x, x>	
	19	<x, x, ↑, x, x>		46	<x, x, ↑, x, x>	
	20	<x, x, x, ←, x>		47	<x, x, x, ↓, x>	
6	21	<←, x, x, x, x>	15	48	<←, x, x, ↓, x>	
	22	<x, ↑, x, x, x>		49	<x, x, ↑, x, x>	
	23	<x, x, ↑, x, x>		16	50	<↓, x, x, ↓, x>
	24	<x, x, x, ←, x>			51	<x, x, ←, x, x>
7	25	<←, x, x, x, x>	17	52	<↓, x, x, ↓, x>	
	26	<x, x, ↑, x, x>		53	<x, x, ←, x, x>	
	27	<x, x, x, ←, x>				

Table 3.5: Droplet manipulation results for grouping-based method with scheduling of droplet routes.

Cycle number	Manipulation <D ₁ , D ₂ , D ₃ , D ₄ , D ₅ >	Cycle number	Manipulation <D ₁ , D ₂ , D ₃ , D ₄ , D ₅ >
1	<x, x, x, ↓, x>	18	<↓, x, x, ↓, x>
2	<x, x, x, ↓, x>	19	<↓, x, ↑, ↓, x>
3	<←, x, ↑, ←, x>	20	<x, x, ↑, x, x>
4	<←, x, x, ←, x>	21	<x, x, ↑, x, x>
5	<←, x, x, ←, x>	22	<x, x, ↑, x, x>
6	<←, x, x, ←, x>	23	<x, x, ↑, x, x>
7	<←, x, x, ←, ←>	24	<x, x, ↑, x, x>
8	<←, ←, x, ←, ←>	25	<x, x, ↑, x, x>
9	<←, ←, x, ←, ←>	26	<x, x, ↑, x, x>
10	<←, ←, x, ←, x>	27	<x, x, ↑, x, x>
11	<←, ←, x, x, x>	28	<x, x, ↑, x, x>
12	<←, ←, x, x, x>	29	<x, x, ↑, x, x>
13	<x, x, x, ↓, x>	30	<x, x, ↑, x, x>
14	<x, ↑, x, ↓, x>	31	<x, x, ↑, x, x>
15	<x, x, x, ↓, x>	32	<x, x, ↑, x, x>
16	<x, x, x, ↓, x>	33	<x, x, ←, x, x>
17	<x, x, x, ↓, x>	34	<x, x, ←, x, x>

The above comparison is next carried out for the rest of the sub-problems. The results are shown in Figure 3.28. Note that some sub-problems, e.g., #4, #6, #8, #9, #10, #11, contain only one droplet pathway. For these sub-problems, the three methods result in the same number of cycles. Therefore, results for these sub-problems are not shown explicitly. Results for sub-problem 7 are also not shown since the two routing pathways in it have no overlap in any columns or rows, thereby no concurrent droplet movement is possible.

As we can see from Figure 3.28, the proposed grouping-based manipulation method always achieves improved concurrency over the one-at-a-time droplet-manipulation method. This improvement is significantly enhanced when the proposed route-scheduling algorithm is applied. The percentage improvement varies with the subproblem, as shown in Figure 3.28. For the entire assay, the proposed route-scheduling algorithm leads to a 47% reduction in the number of droplet-manipulation cycles compared to the one-at-a-time droplet-manipulation method, and 32% compared to the grouping-based algorithm alone.

Next we apply the power-oblivious version of the proposed method to the same problem

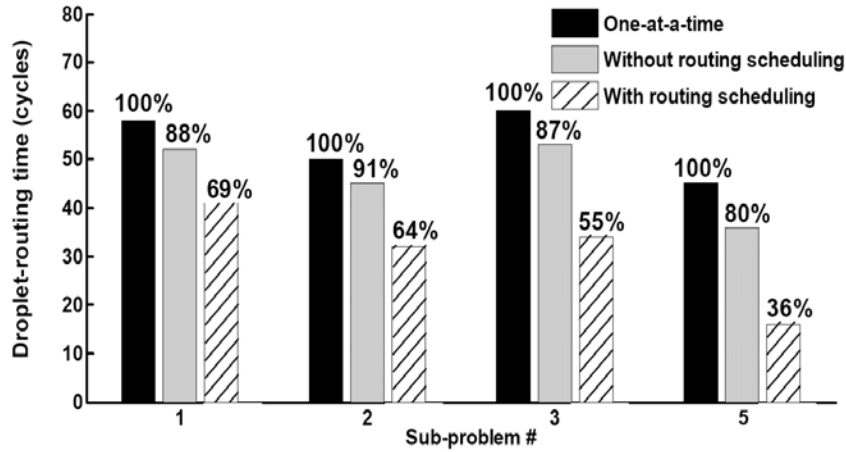


Figure 3.28: Comparison of droplet-routing times for 4 sub-problems (#1, #2, #3, #5).

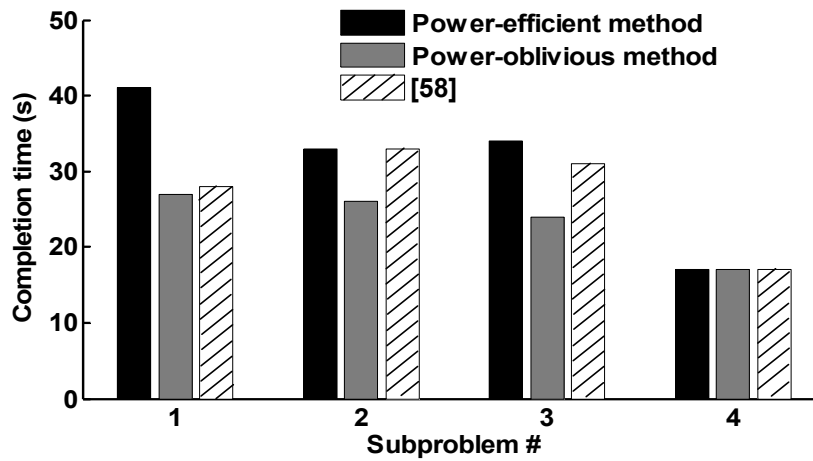


Figure 3.29: Comparison of droplet-routing times for the power-efficient-method, the power-oblivious method, and the method from [63].

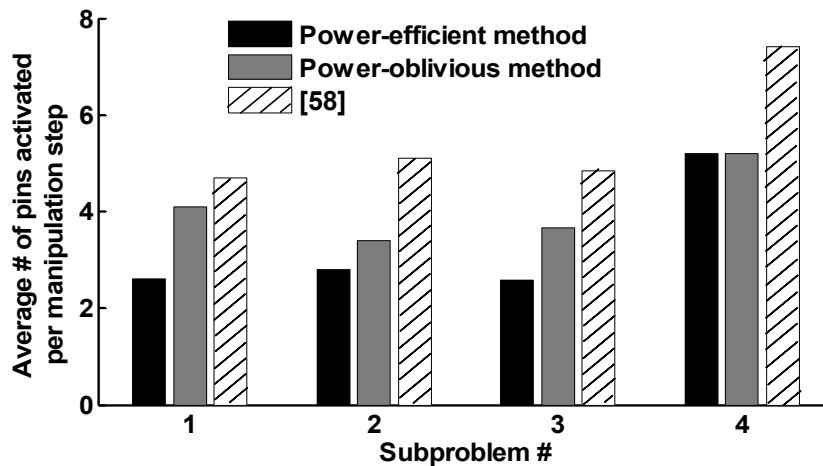


Figure 3.30: Comparison of average # of control pins activated per manipulation step for the power-efficient method, the power-oblivious method, and the method from [63].

instance. The computation time for the power-oblivious routing scheduling and manipulation method for the entire assay is 288 seconds, on a Intel Core Duo 2 GHz PC with 2G of RAM. We compare the results (completion time and power consumption, the latter measured in terms of the average number of pins activated per droplet-manipulation step) with [63] and the power-efficient version of the proposed method. Figure 3.29 and Figure 3.30 show that although the proposed power-efficient droplet-manipulation method leads to a slightly longer completion time than [63] for some subproblems, it activates a much smaller number of control pins in each manipulation cycle. This results in a significant reduction in power consumption.

Compared to [63], the power-oblivious variant of the proposed method achieves a significant reduction in assay completion time. As a tradeoff, it results in increased power consumption compared to the power-efficient method. However, the power consumption is still much smaller than in [63].

3.3 Broadcast-Addressing Method

In this section, we propose an alternative pin-constrained design scheme referred to as broadcast addressing method [64]. The proposed method provides high throughput for bioassays and it reduces the number of control pins by identifying and connecting control pins with “compatible” actuation sequences.

3.3.1 “Don’t-Cares” in Electrode-Actuation Sequences

To execute a specific bioassay, droplet routes and the schedule of operations are programmed into a microcontroller to drive the electrodes. Routing and scheduling information is stored in the form of (ternary) electrode activation sequences, where each bit representing the status of the electrode at a specific time-step. The status can be either “1” (activate), “0” (deactivate) or “x” (don’t-care).

A don't-care status on an electrode means that the electrode can be either active or inactive, as shown in Figure 3.31. At time spot t , a droplet is to be held at electrode E_3 . This electrode needs to be at high voltage ("1"), and the two adjacent electrodes E_2 and E_4 need to be deactivated ("0"). E_1 is not involved in this holding step. Since voltage on E_1 has no impact on the droplet operations for this step, E_1 can be assigned "1" or "0", i.e., "don't-care" or "x". This concept is similar to the don't-cares that arise in logic synthesis during integrated circuit design.

We use the three values "1", "0", and "x" to represent the electrode-activation sequences for a bioassay. An example is shown in Figure 3.32. A droplet is routed anticlockwise, one electrode per step, along the loop consisting of 8 electrodes. Suppose that at time instant (clock cycle) t_0 , the droplet rests on electrode E_2 . The activation sequence for each electrode is now calculated and listed in Figure 3.32(b).

In Figure 3.32(b), each sequence contains several don't-care terms, which can be replaced by "1" or "0". By careful replacing these don't-care terms, the two activation sequences corresponding to E_1 , E_4 can be made identical. For example, we can map the four don't-cares in activation sequence for E_1 with "0010" and map the four don't-cares in activation sequence for E_4 with "0100". We refer to such sequences as *compatible sequences*. Compatible sequences can be generated from a single signal source. Therefore, the corresponding electrodes E_1 and E_4 can be connected to a single control pin.

3.3.2 Optimization Based on Clique Partitioning in Graphs

In this subsection, we focus on reducing the number of control pins by connecting together electrodes with mutually-compatible activation sequences, and addressing them using a single control pin. Therefore, the resulting electrode-access method is referred to as a *broadcast addressing*. We first partition the electrodes into groups. For all the electrodes in any group, the corresponding activation sequences must be pairwise-compatible. Our goal is

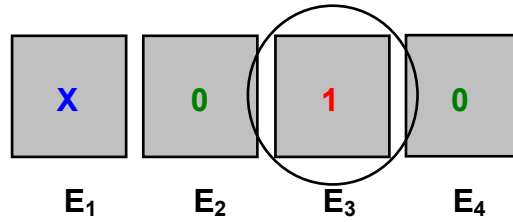
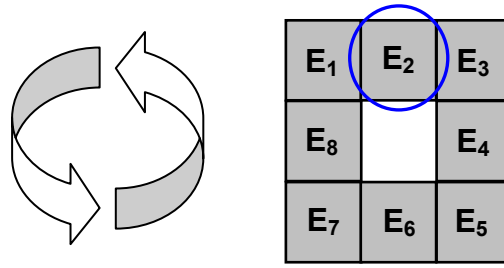


Figure 3.31: Illustration of a “don’t-care” in electrode activation.



(a)

Electrode	1	2	3	4	5	6	7	8
Activation Sequence	0	1	0	0	X	X	X	X
	1	0	0	X	X	X	X	0
	0	0	X	X	X	X	0	1
	0	X	X	X	X	0	1	0
	X	X	X	X	0	1	0	0
	X	X	X	0	1	0	0	X
	X	X	0	1	0	0	X	X
	X	0	1	0	0	X	X	X

(b)

Figure 3.32: Example of activation sequence calculation (a) routing and layout information (b) calculated activation sequences.

to find an optimal partition that leads to the minimum number of groups, which in turn yields the minimum number of control pins.

The problem of finding the minimum number of groups can be easily mapped to the clique-partitioning problem from graph theory [56]. We use the example in Figure 3.32 to illustrate this mapping. Based on the activation-sequence table, an undirected graph, referred to as electrode-activation graph, is constructed; see Figure 3.33. Each node in the graph represents an activation sequence for an electrode. An edge in the graph between a pair of nodes indicates that their corresponding activation sequences are compatible. For example, nodes 1 and 4, which represent the activation sequences for electrode E_1 and E_4 , respectively,

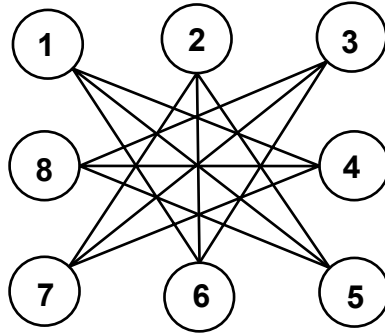


Figure 3.33: Mapping of the activation sequences of Figure 3.32 to an undirected graph.

are connected by an edge because the activation sequences can be converted to a single sequence “01000010” by replacing the don’t-care terms.

As mentioned in Section 3.2.2, a *clique* in a graph is defined as a complete subgraph, i.e., any two nodes in this subgraph are connected by an edge [56]. *Clique partitioning* refers to the problem of dividing the set of nodes into non-overlapping subsets such that the subgraph induced by each subset of nodes is a clique. A *minimal clique partition* is one that covers the nodes in the graph with a minimum number of non-overlapping cliques. The grouping of droplets as discussed above is equivalent to the clique-partitioning problem. A minimal clique partition here for this example is given by $\{1,4\}$, $\{5,8\}$, $\{2,6\}$, $\{3,7\}$. Even though the general clique partitioning problem is known to be NP-hard [57], a number of heuristics are available in the literature to solve it in an efficient manner.

After an efficient partitioning of electrodes is derived, we address all the electrodes in a group using a single control pin. A common activation sequence compatible to all the individual sequences in each group is calculated and used as the input sequence for the control pin. In the above example, electrodes E_1 , E_4 are connected and they share the common activation sequence of $\{01000010\}$. Since we broadcast a common activation sequence to several electrodes, we refer to this addressing method as “broadcast addressing”.

The complete steps in broadcast addressing are as follows:

1. Obtain droplet-routing information from the biochip synthesis results and calculate

the control-signal sequence for each control pin. The control-signal sequence consists of the values 1 (activated), 0 (deactivated), and x (don't-care).

2. Draw an undirected graph representing the relationship between control-signal sequences. For every pair of electrode-activation sequence, if one sequence can be derived from the other by simply changing x's to 1's/0's, then draw an edge between the nodes representing them.
3. Apply clique partitioning to minimize the number of independent control signals.
4. Group and connect the control lines that are in the same clique.

The general clique partitioning problem is known to be NP-hard [57]. Therefore, we use a heuristic based on the union-find algorithm [73], which partitions the graph by iteratively searching for a maximal clique, defined as a clique not contained in any larger clique, and then deleting the maximal clique from the graph. The algorithm takes $O(N^3)$ computation time, where N is the number of electrodes on the chip.

By using this broadcast-addressing method, the input bandwidth for the microfluidic biochip can be significantly reduced. For the example in Figure 3.33, instead of using eight independent control pins to address the electrode loop, broadcast addressing only needs four control pins. A more significant reduction is expected in large arrays with more don't-care terms in activation sequences.

Another advantage of the broadcast-addressing method is that it provides maximum freedom of droplet movement. It does not change the schedule of operations or the droplet-routing pathways for the target bioassay; therefore, bioassays can be executed as fast as on a direct-addressing-based chip. Compared to the array-partitioning-based method presented in Section 3.1, broadcast addressing does not need to limit the number of concurrent droplet movements to get fewer partitions. The proposed method also reduces assay operation time compared to cross-referencing; the latter typically requires several sub-steps for a set of droplet manipulations that can be carried out concurrently in a

direct-addressing-based chip. These advantages are quantitatively evaluated using a real chip example in Section 3.3.4.

3.3.3 Broadcast Addressing for Multifunctional Biochips

Broadcast addressing can also be applied to multifunctional biochips, i.e., biochips targeting the execution of a set of (multiple) predetermined bioassays. For each target bioassay, droplet routing and schedule information are collected and activation sequences are calculated. Next, for each electrode, we merge the activation sequences from the different assays and obtain a collective activation sequence. Note that the compatibility of activation sequences is independent of the ordering of the sequences. Therefore, the merging of activation sequences can be carried out in any arbitrarily-chosen order.

Once the collective activation sequences are derived, the same steps as described in Section 3.3.2 are carried out to derive the electrode partitions and the wiring (connection of input pins to electrodes) plan.

Note that the longer the activation sequences, the more specified entries, i.e., “1” and “0” exist, and the less compatibility we observe. Therefore, multi-functionality may necessitate a larger number of input control pins for the proposed broadcast addressing method. This trade-off is evaluated in the next section.

3.3.4 Experimental Results

In this section, we evaluate the proposed broadcast addressing method by using it to pin-constrained design of biochips for a multiplexed immunoassay, a representative protein assay, and the polymerase chain reaction (PCR) procedure.

Each assay is first mapped to a 15×15 electrode array controlled using the direct-addressing scheme. Next, the proposed broadcast-addressing method is used to reduce the number of control pins.

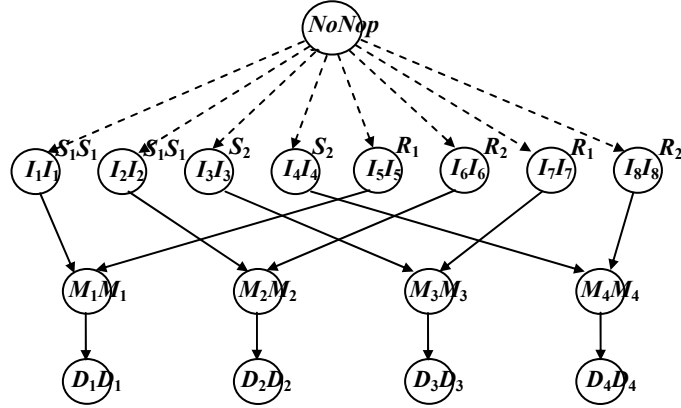


Figure 3.34: Sequencing graph model for the multiplexed bioassay. S_1, S_2 are samples, R_1, R_2 are reagents, $M_1 \sim M_4$ are mixing operations, and $D_1 \sim D_4$ are detection operations.

Multiplexed Assay

We first map the multiplexed biochemical assay described in Section 3.1.4. Figure 3.34 shows the flowchart for the multiplexed assays in the form of a sequencing graph. For each sample or reagent, two droplets are dispensed into the array. Four pairs of droplets, i.e., $\{S_1, R_1\}$, $\{S_1, R_2\}$, $\{S_2, R_1\}$, $\{S_2, R_2\}$, are routed together in sequence for the mixing operation. Mixed droplets are finally routed to the detection site for analysis.

As discussed in Section 3.1.4, the multiplexed bioassays were mapped to a digital microfluidic platform containing a 15×15 array, as shown in Figure 3.11. A depiction of the droplet pathways for multiplexed glucose and lactase assays is given in Figure 3.11.

In the multiplexed assay, eight droplets (two droplets from each sample/reagent) are dispensed and routed to the mixer located at the center. Next, four mixing and detection operations are carried out in a pipeline manner following the schedule shown in Table 3.2. We assume that the droplets are transported at the rate of 1 electrode/second, i.e., 1 Hz.

Next we apply the proposed broadcast-addressing method to the above chip layout. As shown in Figure 3.11, the multiplexed-assay chip utilizes 59 electrodes. We calculate the electrode activation sequences based on the scheduling and routing result presented in Section 3.1.4. A fragment of the activation sequences is listed in Table 3.6. Next, the

Table 3.6: A fragment of the activation sequences for multiplexed assay.

Electrode # (7~20)	Activation Sequences(0s ~ 13s)												
7	1	0	x	x	x	0	1	0	x	x	x	0	...
8	0	1	0	x	x	x	0	1	0	x	x	x	...
9	x	0	1	0	x	x	x	0	1	0	x	x	...
10	x	x	x	x	x	x	x	x	x	x	x	x	...
12	x	x	x	x	x	x	x	x	x	x	x	x	...
13	x	0	1	0	x	x	x	0	1	0	x	x	...
14	0	1	0	x	x	x	0	1	0	x	x	x	...
15	1	0	x	x	x	0	1	0	x	x	x	0	...
16	x	x	0	1	0	x	x	x	0	1	0	x	...
:	:	:	:	:	:	:	:	:	:	:	:	:	:

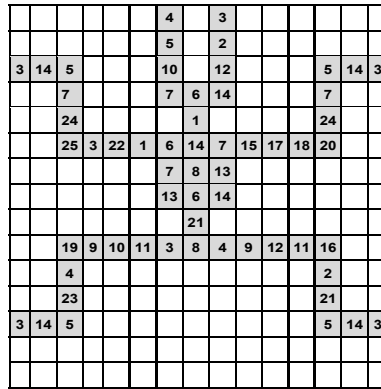


Figure 3.35: Broadcast addressing for the multiplexed assay chip.

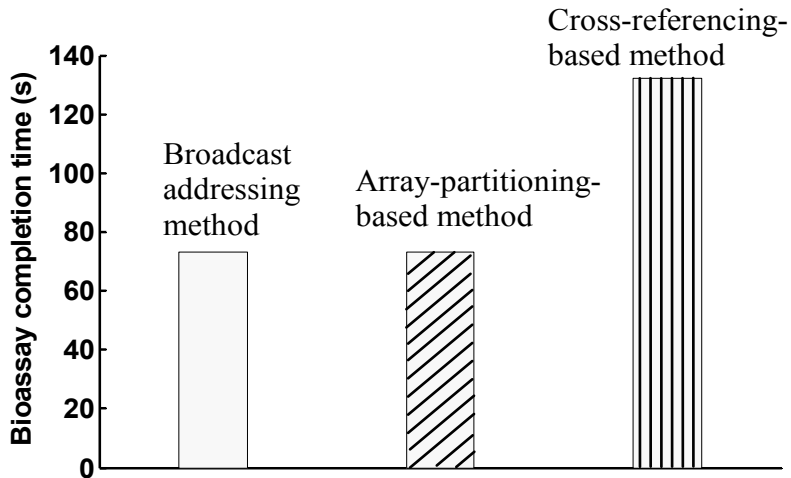


Figure 3.36: Comparison of assay completion times.

clique-partitioning-based broadcast addressing method is used to generate the electrode connections and the pin-assignment plan. The results are shown in Figure 3.35. The pins assigned to the electrodes are shown in the corresponding boxes.

In Figure 3.35, the number of control pins is reduced from 59 to 25, almost a 60% reduction compared to direct-addressing method. Due to considerable reduction in wiring complexity, fabrication cost is reduced significantly. There is no increase in the assay time compared to a direct-addressing chip that uses 59 electrodes.

The cross-reference-based method in also leads to a significant reduction in number of control pins, but at the expense of higher assay completion times. The results are shown in Figure 3.36. With broadcast addressing, we obtain an assay completion time of 73 s. The cross-referencing-based method requires the 30 control pins but a longer completion time of 132 s. The array-partitioning-based method described in Section 3.1 leads to a completion time of 73 s. However it requires 35 control pins, i.e., an increase of 40% compare to the broadcast-addressing method.

Polymerase Chain Reaction (PCR)

For the second assay, we use the mixing stages of the PCR. These stages are used for rapid enzymatic amplification of specific DNA strands. Recently, the feasibility of performing droplet-based PCR on digital microfluidics-based biochips has been successfully demonstrated [26]. Its assay protocol can be modeled by a sequencing graph, as shown in Figure 3.37. Mapping the protocol on to the array, we obtain the chip layout and schedule shown in Figure 3.38 and Figure 3.39, respectively.

Assuming a direct-addressing scheme, the layout in Figure 3.38 requires 62 control pins. However, using the proposed broadcast-addressing method, we reduce the number of control pins to 14. The pin-constrained layout for the PCR chip is shown in Figure 3.40.

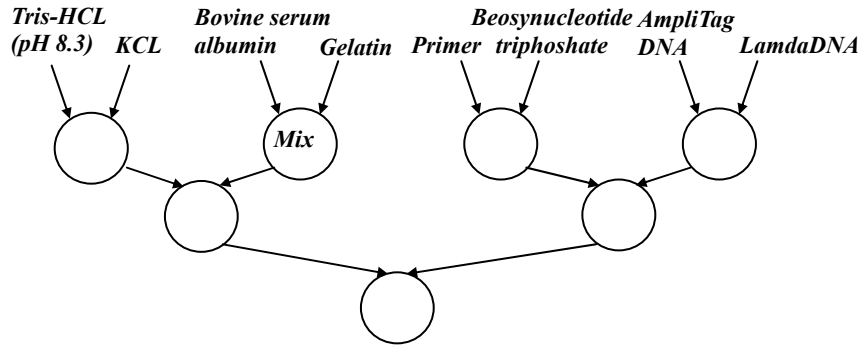


Figure 3.37: Sequencing graph for the mixing stage of PCR.

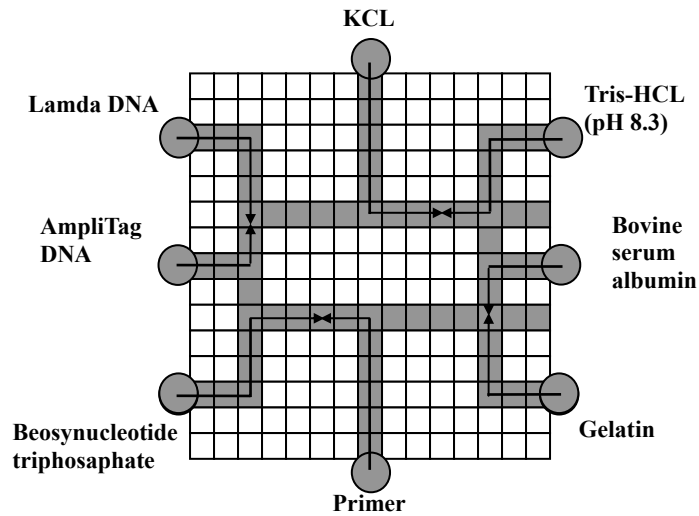


Figure 3.38: Mapping of the PCR assay on a 15x15 array.

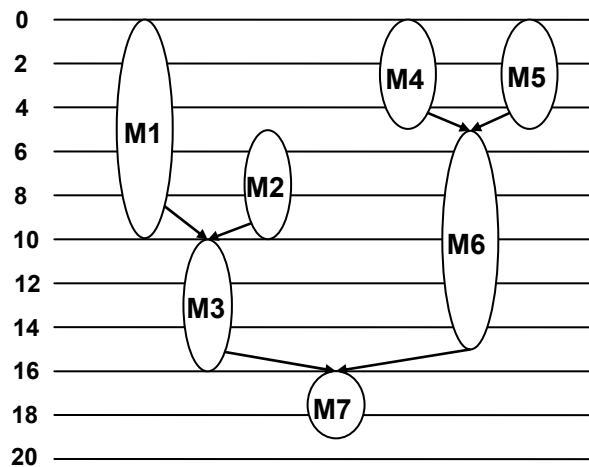


Figure 3.39: Schedule for the PCR assay.

								1							
								2							
1	2	3						3					3	2	1
		7						5					7		
		14						9					12		
		8	3	12	1	6	14	7	11	14	6	13			
		6											9		
1	4	5											6	5	1
		12											14		
		8	7	11	9	13	10	6	13	7	11	6			
		4						11				2	2		
		13						6					8		
1	2	3						3					3	2	1
								2							
								1							

Figure 3.40: Broadcast addressing for the PCR chip.

Protein Dilution

The third assay that we consider consists of the dilution steps in a real-life protein assay described in Section 2.4. The flow chart is shown in Figure 2.5. A droplet of the protein sample with concentration C is diluted on chip through seven hierarchies of binary mixing/splitting phases to derive the target droplet with a sample concentration of $C/128$. After dilution, droplets of reagents, such as Coomassie brilliant blue G-250 dye, are dispensed into the chip, and they mix with the diluted sample droplets. Next the mixed droplet is transported to a transparent electrode, where an optical detector (e.g., a LED-photodiode setup) is integrated. The protein concentration can be measured from the absorbance of the products of this colorimetric reaction using a rate kinetic method.

We map the protein assay to the 15×15 array. Figure 3.41 shows the chip layout and Figure 3.42 illustrates the schedule for this protocol. In Figure 3.41, 52 electrodes are used in the chip layout. This number is reduced to only 27 after the broadcast-addressing method is applied; see Figure 3.43.

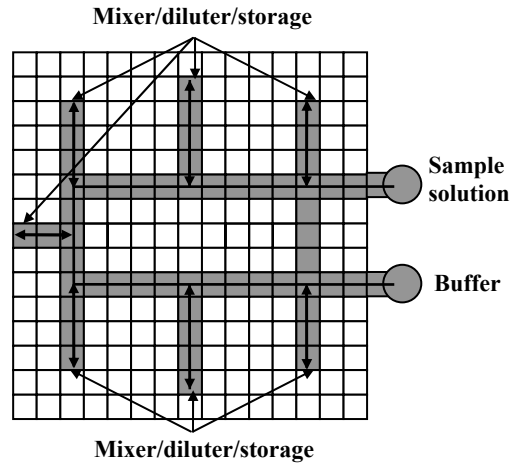


Figure 3.41: Layout for protein-dilution chip.

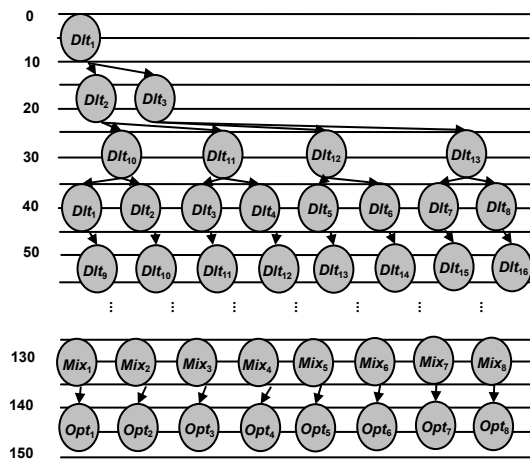


Figure 3.42: Schedule for the protein dilution assay, Dlt – dilution, Mix – mixing, Opt – optical detection.

						3											
		6				1									24		
		8				4								7			
		11				6								12			
		7	3	15	18	26	14	7	25	12	13	14	8	4			
		16												9			
		7	4	18										17			
		13												20			
		9	7	26	11	13	27	24	23	21	22	19	9	5			
		4					23							16			
		2					11							10			
		1					3							7			
							2										

Figure 3.43: Broadcast-addressing for the protein-dilution chip.

					18	12	24								
					7	22	5								
24	8	12			11	23	15					15	14	3	
		5			5	22	8							13	
		36				9								24	
		35	24	1	9	22	8	5	6	25	33	34	27	28	
		17				5	13	21					23		
24	10	26				21	22	8				21	37	13	
		19					10						8		
		27	12	11	23	24	13	18	37	15	23	28	19	17	
		18					23						16		
		4					8						10		
24	8	7					14					5	8	24	
							2								
							4								

Figure 3.44: Pin-assignment layout for multi-functional chip.

Broadcast-Addressing for a Multi-functional Chip

Finally, we evaluate the performance of the proposed method for multi-functional biochip design. Here we design a multi-functional biochip that can execute all the three assays described in the previous subsections. The pin-assignment for the multi-functional chip can be obtained by combining the chip layouts for the three different assays, see Figure 3.44. Note that only 81 electrodes on the 15×15 array are used in this layout and thereby need to be addressed.

Next we consider the addressing problem for the multi-functional chip. The activation sequences for the PCR assay and protein assay are determined and combined with that from the multiplexed assay. The broadcast addressing method is carried out and it generates a chip layout with only 37 control pins.

The addition of two assays to the biochip for the multiplexed assay, and 22 (81–59 = 22) new electrodes, leads to only 13 extra control pins. These results highlight the scalability attribute of the proposed design method.

3.4 Chapter Summary and Conclusions

We proposed an efficient partitioning and pin assignment algorithm for pin-constrained digital microfluidic biochips. The partitioning method is based on the concept of droplet trace,

which is calculated from scheduling and placement results from automated synthesis tools. An efficient pin assignment scheme based on Connect-5 algorithm is used to derive the actual pin layout.

We have also proposed a droplet manipulation method for a “cross-referencing” addressing method that uses “row” and “columns” to access electrodes in digital microfluidic arrays. By mapping the droplet movement problem to the clique partitioning problem from graph theory, the proposed method allows simultaneous movement of a large number of droplets. A linear-time heuristic algorithm based on row-scanning and column-scanning has been used to derive the clique partitions. To further increase routing concurrency, a route-scheduling algorithm has been developed to pre-align droplet movements. A power-oblivious version of the method is also presented; it which allows higher-throughput manipulations with a slight increase in power consumption. We have used random synthetic benchmarks and a set of multiplexed bioassays to evaluate the proposed method.

Finally, we have described a broadcast-based electrode-addressing method for pin-constrained digital microfluidic biochips. We have shown how compatible electrodes are identified and connected. This procedure leads to a considerable reduction in the number of control pins. We have used the proposed method to solve the electrode addressing problem for a multi-functional biochip and achieved a significant reduction in the input-control bandwidth required for a set of bioassays. These pin-constrained design methods will allow bioassays for high-throughput sequencing, immunoassays, and clinical diagnostics to be mapped to pin-constrained and low-cost biochips, and simplify the design and implementation of such biochips.

Chapter 4

Testing and Diagnosis

Dependability is essential for microfluidic biochips that are used for safety-critical applications. Therefore, these devices must be adequately tested after manufacture and during bioassay operations. We present a parallel scan-like testing methodology for digital microfluidic devices [65]. A diagnosis method based on test outcomes is also proposed. The diagnosis technique is enhanced such that multiple defect sites can be efficiently located using parallel scan-like testing. Defect diagnosis can be used to reconfigure a digital microfluidic biochip such that faults can be avoided, thereby enhancing chip yield and defect tolerance.

In addition to error detection and localization of defect sites, we also focus on ensuring correct operation of functional units. We first introduce the concept of functional testing of microfluidic biochips [66]. We address fundamental biochip operations such as droplet dispensing, droplet transportation, mixing, splitting, and capacitive sensing. Long electrode actuation times are avoided to ensure that there is no electrode degradation during testing. The functional testing of pin-constrained biochips is also studied. We evaluate the proposed test methods using simulations as well as experiments for a fabricated biochip.

4.1 Parallel Scan-like Test

In this section, we describe efficient fault detection and diagnosis methods based on multiple-droplet testing. The key idea here is to manipulate multiple droplets in parallel to test the microfluidic array in a scan-like manner.

Table 4.1 lists common defects and their causes. It also maps each defect to a fault model and an observable error. Note that all of these defects are catastrophic, i.e., their occurrence

will cause the chip to fail. Compared to the defects listed in [28], we have identified several new defects that either result from manufacturing problems or appear during bioassay execution, e.g., electrode charging, misaligned parallel plates, and grounding failure. Note that even though the causes for these new defects are unique, their observable errors are already included in the set of errors targeted in [28]. Therefore, all the defects listed in Table 4.1 can be detected by manipulating test droplets to traverse the candidate faulty electrodes. Note that to detect an electrode-short defect, a test droplet needs to traverse two adjacent electrodes that are involved in the short. The test droplet will reside in the middle of the two shorted electrodes that are activated simultaneously; there will not be sufficient overlap area with the next electrode for further transportation.

Most prototype digital microfluidic devices consist of a two-dimensional array of electrodes with one or more sources and sinks, i.e., reservoirs, on the boundary, as shown in Figure 4.1 [25]. In this regular structure, electrodes are carefully aligned in columns and rows. We next describe the parallel scan-like test method, named thus because it manipulates multiple test droplets in parallel to traverse the target microfluidic array, just as test stimuli can be loaded in parallel to multiple scan chains in integrated circuits.

We first describe the special case of a single test droplet. We determine the pathway for the test droplet, irrespective of the bioassay operation, as shown in Figure 4.1. Starting from the droplet source, the test droplet follows the pathway to traverse every cell in the array, and it finally reaches the sink. During concurrent testing, a test droplet is guided to visit the available cells in accordance with a predetermined path. If the target cell is temporarily unavailable for testing, i.e., it is occupied by a droplet or it is adjacent to active microfluidic modules, the test droplet waits in the current position until the target cell becomes available. The test outcome is read out using a capacitive sensing circuit connected to the electrode for the sink reservoir, as shown in Figure 4.2. The figure shows details about the set setup and

Table 4.1. List of catastrophic defects for biochips.

Cause of defect	Defect type	*	Fault model	Observable error
Excessive actuation voltage applied to an electrode	Dielectric breakdown	1	Droplet-electrode short (a short between the droplet and the electrode)	Droplet undergoes electrolysis, which prevents its further transportation
Electrode actuation for excessive duration	Irreversible charge concentration on an electrode	1	Electrode-stuck-on (the electrode remains constantly activated)	Unintentional droplet operations or stuck droplets
Excessive mechanical force applied to the chip	Misalignment of parallel plates (electrodes and ground plane)	1	Pressure gradient (net static pressure in some direction)	Droplet transportation without activation voltage
Coating failure	Non-uniform dielectric layer	1	Dielectric islands (islands of Teflon coating)	Fragmentation of droplets and their motion is prevented
Abnormal metal layer deposition and etch variation during fabrication	Grounding Failure	1	Floating droplets (droplet are not anchored)	Failure of droplet transportation
	Broken wire to control source	1	Electrode open (electrode actuation is not possible)	Failure to activate the electrode for droplet transportation
	Metal connection between two adjacent electrodes	2	Electrode short (short between electrodes)	A droplet resides in the middle of the two shorted electrodes, and its transport along one or more directions cannot be achieved
Particle contamination or liquid residue	A particle that connect two adjacent electrodes	2	Electrode short	A droplet resides in the middle of the two shorted electrodes, and its transport along one or more directions cannot be achieved
Protein absorption during a bioassay	Sample residue on electrode surface	1	Resistive open at electrode	Droplet transportation is impeded.
			Contamination	Assay results are outside the range of possible outcomes

* Number of cells involved in the defect.

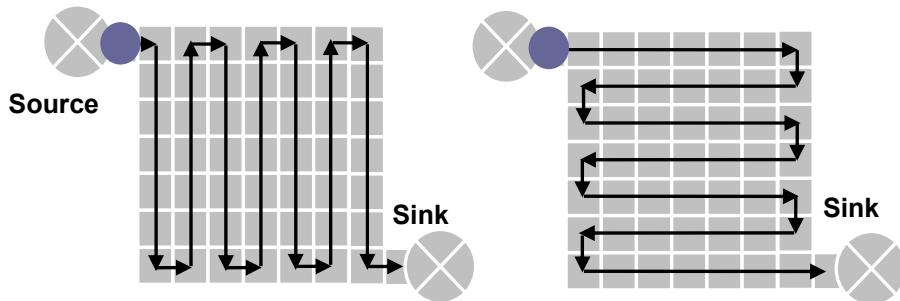


Figure 4.1: Illustration of single droplet scan-like test using a single droplet.

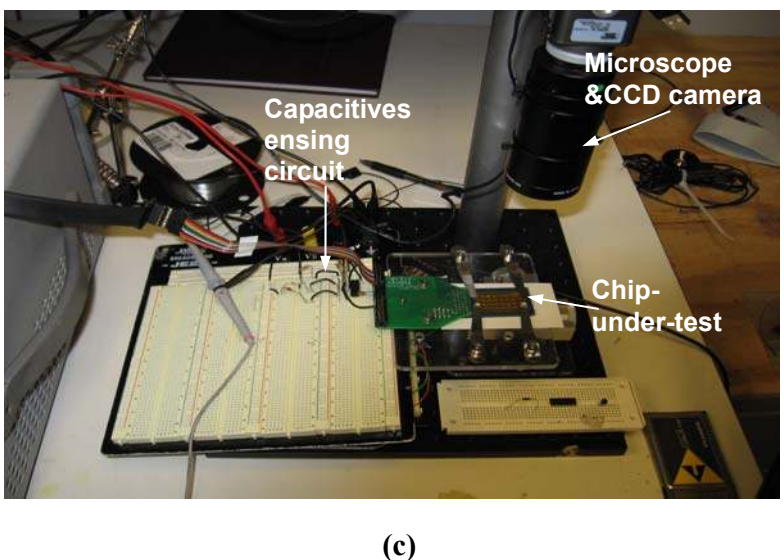
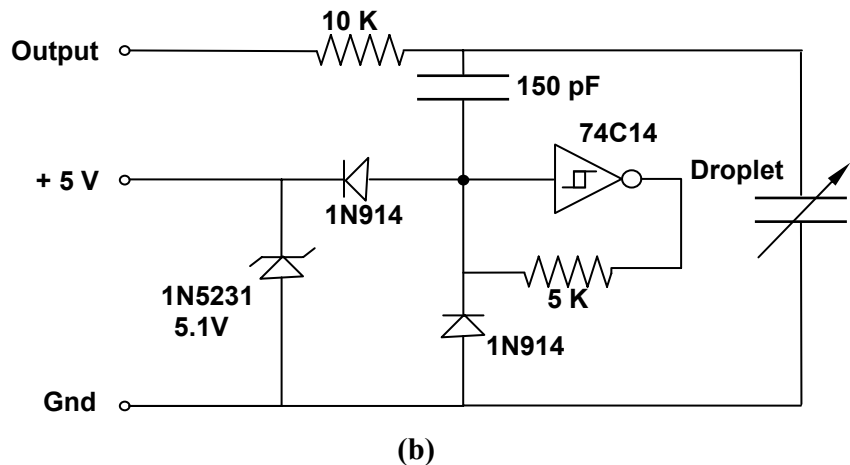
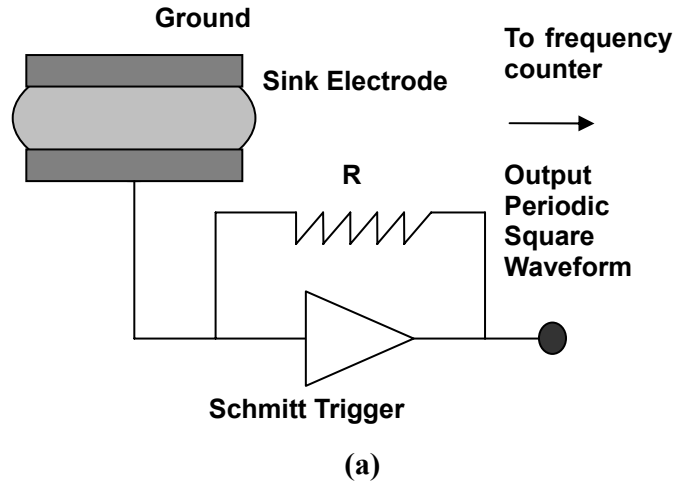


Figure 4.2: Capacitive sensing circuit: (a) outline [67]; (b) detail circuit design; (c) experimental setup.

how it was validated. This single-droplet, scan-like algorithm is easy to implement. Moreover, the test plan is general, in the sense that it can be applied to any microfluidic array and for various bioassay operations.

However, in this simple test procedure method, $N \times M$ steps (clock cycles for droplet actuation) are needed for the test droplet to traverse an $N \times M$ microfluidic array. As a result, the test time may be excessive for large arrays. For example, a 600,000-electrode-array manufactured by Silicon Biosystems (based on dielectrophoresis) will require 600,000 clock cycles [68]. At a typical actuation clock frequency of 1 Hz, this amounts to seven days of test application time! Moreover, in on-line testing, the test droplet may have to be stalled several times, and each time a long waiting period may be necessary. Finally, the test outcome for the single-droplet scan-like test provides no diagnostic information. If any cell in the array is faulty, the capacitive sensing circuit will have no readout; however, it is not possible to identify the location of the faulty cell.

The above problems can be tackled by carrying out the scan-like test in parallel using multiple droplets. Each column/row in the array is associated with a test droplet and its “target region”. A target region for a droplet includes the cells that are traversed by this droplet. The proposed method can be viewed as carrying out a single-droplet scan-like test in different target regions in parallel, therefore we refer to this method as *the parallel scan-like method*.

4.1.1 Off-line Test and Diagnosis

In off-line testing, test droplets are dispensed from the test droplet source to the start electrodes of their target regions. Since we use columns/rows as target regions, the start electrodes are located on the array boundary, as shown in Figure 4.3. For each target region, the start electrode acts as the test-droplet source for the underlying single-droplet scan-like

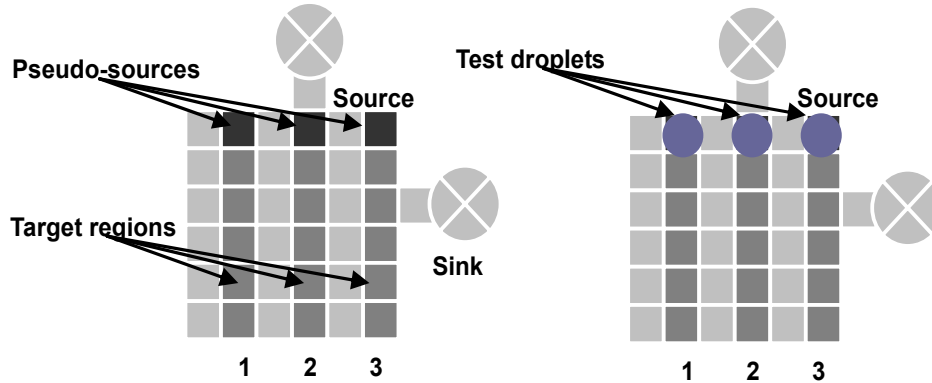


Figure 4.3: Example of target regions and pseudo-sources.

method. Therefore, we refer to start electrodes here as pseudo-sources. Starting from these pseudo-sources, test droplets are routed in parallel (similar to a waterfall in nature) to the electrodes at the other end of the corresponding target regions. These end-points are referred to as pseudo-sinks. Finally, the test droplets are routed to the sink reservoir. Note that in above method, we assume that a microfluidic array has only one source and one sink reservoir to facilitate chip packaging and reduce fabrication cost. Dispensed from the single source, test droplets are aligned one-by-one and routed in sequence, like components in an assembly line, along the periphery nodes to their pseudo sources. The reverse process is carried out when the test droplets are routed from the pseudo sink to the sink reservoir.

As in [28], the test outcome is read out using the capacitive sensing circuit located at the sink reservoir. The major enhancement here is that multiple test droplets can be detected at the sink. Instead of a single pulse, the capacitive sensing circuit can detect a pulse sequence corresponding to multiple test droplets; see Figure 4.4. Different fault patterns (i.e., groups of failing cells) are mapped to different pulse sequences.

Sufficient spacing between droplets must always be maintained during droplet routing [27]. Therefore, in order to avoid unintentional merging of droplets, test droplets must be at least one electrode away from each other. This implies that only half of the total number of columns/rows can be tested in one parallel scan-like test iteration. Two iterations are needed

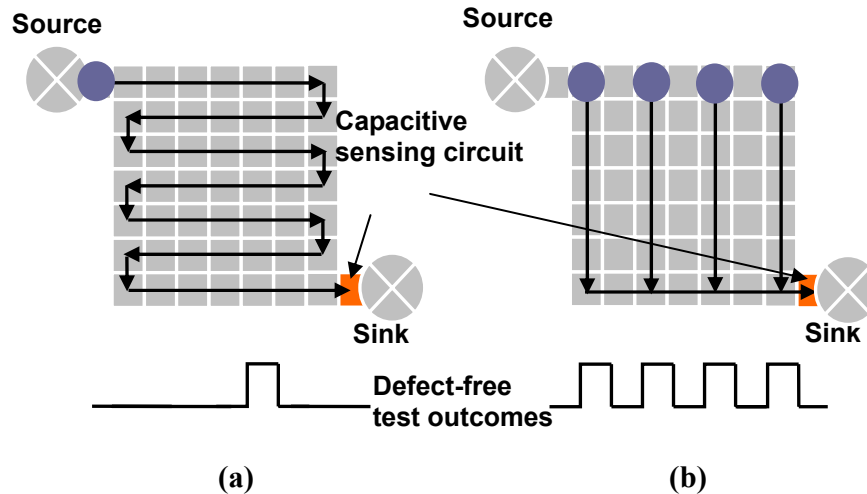


Figure 4.4: Example defect-free test outcome for (a) single-droplet scan-like method, and (b) parallel scan-like method.

to detect defects involving single cells in the array. For defects involving two cells, e.g., shorts between two adjacent electrodes, all pairs of adjacent electrode must be tested. Therefore, four iterations are needed to test the microfluidic array—two iterations for the vertically-connected pairs and two additional iterations to traverse all the horizontal connections. In addition, a “peripheral test” is carried out before parallel scan-like testing to ensure that a test droplet can be correctly dispensed from the source to pseudo-sources and routed from pseudo-sinks to the sink.

The complete parallel scan-like test procedure is as follows:

Step I. Peripheral Test: A test droplet is dispensed from the source. It is routed to traverse all the peripheral electrodes, and the droplet finally returns to the sink, as shown in Figure 4.5.

Step II. Column Test: Two iterations of parallel scan-like test with one column shift are carried out. This step tests every single cell and all “edges” (pairs of adjacent cells) in each column. Therefore, it is referred to as “column test”.

Step III. Row Test: Repeat parallel scan-like test (two iterations) for the rows to detect defects involving pairs of adjacent cells in each row. This step is referred to as “row test”.

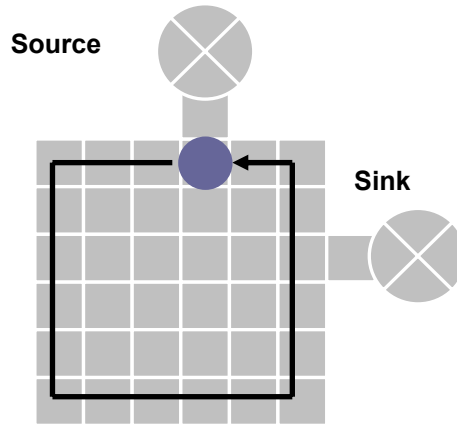


Figure 4.5: Step I (Peripheral testing).

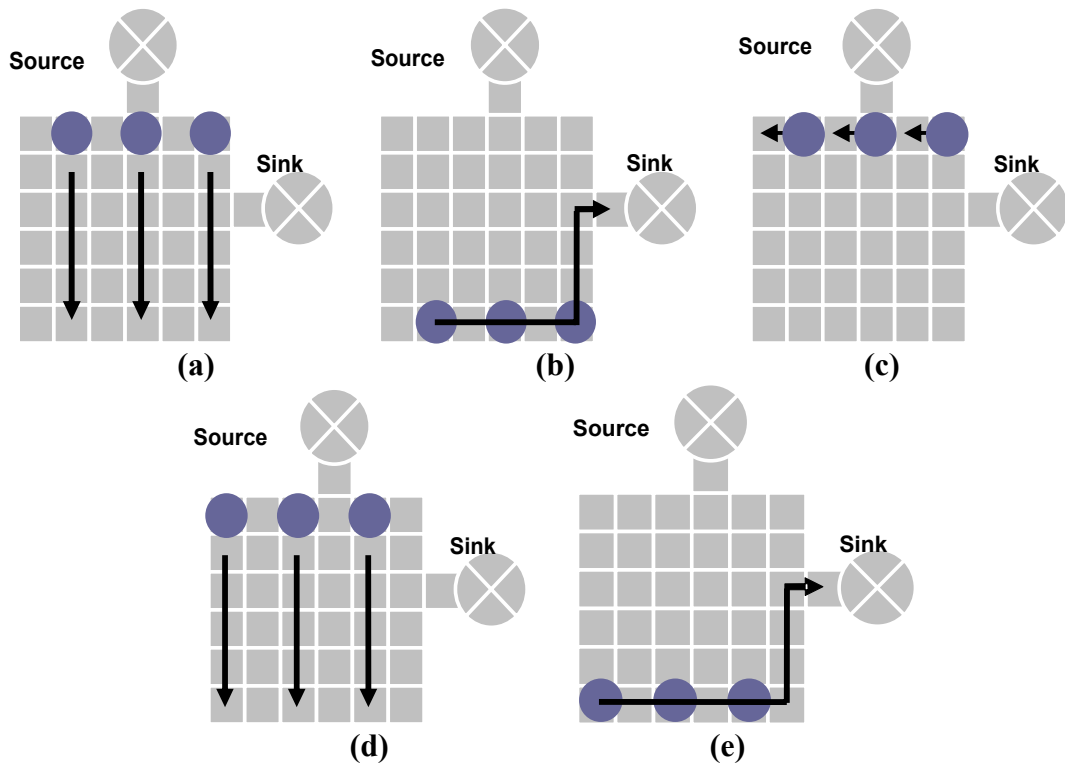


Figure 4.6: Step II: (a) parallel scan-like test for even columns; (b) routing of test droplets to sink (c) test droplet routed to odd columns; (d) parallel scan-like test for odd columns; (e) routing of test droplets to sink.

We next use a 6×6 array as an example. A total of three test droplets ($6 / 2 = 3$) are dispensed and routed in each parallel scan-like test iteration. The various steps in the test application procedure are shown in Figures 4.5-4.7.

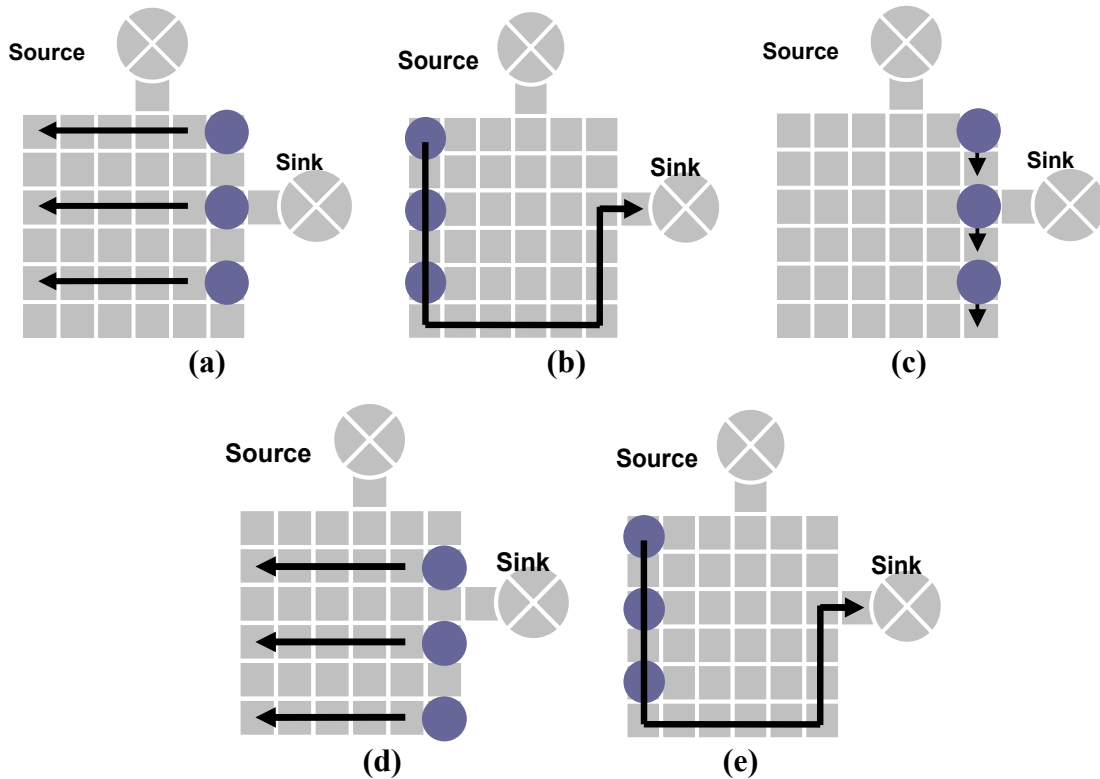


Figure 4.7: Step III: (a) parallel scan-like test for odd rows; (b) routing of test droplets to sink (c) test droplet routed to even rows; (d) parallel scan-like test for even rows; (e) routing of test droplets to sink.

In order to achieve defect tolerance via reconfiguration, a diagnosis method is needed to locate faulty cells. We do not attempt to identify the defect type in this work. We only distinguish between a defect involving one cell and a defect involving two cells, i.e., an electrode-short. We next present an efficient diagnosis procedure based on parallel scan-like testing.

For defect-free chips, test droplets, which start simultaneously at the corresponding pseudo-sources, traverse their target columns/rows and reach their pseudo-sinks at the same time. The droplets are then routed from the pseudo-sinks to the sink, triggering a pulse sequence, as shown in Figure 4.4(b). If there is a defect in a row or a column, the corresponding droplet will not arrive at the pseudo-sink. Different pulse patterns correspond to different defect locations. Consider the example shown in Figure 4.8. The output pulse

sequence indicates a defect in the fifth column. The defect site can be precisely identified by carrying out the row tests. In some cases, it is difficult to map test outcomes to candidate fault patterns, e.g., a test outcome missing the first pulse is the same as that missing the last pulse. To solve this problem, the arrival time for the first test droplet is calculated before test application; in this way, we get a reference point in order to avoid ambiguous interpretation of test outcomes.

Using the above method, a single faulty cell can be located as shown in Figure 4.9; the test droplet will be stuck at the faulty cell in both the column-test and row-test steps. If the defect is an electrode-short, the test droplet will be stuck at the short site in either the row-test or column-test, but not both. No additional diagnosis steps are needed. In contrast, Euler-path-based testing [36] rely on a binary search process to determine the exact location of the defect. Thus, parallel scan-like test saves time for many testing applications where only defect-type information is needed.

Defect localization is more complicated when the fault is due to a short between two electrodes. In this case, the sensor readout indicates an error for only one step, i.e., either column-test or row-test; see Figure 4.10. A binary-search method can next be used to locate the shorted electrode pair by iteratively partitioning the column/row and carry out single line parallel scan-like test, as shown in Figure 4.10.

Compared to the Euler-circuit based method, the two-dimensional array partitioning of [36] is simplified to 1-D. As a result both number of test iterations and the complexity of each iteration are reduced, which lead to a significant decrease in diagnosis time.

The above discussion on diagnosis assumes a single faulty cell. The procedure can also be used to locate multiple faulty cells, but it does not guarantee that only faulty cells are placed in the set of candidate defect sites.

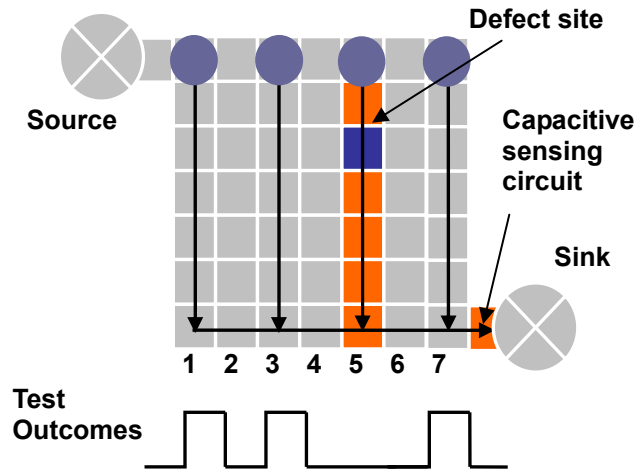


Figure 4.8: Example of test outcome for a faulty array.

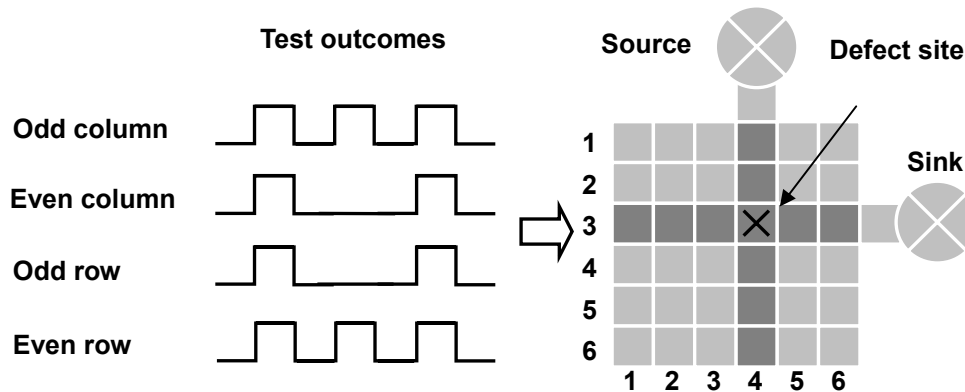


Figure 4.9: Diagnosis of a single-electrode defect by “cross-parallel” scan-like test.

4.1.2 On-line Parallel Scan-like Test

The proposed parallel scan-like test method can also be used for on-line testing with a few modifications. First, note that some test droplets are stalled while the target cells are occupied by the sample and reagent droplets needed for the bioassay. Therefore, some test droplets may be out of step with each other, as shown in Figure 4.11. The three droplets arrive at the pseudo-sinks at different times. Second, test droplets are routed to the sink only after all the test droplets arrive at the boundary. While this procedure leads to an increase in

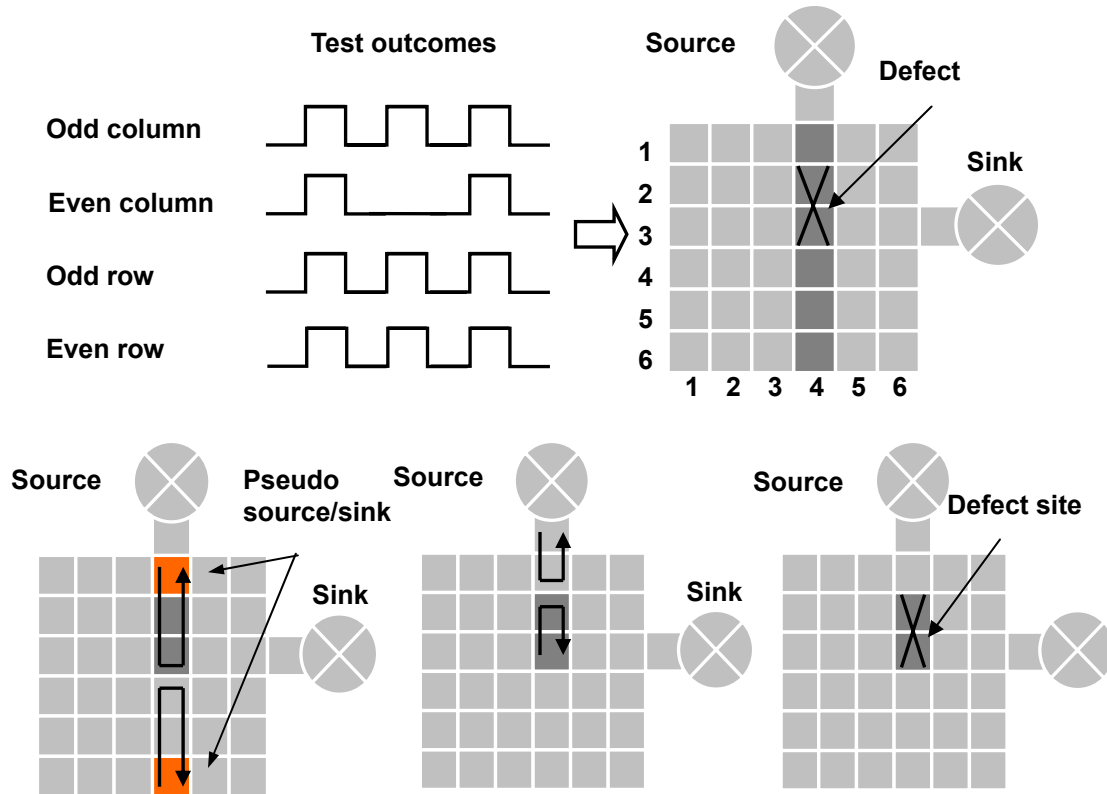


Figure 4.10: Diagnosis of an electrode-short defect using parallel scan-like test.

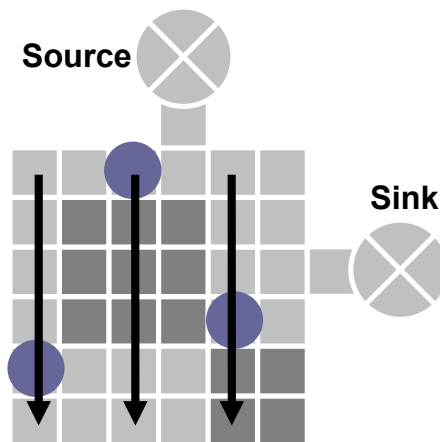


Figure 4.11: On-line parallel scan-like test for a 6x6 microfluidic array. Shaded cells correspond to modules in use for bioassay operations.

testing time, it guarantees a regular output at the detection sensor that is easy to read, therefore the capacitive sensing circuit can be kept simple. On-line fault diagnosis,

although based on the same idea as off-line diagnosis, is more complicated since test droplets are moving out-of-step with each other. To determine the arrival time for droplet detection, operation scheduling and module placement results [15] are used to calculate the waiting time for test droplets. Once arrival times are determined, on-line fault diagnosis can be carried out using the same procedure presented earlier for off-line diagnosis.

4.2 Diagnosis of Multiple Defects

The proposed parallel scan-test method efficiently tests the target biochip and locates defects. However, it is not able to always unambiguously and accurately locate multiple defect sites. In this section, we integrate a redundant test method into the parallel scan-like test technique to address this problem.

4.2.1 Incorrectly-Classified Defects

When multiple defects exist in the array, multiple columns and multiple rows might fail during the parallel scan-like test method. However, unlike in the case of a single defect, we cannot identify the multiple defect locations by simply examining the failing columns and rows. This is because the failing columns and rows intersect not only at the defect site but also at some defect-free electrodes, which are referred to as *incorrectly-classified defect sites*. This problem is illustrated in Figure 4.12.

The above problem can be solved by carrying out binary search for each column/row that fails the test, as shown in Figure 4.13. This method eliminates the likelihood of incorrectly-classified defects false and helps us to precisely locate the actual defect sites. However, it suffers from the drawback that precise defect localization is not possible when there are “untestable sites” in the array, a problem described next.

4.2.2 Untestable Sites

An untestable site is defined as an electrode that cannot be traversed by any test droplets in the parallel scan-like test method. A site becomes untestable when there are defects in both its row and column, and in all four directions, i.e., North, South, East, and West, as shown in Figure 4.14. In this case, test droplets are impeded by the defects and they cannot reach the untestable site.

The above problem can be addressed by carrying out one more iteration of parallel scan-like test. As in column and row test, we referred to this test iteration as diagonal test, as shown in Figure 4.15. In the additional test iteration, multiple test droplets are manipulated to traverse the array from one diagonal direction (top-left to bottom-right, or top-right to bottom-left) or both, referred to here as *single-diagonal test* and *cross-diagonal test*, respectively. Untestable sites that cannot be reached from vertical and horizontal directions can be reached diagonally. Even though this approach cannot guarantee the testability of all the untestable sites, it significantly reduces the probability that an electrode site is untestable.

Another advantage of the diagonal test procedure is that it can also help in the avoidance of incorrectly-classified defect sites. Only the sites that lie on failing columns, failing rows, and failing diagonals are identified as defects. Although this method does not completely eliminate the likelihood of incorrectly-classified defects, it provides acceptable diagnostic resolution by eliminating most incorrectly-classified defects. These advantages are highlighted quantitatively in Section 4.3.2

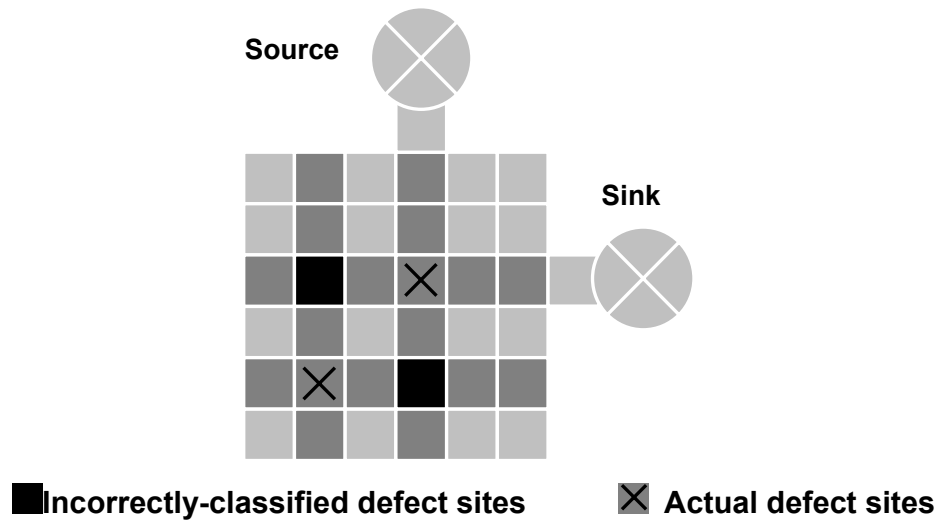


Figure 4.12: An example of incorrectly-classified defects.

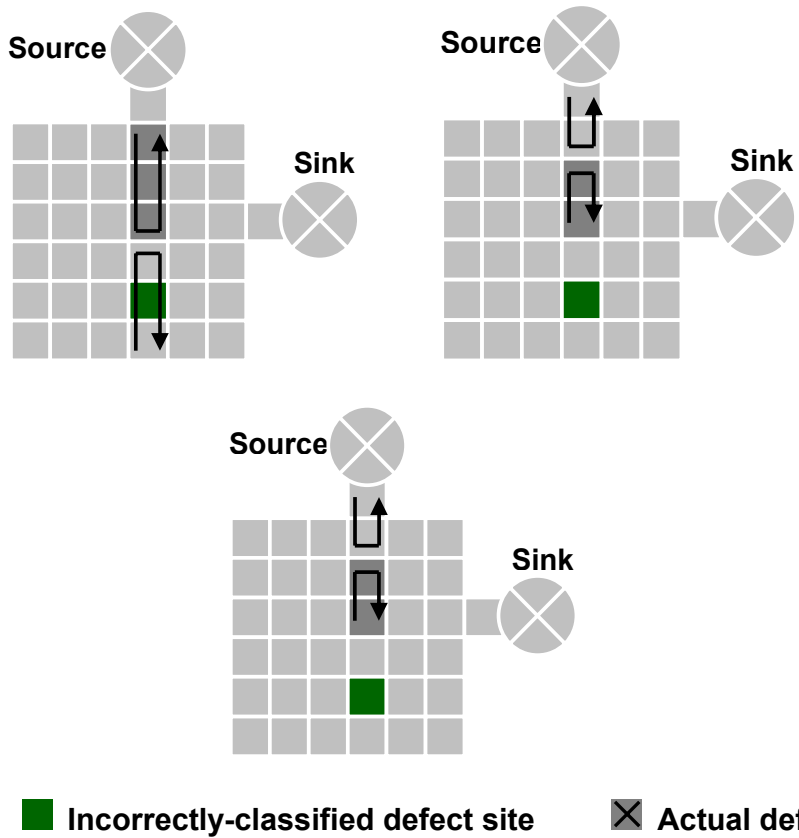


Figure 4.13: Elimination of incorrectly-classified defects using binary-search-based diagnosis.

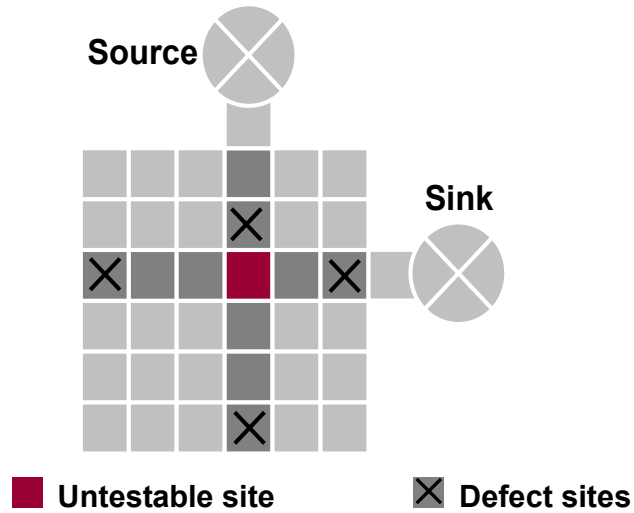


Figure 4.14: An example of an unstable site.

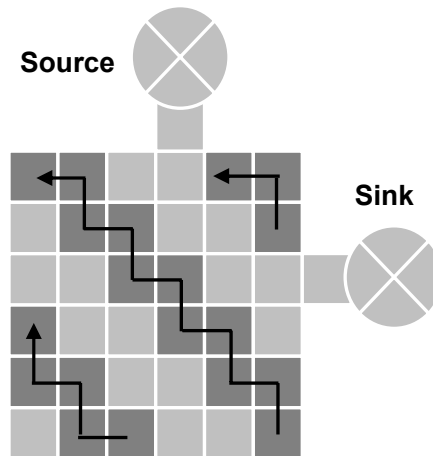


Figure 4.15: Illustration of diagonal parallel scan-like test.

4.3. Performance Evaluation

In this section, we evaluate the performance of the proposed testing and diagnosis method. We first carry out complexity analysis of parallel scan-like testing and compare it with results obtained from the Euler-path-based method [36]. Next, we use probabilistic analysis to evaluate the improvement in diagnostic resolution obtained using the proposed technique for locating multiple defects.

4.3.1 Complexity Analysis

We first calculate the complexity of parallel scan-like testing. For simplicity we only discuss off-line test of a target array that contains a single defect. The parallel scan-like test method is based on three stages, i.e., peripheral test, column-test and row-test.

To test a $N \times N$ target array, peripheral test is first carried out, and this stage takes which take $4N$ steps. Each step is defined as a droplet manipulation from one electrode to another, which takes 1 second for a typical actuation frequency of 1 Hz. Next, column- and row-tests are carried out, each takes N steps. Thus the total test procedure includes $8N$ steps, i.e., $O(N)$. Fault diagnosis is based on one-dimensional binary partitioning, therefore it is also $O(N)$. Compared to Euler-path-based method, which has $O(N^2)$ complexity, the time needed for both testing and diagnosis are significantly reduced.

To make a more practical comparison, we apply the proposed parallel scan-like test method and the Euler-path-based method to the off-line testing of microfluidic arrays with sizes varying from 10×10 to 50×50 electrodes. Note that the complexity for both the proposed method and the Euler-path-based method is independent of defect location. Thus for each size, a sample array with a randomly injected faulty cell is generated as a target array. To get the precise test time, we calculate the time needed to route the droplet from source to the pseudo source and from pseudo-sink to the sink reservoir, and add to the test time derived using the above complexity analysis. The results are shown in Figure 4.16.

As predicted by the complexity analysis, the test time for the Euler-path-based testing increases quadratically with the array size, while the parallel scan-like test time increases only linearly. A significant improvement can be seen for large arrays.

4.3.2 Probabilistic Analysis

Next we calculate the probability of the occurrence of incorrectly-classified defects, i.e.,

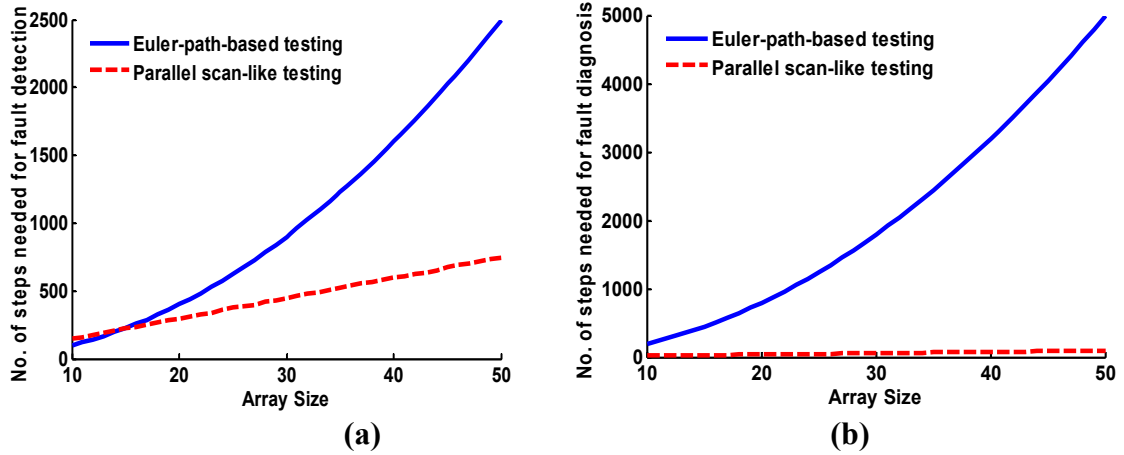


Figure 4.16: Comparison of (a) testing complexity, and (b) diagnosis complexity (additional steps) of parallel scan-like test and Euler-circuit based method.

the probability of an electrode to be a candidate defect site when it is not defective. Assume that each electrode fails independently with probability p . The probability that an electrode is defect-free is therefore simply $1 - p$. An electrode is a candidate defective electrode if there is an actual defect in either the same column or the same row as this electrode; see Figure 4.17. Therefore, the probability P_{cand} that a defect-free electrode is classified as a candidate defect site in an $N \times N$ array is given by:

$$P_{\text{cand}} = (1-p)(1-(1-p)^{N-1})^2$$

For $p \ll 1$, and large N , we get:

$$\begin{aligned} P_{\text{cand}} &= (1-p)(1-(1-p)^{N-1})^2 \\ &\approx p^2 (1-p) (N-1)^2 \\ &= p^2 (1-p) N^2 \end{aligned}$$

When diagonal testing is carried out, a defect-free electrode is classified as a candidate electrode only if the following conditions hold: (i) there is a defect in the same column as the electrode; (ii) there is a defect in the same row as the electrode; (iii) there are defects on one

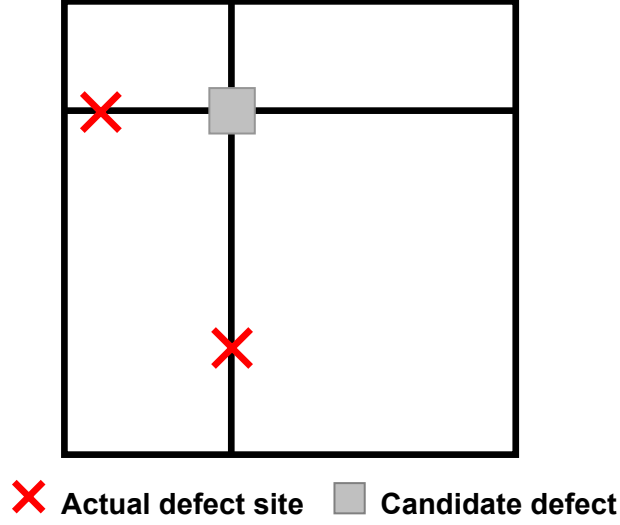


Figure 4.17: Illustration of a candidate defect (incorrectly classified).

or both the diagonals on which the electrode lies. These situations are illustrated in Figure 4.18(a) and Figure 4.18(b).

If single-diagonal diagnosis is carried out as in Figure 4.18(a), the probability $P_{\text{incorrect}}$ of an incorrect classification for an electrode is given by:

$$P_{\text{incorrect}} = (1-p)(1-(1-p)^{N-1})^2(1-(1-p)^{N-1+\min\{i,j\}+\max\{i,j\}})$$

If cross diagonal diagnosis is carried out as in Figure 4.18(b), the probability of false defect is

$$(1-p)(1-(1-p)^{N-1})^2(1-(1-p)^{N-1+\min\{i,j\}-\max\{i,j\}})(1-(1-p)^{N-1+\min\{N-i,j\}-\max\{N-i,j\}})$$

Using these equations, we calculate the probability of false defect occurrence under different probability of defect occurrence. The results are shown in Figure 4.19.

In Figure 4.19, it can be seen that a significant increase in diagnostic resolution, i.e., the ratio of the number of actual defects to the number of classified defects, is achieved by carrying out single-diagonal diagnosis. Further improvement can be achieved when cross-diagonal diagnosis is applied. However, the difference in the results for cross-diagonal diagnosis and single-diagonal diagnosis is less apparent for smaller values

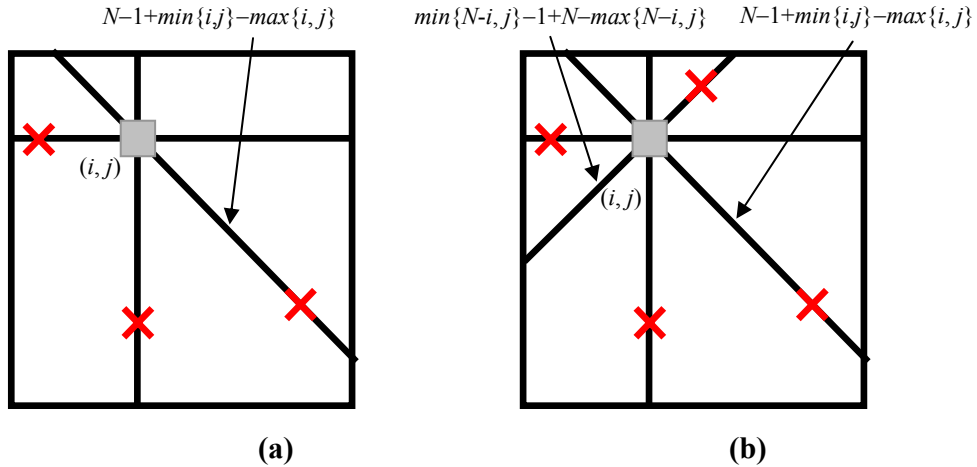


Figure 4.18: Illustration of the conditions that lead to incorrect classification of candidate defects for (a) single-diagonal diagnosis, (b) cross-diagonal diagnosis.

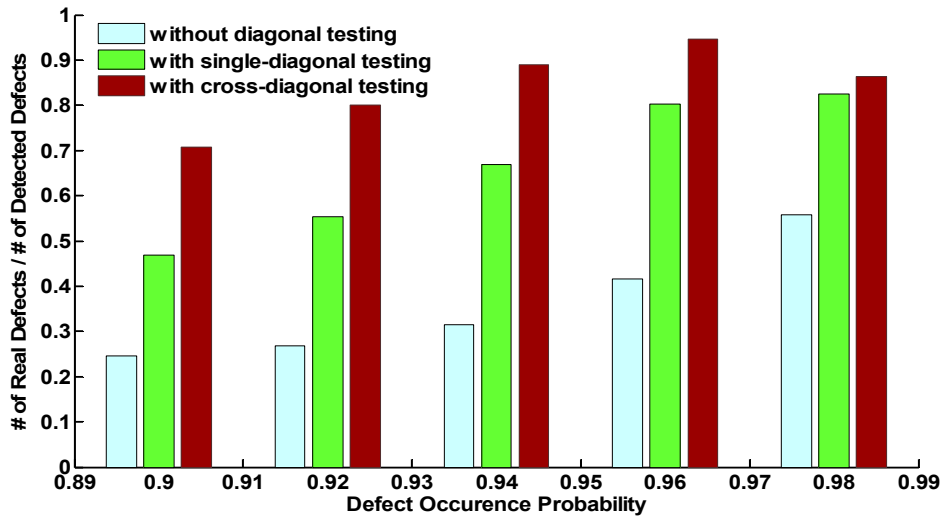


Figure 4.19: Simulation results highlighting the likelihood of incorrect classification.

of the defect occurrence probability.

4.3.3 Occurrence Probability of Untestable Sites

Next we analyze the probability of the occurrence of untestable sites. Again assume each electrode is failing with the same probability of p . The electrode is untestable if there is one real defect in each of its four directions, as shown in Figure 4.14. Therefore, for a single

electrode in row i and column j , the probability P_{untest} that it cannot be tested is:

$$P_{\text{untest}} = (1-(1-p)^{i-1})(1-(1-p)^{N-i})(1-(1-p)^{j-1})(1-(1-p)^{N-j})$$

When diagonal testing is carried out, the electrode is untestable only if there is also a real defect in the same diagonal or diagonals as shown in Figure 4.20(a) and Figure 4.20(b).

If single-diagonal diagnosis is carried out as in Figure 4.20(a), the probability P_{untest} of an untestable electrode is

$$P_{\text{untest}} = (1-(1-p)^{i-1})(1-(1-p)^{N-i})(1-(1-p)^{j-1}) \cdot \\ (1-(1-p)^{N-j})(1-(1-p)^{\min\{i,j\}-1}) \cdot \\ (1-(1-p)^{N-\max\{i,j\}}).$$

If cross-diagonal diagnosis is carried out as in Figure 4.20(b), the probability P_{untest} of an untestable electrode is given by:

$$P_{\text{untest}} = (1-(1-p)^{i-1})(1-(1-p)^{N-i})(1-(1-p)^{j-1}) \cdot \\ (1-(1-p)^{N-j})(1-(1-p)^{\min\{i,j\}-1})(1-(1-p)^{N-\max\{i,j\}}) \cdot \\ (1-(1-p)^{\min\{N-i,j\}-1})(1-(1-p)^{N+\max\{N-i,j\}}).$$

Using the above formulas, we calculate the probability of false defect occurrence under different probability of untestable electrode occurrence. The results are shown in Figure 4.21. We see that diagonal testing leads to a significant reduction in the probability that a cell is untestable. Even though the proposed multiple-defect diagnosis method does not guarantee the testability of all electrodes, it reduce the occurrence probability of untestable sites to almost zero (less than 0.0001).

4.4 Application to a Fabricated Biochip

In this section, we apply the parallel scan-like test method to a fabricated biochip. The chip-under-test is a PCB microfluidic platform for DNA sequencing, as shown in Figure

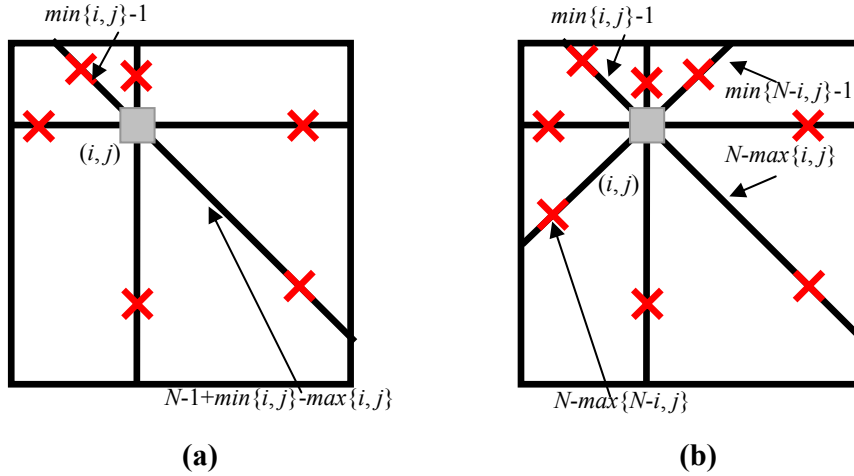


Figure 4.20: Illustration of untestable sites for (a) single-diagonal diagnosis (b) cross-diagonal diagnosis.

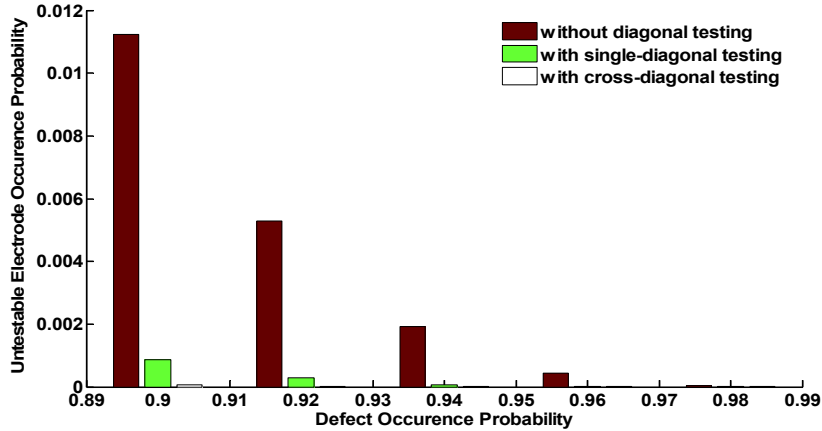


Figure 4.21: Simulation results highlighting the probability of untestable sites.

4.22. The platform consists of a 7×7 array, 8 reservoirs and routing electrodes that connect reservoirs to the array. A total of 9 cells are reserved for grounding, and they are not available for droplet transportation.

As a baseline, we first apply Euler-path-based testing to this chip. The test procedure takes 57 seconds, assuming a (typical) 1 Hz electrode-actuation frequency. Next we carry out parallel scan-like test (the column-test stage is shown in Figure 4.23). Since 9 electrodes are not reachable, the testing of even columns and rows are not need. The test application procedure takes 46 seconds, again for a 1 Hz actuation frequency.

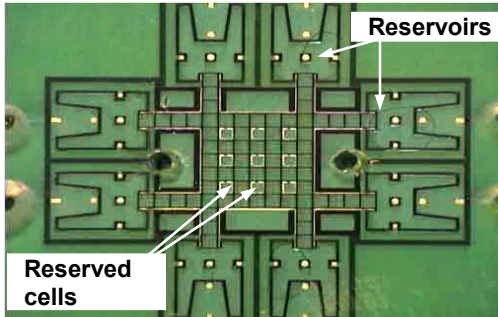


Figure 4.22: Fabricated biochip for DNA sequencing.

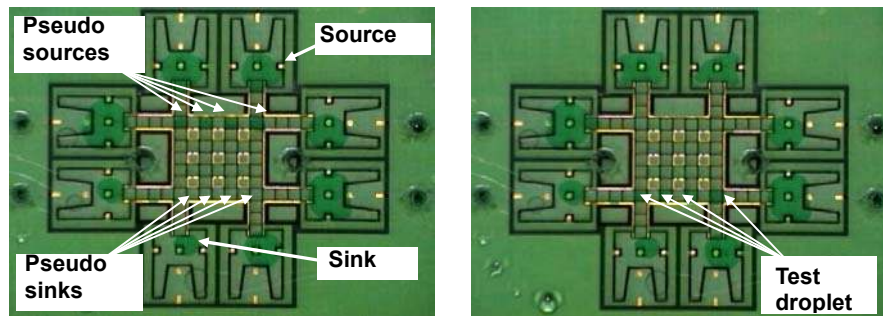


Figure 4.23: Column-test step of parallel scan-like test.

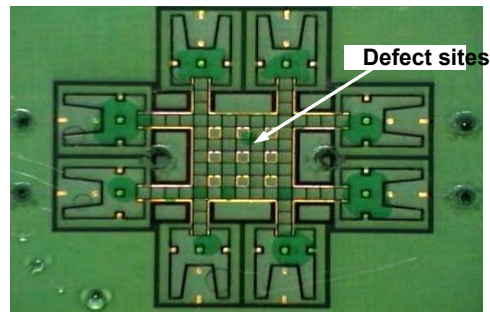


Figure 4.24: Parallel scan-like diagnosis of single cell defect.

Next we study the time needed for fault diagnosis for the two methods. We use a fabricated chip, which is known to contain one defect *a priori* (determined by inspection and electrical measurements). The chip with the defect is shown in Figure 4.24. For the Euler-path-based method, we carried out a binary search to locate the defect cell. A total of seven iterations are needed and the total diagnosis time is 173s. This value is obtained by summing up the times for the different diagnosis iterations which are 57s, 44s, 32s, 16s, 8s, 4s, and 2s, respectively. On the other hand, parallel scan-like test can simply determine the

defect site from testing readouts. No additional diagnosis steps are needed and the diagnosis time is the same as the testing time, i.e., 44s, which correspond to a 75% reduction compared to [36].

4.5 Functional Test

Since complicated fluidic operations are repeatedly executed with high precision in compact microfluidic arrays, a group of cells is repeatedly required to perform a large number of operations. Traditional structural test methods, which use test droplets to traverse the target array, do not provide a sufficient level of confidence that these fluidic operations can be reliably performed on the array [28]. For instance, some unit cells, i.e., electrodes, may function correctly during droplet transportation, but they might malfunction during droplet dispensing from reservoirs. Likewise, unit cells that can be reliably combined to operate as a mixer may malfunction when they are used for droplet splitting. Moreover, a structural test does not cover non-reconfigurable modules such as capacitive sensing circuits. A defect involving any of the modules may result in catastrophic failure during bioassay execution. Therefore, before we use synthesis methods to map bioassay protocols to a microfluidic array [15], it is important to carry out functional testing to verify the integrity of the underlying microfluidic platform. To ensure that manufactured biochips are competitive in the emerging low-cost market for disposable biochips and to avoid electrode degradation resulting from excessive actuation, test methodologies should be inexpensive, quick, and effective.

In this subsection, we first present various defects that are typical for digital microfluidic biochips. We relate these defects to logical fault models that can be viewed not only in terms of traditional shorts and opens, but which also target biochip functionality. Based on these fault models, we introduce the idea of functional testing of digital microfluidic modules. We

develop cost-effective functional test methods that target fluidic operations such as droplet dispensing, droplet transportation, mixing, and splitting. These methods also test the functionality of non-reconfigurable modules such as capacitive sensing circuits. The proposed methods allow functional testing using parallel droplet pathways in both on-line and off-line scenarios. For each function, the proposed approach identifies “qualified regions”, i.e., groups of cells that pass the test. Instead of placing fluidic modules in a fault-oblivious manner on the microfluidic array, synthesis tools can map modules only to qualified regions. In this way, the reliability of the synthesized biochip is significantly increased. The application of these methods to pin-constrained biochips is also discussed. We evaluate the proposed functional test methods using simulations as well as experiments for a fabricated biochip.

To avoid ambiguity, we refer to the test method proposed in Section 4.1 as structural test, since they route test droplets to all the electrodes in the array to ensure structural integrity. Structural test targets physical defects, which are defined as the underlying cause of erroneous chip operation, where the defect affects either a unit cell (electrode) or the electrical connection to the unit cell. Functional testing, on the other hand, involves test procedures to check whether groups of cells can be used to perform certain operations, e.g., droplet mixing and splitting. For the test of a specific operation, the corresponding patterns of droplet movement are carried out on the target cluster of cells. If a target cell cluster fails the test, e.g., the mixing test, we label it as a malfunctioning cluster, which implies that the synthesis tool cannot place the corresponding module—a mixer in this case—in this region.

As in the case of structural testing, we first develop a fault model for functional testing. Since functional testing is an extension of structural testing, all the defect-oriented fault models for also included here. Therefore, we start from the fault models proposed in Section 4.1. Malfunctions in fluidic operations are then identified as shown in Table 4.2.

Table 4.2. Fault models for digital microfluidic biochips.

Cause of malfunction	Malfunction type	*	Fault model	Observable error
Electrode actuation for excessive duration	Irreversible charge concentration on the dispensing electrode	3	Dispensing-stuck-on (droplet is dispensed by not fully cut off from the reservoir)	No droplet can be dispensed from the reservoir
Electrode shape variation in fabrication	Deformity of electrodes	3	No overlap between droplets to be mixed and center electrode	Mixing failure
Electrode electrostatic property variation in fabrication	Unequal actuation voltages	3	Pressure gradient (net static pressure in some direction)	Unbalanced volumes of split droplets
Bad soldering	Parasitic capacitance in the capacitive sensing circuit	1	Oversensitive or insensitive capacitive sensing	False positive/negative in detection

* **Number of cells involved in the defect.**

Next we propose efficient functional test methods to detect the defects and malfunctions listed in Table 4.2. Dispensing test, mixing test, splitting test, and capacitive sensing test are developed to address the corresponding malfunctions. A routing test procedure is used to detect all physical defects.

4.5.1 Dispensing Test

Dispensing test targets the malfunctioning of the dispensing operation. Figure 4.25 provides a comparison between normal dispensing and an example of dispensing failure. As shown in Figure 4.25(b), the dispensed droplet in a malfunctioning scenario cannot be detached from the droplet in the reservoir. Therefore, when we move the dispensed droplet away from the reservoir, an additional droplet from the reservoir is extracted and moved as well. In this case, the dispensed “droplet” can be several times larger than the normal size, which may result in the catastrophic failure of a volume-sensitive bioassay.

Here we propose a test method based on capacitive sensing to detect these dispensing failures. The circuit diagram for fault detection is shown in Figure 4.2. It has been shown in the literature that dispensing involves a reservoir and the three electrodes closest to it [67].

We therefore define every reservoir together with its three neighbor electrodes as a *dispensing cluster*. The third electrode in the cluster, i.e., the electrode farthest from the reservoir, is connected to a capacitive sensing circuit for test readout, see Figure 4.26. When the test starts, a droplet is dispensed from the reservoir until it reaches the third electrode. We expect a positive pulse with a normal amplitude for both fault-free dispensing and dispensing failure. Next, we route the dispensed droplet one electrode in the forward direction. During correct dispensing, the fully-dispensed droplet moves completely to the fourth electrode, thereby no pulse is detected by the capacitive sensing circuit output at this time. However, if the droplet undergoes a dispensing failure and it is still connected to the liquid in the reservoir, there must be some fluid left at the third electrode, which is indicated by a positive pulse, with smaller amplitude, in the test readout. Therefore, we can easily detect a dispensing failure by reading the output of the capacitive sensing circuit, as shown in Figure 4.27.

To identify abnormal droplets, two threshold values for the pulse amplitude are used. These thresholds are determined through calibration of the sensing circuit. First, we fix a nominal value μ and a maximum allowable droplet volume variance σ . Then two droplets with volume of $\mu + \sigma$ and $\mu - \sigma$ are routed to the sensing circuit. Signal levels are recorded and used as the upper and lower threshold values respectively.

4.5.2 Routing Test and Capacitive Sensing Test

Routing test focuses on evaluating a single electrode's ability to transport droplets. This procedure is similar to that proposed earlier for structural test [28,34]. In structural test, a test droplet is dispensed and routed to cross the target electrode from two orthogonal directions, i.e., along the row and the column directions. The routing problem can be solved by mapping the array to an undirected graph and applying the Euler-path-based method [36], as shown in Figure 4.28. On the other hand, a test droplet must be routed along all four directions relative to the target electrode. We can solve the route planning problem in this case by mapping the

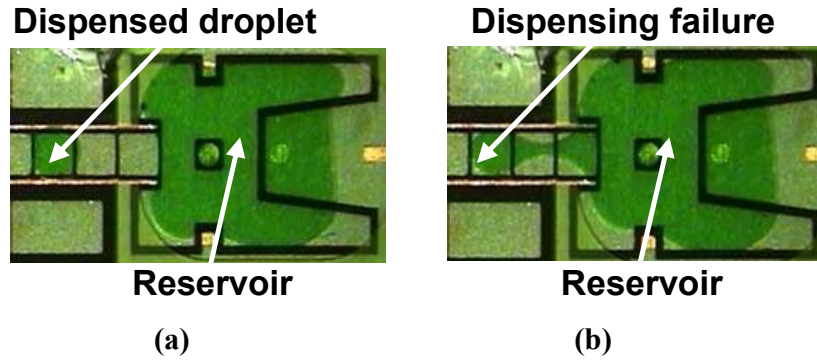


Figure 4.25: Illustration of (a) normal dispensing and (b) dispensing failure, for a fabricated microfluidic biochip.

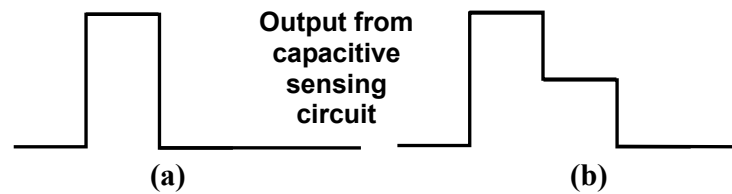


Figure 4.27: Test readouts for (a) normal dispensing and (b) dispensing failure.

target array to a directed graph, which can be easily derived by replacing every edge in the undirected graph with two directed edges in opposite directions. The Euler-path based method is then applied to the directed graph to derive a test plan for routing test.

Note that in structural test, a test droplet is routed to traverse the array, following the Euler path derived from the undirected graph in Figure 4.28(a). As a result of this procedure, half of the directed edges in Figure 4.28(b) are also traversed, with exactly one edge visited for each pair of directed edges between two nodes. The edges that are not visited can be ordered to form a “reverse path” corresponding to the Euler path derived above. Therefore, routing test can be carried out by applying two iterations of structural test in opposite directions. Recall that all the defects listed in Table 1 can be detected by structural test; therefore, they are also detected by the above routing test procedure.

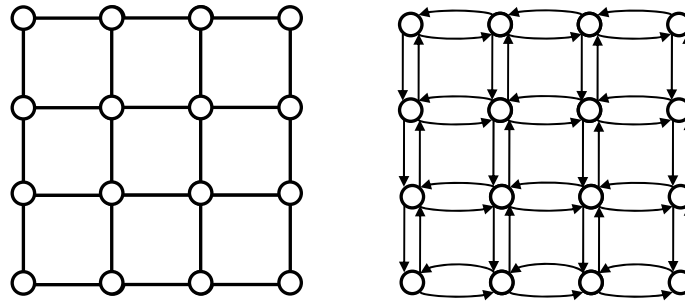


Figure 4.28: (a) An undirected graph for Euler-path-based structural test; (b) The corresponding directed graph for Euler-path-based routing test.

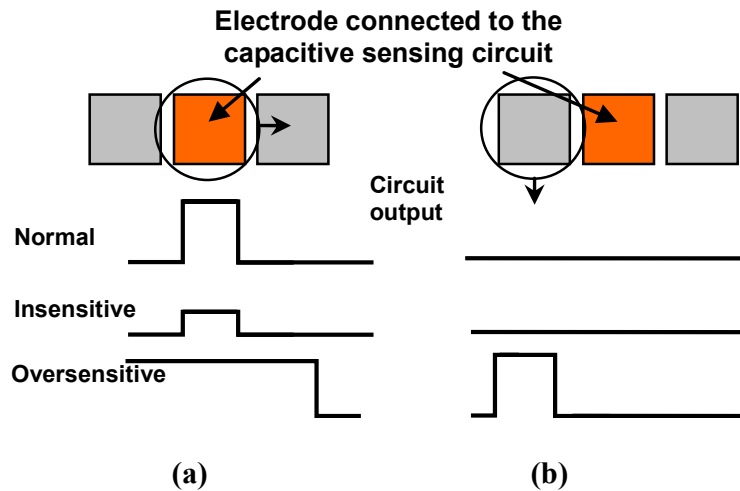


Figure 4.29: Test outcomes for the capacitive sensing circuit.

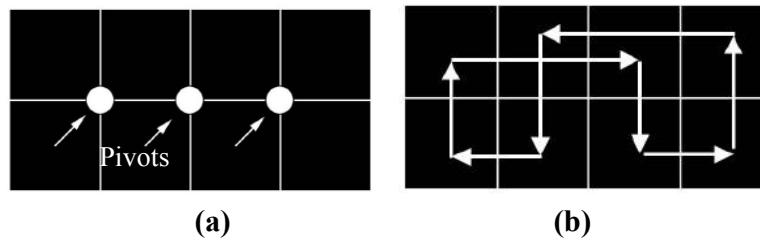


Figure 4.30: (a) Pivots and (b) routing path for a 2x4 microfluidic mixer.

The above test procedure also tests the functionality of the capacitive sensing circuit. If a test droplet is routed to visit the electrode connected to the target capacitive sensing circuit, a positive pulse is expected at the output of the sensing circuit. By examining the amplitude of the positive pulse, we can determine whether a capacitive sensing circuit is normal, insensitive, or oversensitive, as shown in Figure 4.29.

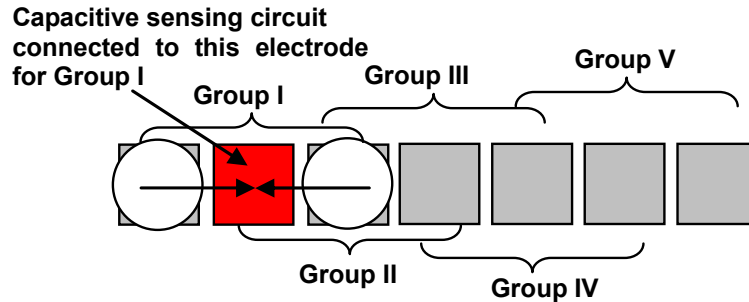


Figure 4.31: Example of merging test.

4.5.3 Mixing and Splitting Test

Next we discuss the functional testing of two widely used microfluidic modules—mixers and splitters. In a digital microfluidic biochip, two droplets are mixed within a cluster of electrodes, referred to as mixer. Even though mixer designs and configurations vary considerably [49], the underlining mixing mechanisms remain the same for all designs and configurations. Two droplets are merged at one electrode and routed to move about some pivots in the mixer, as shown in Figure 4.30.

Thus a mixing functional test is equivalent to the testing of the merging and routing operations within the target cell cluster. Recall that droplet routing test has been addressed in Section 4.5.2; therefore, a mixing test can be reduced to a droplet merging test, which checks a series of three adjacent electrodes to determine whether two droplets can be merged on them. For a microfluidic array, a simple test method carries out droplet merging on every group of three adjacent electrodes, one at a time. For such a three-electrode test, the test outcome is read out using a capacitive sensing circuit connected to the center electrode, on which droplets are supposed to be merged, as shown in Figure 4.31. However, since every electrode can be the center of a set of three-electrodes, we have to connect a capacitive sensing circuit to it, which results in high production cost. Moreover, the serial processing method requires a large number of droplet manipulation steps and electrode actuations. As shown in Table. 4.2, excessive actuation may result in a variety of catastrophic defects.

Therefore, efficient algorithms are needed for droplet-merging test.

Droplet splitting is simpler compared to mixing. The fluidic operation involves three adjacent electrodes. By applying an appropriate electrode-actuation sequence, a droplet that rests on the center electrode is split into two smaller droplets, which rest on the two side electrodes. Thus a split operation can be viewed as reverse of droplet merging. Consequently, splitting test can be carried out by applying the merging test methods in a reverse manner. The only difference lies in the fact that instead of connecting a capacitive sensing circuit to the center electrode, splitting test attaches two capacitive sensing circuits to the two side electrodes. The test outcome is evaluated by comparing output amplitudes of the two sensing circuits.

We next combine these two tests into a unified test application procedure. We start from the simple case where mixing and splitting test are carried out for two three-electrode groups centered at one electrode. For simplicity we limit our discussion to linear merging and splitting, i.e., the electrodes involved are linearly aligned in the same row/column. The test procedure is illustrated in Figure 4.32.

In Figure 4.32, we carry out mixing and splitting test using four steps, i.e., horizontal splitting, horizontal mixing, vertical splitting and vertical mixing. Note that the ordering is carefully chosen such that the four steps can be carried out consecutively, with no additional routing steps needed in between. However, this procedure still requires every electrode to be connected to a capacitive sensing circuit. Moreover, in order to extend this test scheme to a microfluidic array, we need $4N^2$ manipulation steps for an $N \times N$ array of electrodes, which is very inefficient.

To achieve higher test efficiency and lower hardware cost, we apply the single-electrode test methods in parallel for array testing. The key idea is to carry out mixing and splitting test for all the electrodes in a row/column concurrently. For simplicity of analysis, we demonstrate the method using a directed graph, where each electrode is mapped to a node in

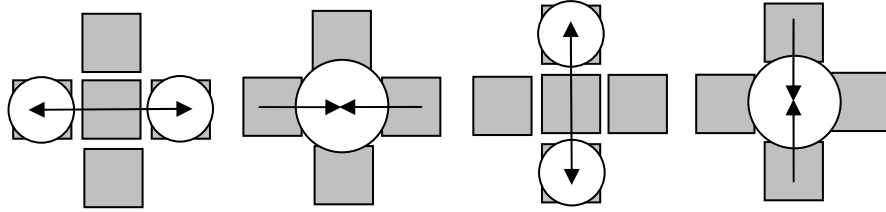


Figure 4.32: Mixing and splitting test for all the groups of three electrodes that are centered on a given electrode.

the graph, and each mixing or splitting test step is represented by a pair of directed edges; see Figure 4.33.

The electrodes in Figure 4.33 are labeled as being either “even” or “odd”. We carry out the horizontal splitting test for all the even electrodes concurrently. The split droplets get merged at the odd electrodes, therefore the merging test is done at the same time. Similarly, by carrying out the splitting test for all the odd electrodes concurrently, we can easily complete the horizontal merging test for all the even electrodes. Thus we can carry out all the horizontal tests (merging and splitting) in one row using only two manipulation steps. Similarly, all the vertical tests in one column can be completed in two manipulation steps.

Following the above observations, we propose a parallel procedure to carry out mixing and splitting tests efficiently. The steps of the procedure are as follows:

1. Route large droplets to all the even electrodes in a row.
2. Carry out splitting test for all the even electrodes concurrently (large droplets are now on odd electrodes).
3. Carry out splitting test for all the odd electrodes concurrently (large droplets are now on even electrodes).
4. Route the droplets consecutively to a capacitive sensing circuit for test readouts.
5. Repeat the test procedure for the next row.
6. Repeat steps 1-5 for columns.

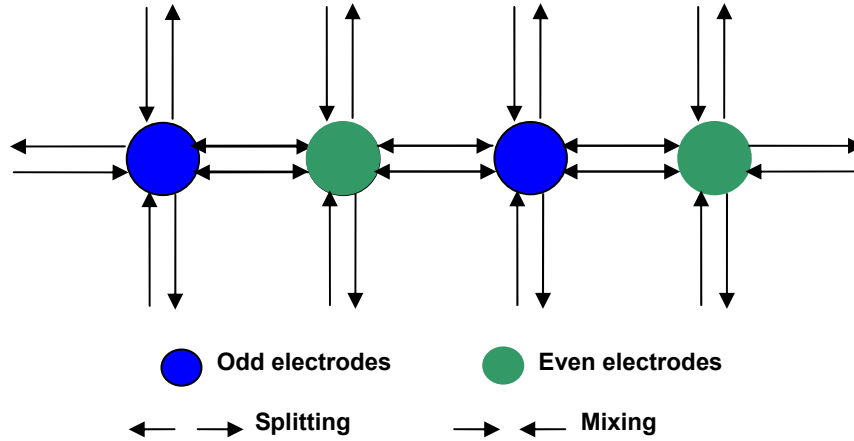


Figure 4.33: Parallel mixing and splitting test for a row of electrodes.

Note that in the above method, only one capacitive sensing circuit is used, therefore the hardware cost is significantly reduced. However, additional droplet routing steps are needed. In order to minimize the number of droplet manipulations, test results are read out not directly after each splitting or merging test, but after the both of them are carried out, as shown in the above steps. As a trade-off, a more complicated test-result interpretation scheme is required.

If all the tests in one row are executed without the detection of a malfunction, droplet volume should be almost the same. However, if a malfunction occurs, volume variation is expected, as shown in Figure 4.34.

In Figure 4.34, the shaded droplet undergoes an unbalanced split during the splitting test. Since all other droplets are split evenly, this malfunction results in a pair of test droplets of abnormal volume, one bigger and the other smaller. If the next step of test yields no malfunction, the droplet volume variation is propagated one electrode away. Therefore, we can easily detect the malfunction by checking the test results.

The proposed test method achieves high efficiency. An implicit assumption here is that adjacent electrodes are not defective. Such defects can be detected by separate structural test [36]. For an $N \times N$ array, only $N^2 + N$ manipulation steps are needed, while the test method in prior work [36] requires $4N^2$ steps. Moreover, the method uses only one capacitive sensing

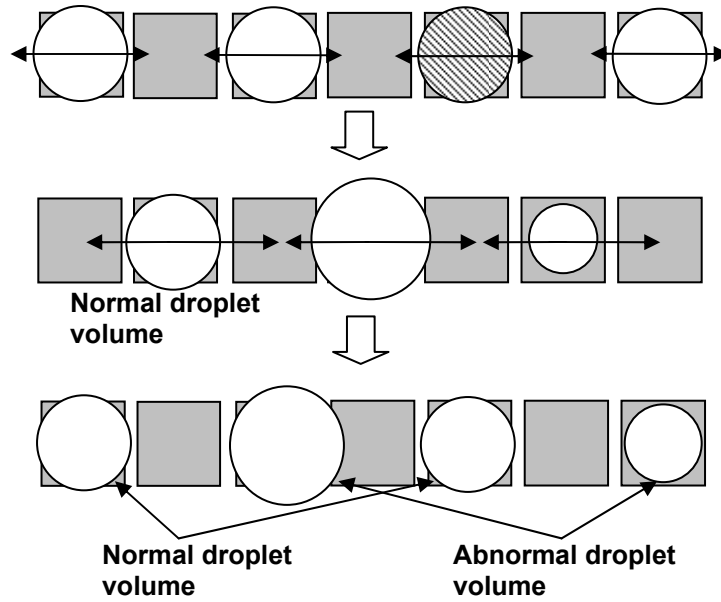


Figure 4.34: Parallel mixing and splitting test for a row of electrodes.

circuit, irrespective of the array size. This is in contrast to [36], which requires N^2 capacitive sensing circuits for an $N \times N$ microfluidic array. The potential reduction in production cost is therefore significant.

4.5.4 Application to Pin-Constrained Chip Design

In the discussion of the functional test method in Sections 4.5.1-4.5.3, we have assumed that the chip is controlled using the direct-addressing method. In this subsection, we investigate the application of the functional test method to pin-constrained biochips. Four different pin-constrained biochip prototypes designed using the techniques mentioned in Chapter 3 are discussed.

An n -phase chip

In the n -phase chip, every n th electrode is electrically connected and controlled using a single control pin. Therefore, every n th electrode must be activated/deactivated simultaneously, as shown in Figure 4.35. Due to this constraint, the mixing and splitting test cannot be carried

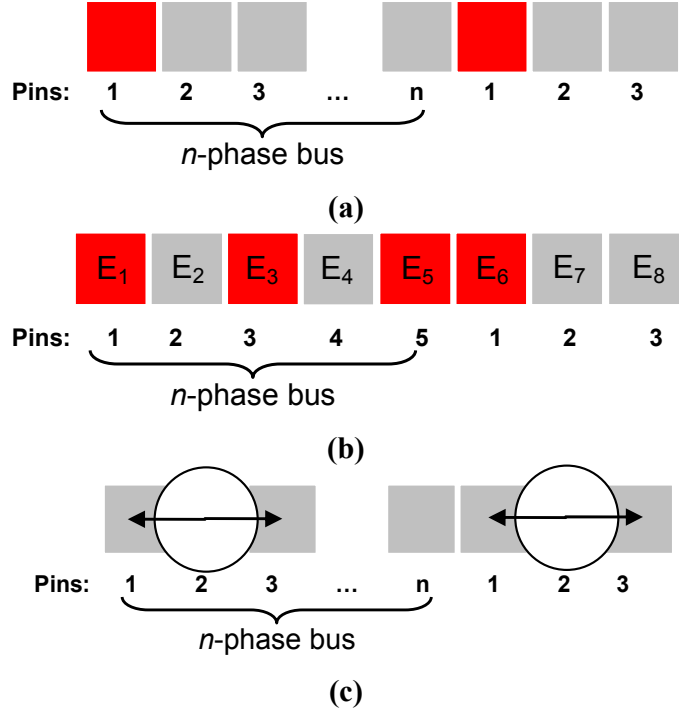


Figure 4.35: Functional test on an n -phase-bus chip.

out currently on every other electrode following the six steps presented in Section 4.5.3. In the example of Figure 4.35(b), a linear array consisting of eight electrodes is controlled using 5-phase bus. To carry out splitting test on every other electrode, e.g., E_1 , E_3 and E_5 , the control pins 1, 3 and 5 need to be activated currently to hold the test droplets on these three electrodes. However, since electrode E_6 is connected to Pin 1, it is also activated. This causes unintentional splitting of the test droplet on E_5 .

To avoid this problem, the proposed functional test procedure needs to be modified. For a microfluidic array controlled using an n -phase bus, current mixing and splitting tests are only allowed on every n th electrode. Since these electrodes are connected to the same control pins, carrying out splitting or mixing test on these electrodes requires the same pattern of pin activations. As shown in Figure 4.35(c), to execute a splitting test on both E_2 and E_7 requires Pin 1 and Pin 3 to be activated and Pin 2 to be deactivated. Therefore, carrying out the splitting/mixing test on these electrodes concurrently will not lead to unintentional fluidic operations.

Cross-referencing-based chip

The proposed functional test method can also be applied to cross-referencing-based chips. The test procedure is the same as the one described in Section 4.5.3. As shown in Figure 4.36, to execute the splitting test for all the even electrodes in second row, test droplets are first routed to all the odd electrodes. Next, the second row pin and all the odd column pins are activated. This also completes the mixing test for all the odd electrodes. In the final step, the row pin and all the even column pins are activated, thereby completing the splitting test for all the odd electrodes and mixing test for the even electrodes.

Array-partitioning-based chip

In Section 3.1, a biochip array is divided into several partitions based on the droplet routing results. In each partition, the “Connect-5” pin-assignment algorithm is used to control the electrodes in it using five pins, as shown in Figure 4.37. Such a pin-assignment algorithm allows free movement of a single droplet in the partition. However, only one droplet movement is allowed. Concurrent manipulation of more than two droplets within a single partition will lead to unintentional operations. Therefore, no operations that involve two or more droplets, e.g., mixing and splitting, can be carried out within a single partition. In fact, these operations can only be executed using the electrodes on the boundary of two partitions. Therefore, only the electrodes on the boundary need to be tested for such malfunctions.

Note that since only one droplet is allowed in each partition, while a mixing or splitting test is carried out on one electrode, e.g., E_1 , on the boundary of two partitions (Partition 2 and Partition 4) in Figure 4.37, no other electrodes in the same partition, e.g. E_3 , can be tested at the same time. As a result, boundary electrodes can only be tested one by one and the concurrency of the functional test is compromised. Fortunately, the total length of the

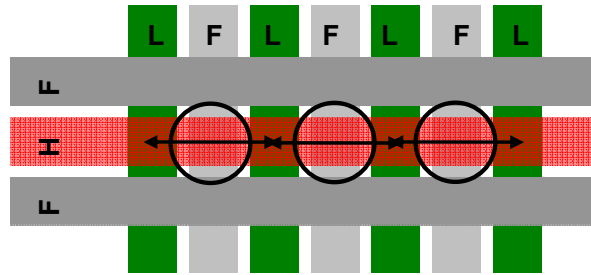


Figure 4.36: Functional test on cross-referencing-based chip.

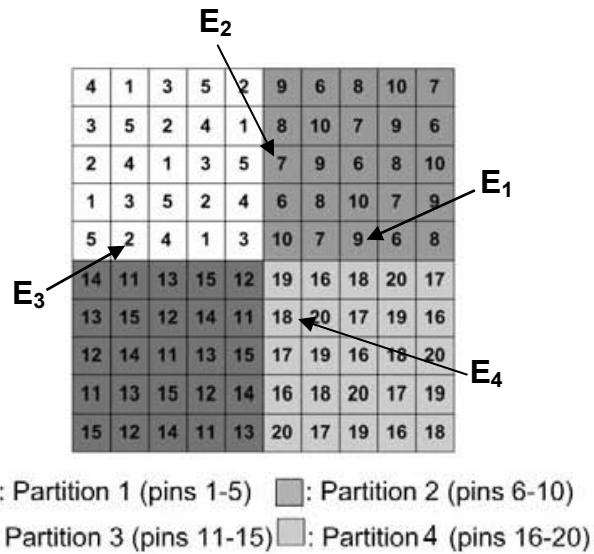


Figure 4.37: Functional test on array-partitioning-based chip.

boundaries on a chip is bounded by $2 \times N$ for an $N \times N$ array. Only a limited number of iterations of functional test are needed. Moreover, it can be seen from Figure 4.37 that the functional test of electrodes on different boundaries of different partitions, e.g., E_1 on the boundary of partition 2 and partition 4 and E_3 on the boundary of partition 1 and partition 3, uses completely different sets of control pins. Thus these tests can be carried out concurrently. This increases the parallelism of the functional test. Moreover, a higher test frequency (i.e., droplet activation rate) can be used to shorten the test time.

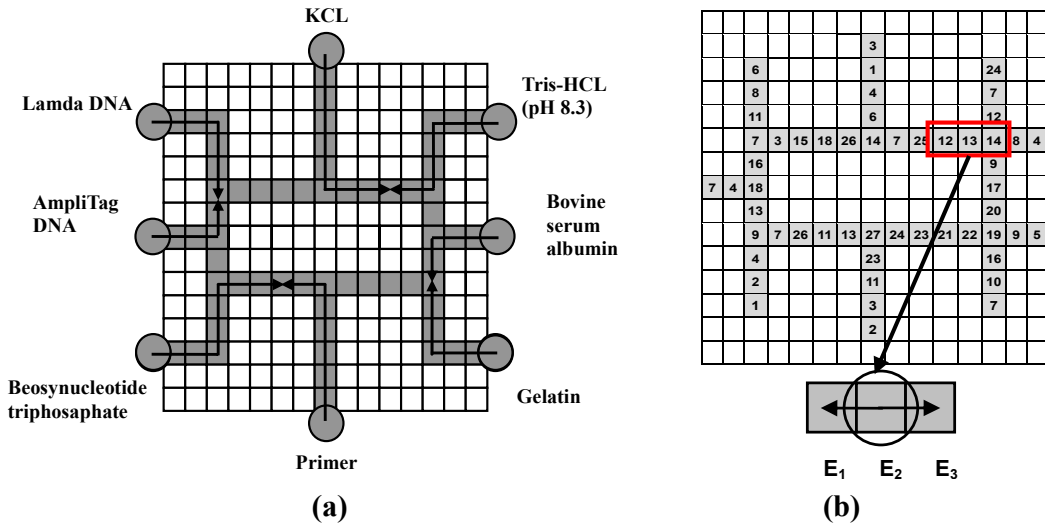


Figure 4.38: Functional test for broadcast-addressing-based chip (a) layout and droplet routes for the multiplexed-assay chip (b) broadcast-addressing-based pin-assignment result.

Broadcast-addressing-based chip

Next we apply the functional test method to a broadcast-addressing-based chip. Due to the constraints introduced by the pin-assignment results, not all the cells on the chip can be tested. For example, mixing and splitting test cannot be applied in the highlighted area in Figure 4.38(b). To mix two droplets seated on E₁ and E₃ at the electrode E₂, controlled by highlighting control pin 13, since the electrode above E₃ will also be activated while we move the droplet from E₃ to E₁ and split it.

However, as discussed in Section 3.3, a broadcast-addressing-based chip is designed to execute for a predetermined set of known bioassays. We know exactly where the mixing and splitting operations will be carried out. So there is no need to test for malfunctions on other cells on the chip. Since the number of electrodes to be tested is very limited, only a small number of test steps are needed. Again a higher test frequency can be used to reduce test time. To further increase concurrence, we can use the algorithm presented in [63] to check the compatibility of droplet movements during functional test.

4.6 Experimental and Simulation Results

In this section, we apply the proposed functional test methods to a fabricated chip. The chip-under-test is a PCB microfluidic platform for the Polymerase Chain Reaction (PCR), as shown in Figure 4.39. The platform consists of two columns and two rows of electrodes, three reservoirs, and routing electrodes that connect the reservoirs to the array.

We applied the proposed functional test methods to this chip. Dispensing test and routing test are trivial due to the simple structure of the chip. Therefore, we only focused on the mixing and splitting tests. Following the steps in Section 4.5.3, we first targeted the bottom row and dispensed five test droplets to the odd electrodes, as shown in Figure 4.40(a). Then splitting test of the even electrodes was carried out. Droplets were split and merged on the even electrodes. In Figure 4.40(b), we see a series of droplets of the same volume resting on the even electrodes, which means that all the odd electrodes passed the splitting test, and merging at the even electrodes worked well. However, when we carried out the splitting test on the even electrodes, a large variation in droplet volume was observed on the 3rd and 5th electrodes; see Figure 4.40(c). This variation implied a malfunction, leading to unbalanced splitting on the 4th electrode. The malfunction was detected when the droplets were routed to the capacitive sensing circuit. We then labeled the 4th electrode on the bottom row as an unqualified splitting site so that synthesis tools will not map a splitter to it. Thus the system robustness of the synthesized design was enhanced.

We next evaluate the improvement in system robustness using a biochip for the protein dilution assay described in Chapter 2. Again, a 10×10 microfluidic array is used to execute the assay.

Next we evaluate system dependability of the synthesized design for three cases: (i) no testing is carried out; (ii) only structural test is carried out; (iii) functional test is carried out. A design is deemed to fail if any module suffers from a defect or a malfunction, e.g., a mixer suffering from a faulty mixing site.

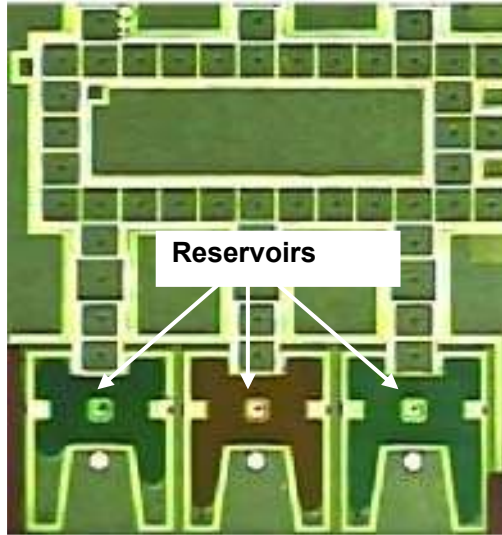
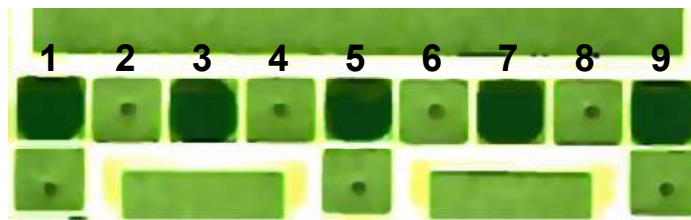
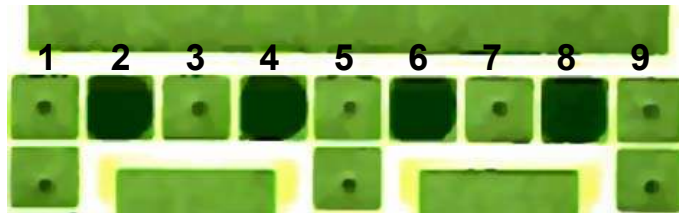


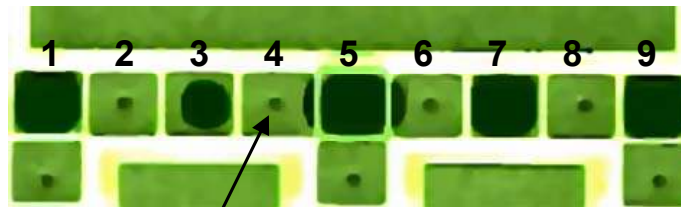
Figure 4.39: Fabricated biochip for PCR.



(a)



(b)



Malfunction
(unbalanced splitting)

(c)

Figure 4.40: Mixing and splitting test for a fabricated PCR chip (bottom row of Figure 4.39).

We generate 200 simulated samples of faulty arrays. Each faulty array is derived by randomly injecting fault (due to defects and malfunctions) in the array. Note that we do not specify the types of injected malfunctions. Here we simply assume that all the injected malfunctions can be detected by the proposed functional test and cannot be detected through structural test. Let A be the event that a unit cell has a defect. We let the defect probability $p = P[A]$ take two values, namely 0.01 and 0.05. Let B be the event that a group of electrodes suffer from a malfunction. Since a defective unit cell leads to a malfunction of the module where it is used, we are interested here in the conditional probability $q = P[B | \bar{A}]$, i.e., the probability that a module is malfunctioning even if there is no defect in it. We consider four values of q , namely 0.02, 0.04, 0.06, 0.08. For each faulty array, the structural test method from [36] is carried out to detect and locate defect sites.

Next the synthesis method from [15] is used to map the protein array on to defect-free regions of the array. We also use functional test to detect and locate malfunctions in the array. These malfunctions are then bypassed during the synthesis of the chip for the protein array. As a baseline, we also carry out the synthesis for an array to which neither structural test nor functional test have been applied.

First we determine the failure rate R , $0 \leq R \leq 1$, for the three scenarios when the protein assay is mapped to an array with defects and malfunctions. When functional testing is carried out, the failure rate due to target defects and malfunctions is zero because all of them are detected by the test procedure. If no testing is carried out, the failure rate is as high as 0.85, i.e., the protein assay fails for as much as 85% of the 200 simulated chips. If structural testing is used, the failure rate is lower, but it is still significant—as high as 0.75.

Figure 4.41 shows that as the malfunction probability increases, the failure rate R becomes considerable even when structural testing is used. Moreover, the benefits of structural testing are less evident for the smaller value of the defect probability, i.e., $p = 0.01$. Therefore, functional testing is needed to augment droplet-transportation-based structural

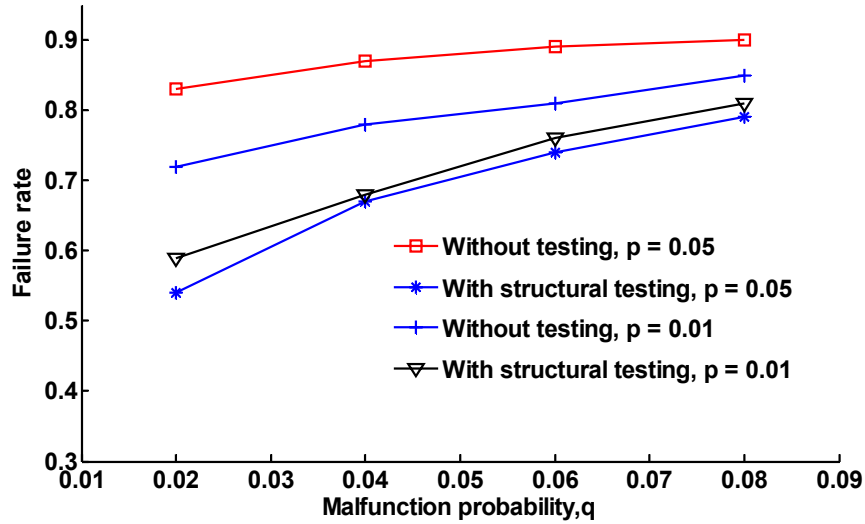
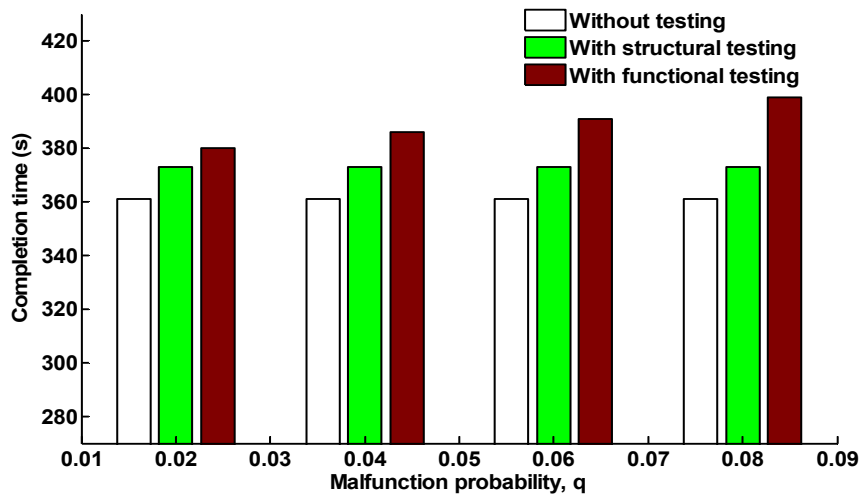


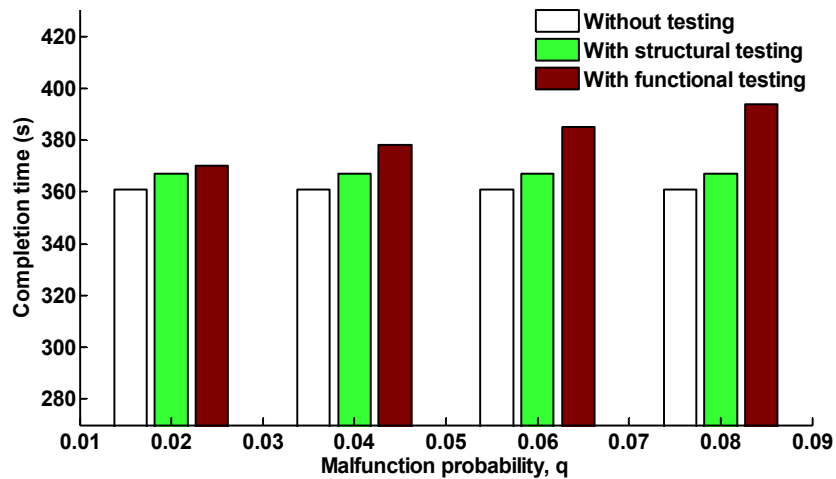
Figure 4.41: Failure rate for synthesized design without testing and with structural testing.

testing for digital microfluidic arrays. A counter-intuitive finding from Figure 4.41 is that the failure rate is lower for $p = 0.05$ compared to $p = 0.01$. This occurs because large p implies that there is low likelihood of a defect-free cell. Hence structural testing is likely to catch such defects.

The detection of more defects and malfunctions using functional testing leads to a corresponding increase in the assay completion time. This happens because fault detection and fault location leave fewer unit cells available for the protein assay, and the synthesis procedure employs less parallelism in the execution of the microfluidic operation. Figure 4.42 shows the assay completion time for the three scenarios that we are considering for the protein assay. As expected, the bioassay time is slightly higher when functional testing is used, and the increase is more for higher values of the malfunction probability q . This increase is acceptable because functional testing ensures that the assay will run to completion if the fluidic operations are mapped only on qualified region of the array.



(a)



(b)

Figure 4.42: Bioassay completion time for synthesized design without test, with structural test, and with functional test for defect occurrence probability of (a) $P[A] = 0.01$. (b) $P[A] = 0.05$.

Next we evaluate the functional test scheme on pin-constrained arrays. For each pin-constrained design method presented in Section 4.5.4, we choose one chip design prototype for discussion, as shown in Figure 4.43. Figure 4.43(a) shows a linear chip used for on-chip dilution. The chip is addressed using a five-phase bus. Figure 4.43(b) presents a 10×10 array for the multiplexed bioassay described in Chapter 2. Figure 4.43(c) provides an array-partitioning-based chip design for the multiplexed assay. The chip is divided into four

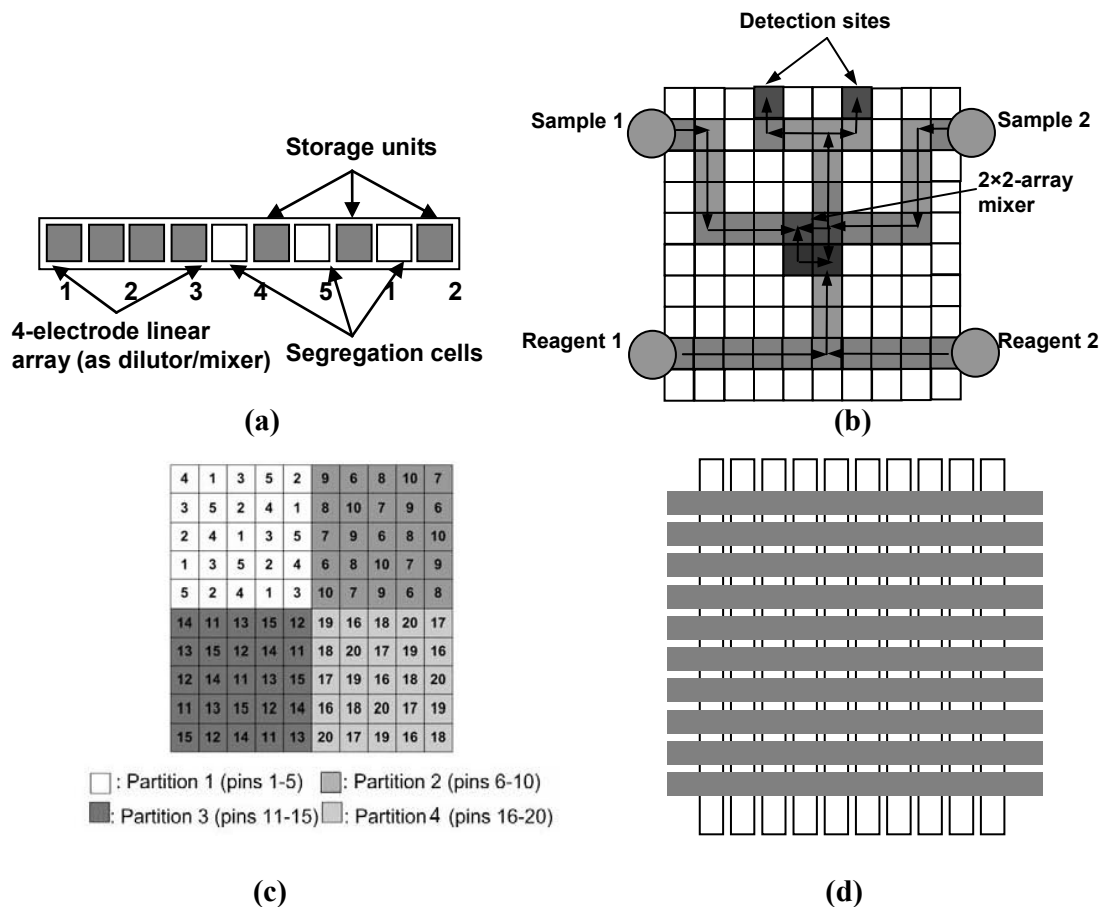
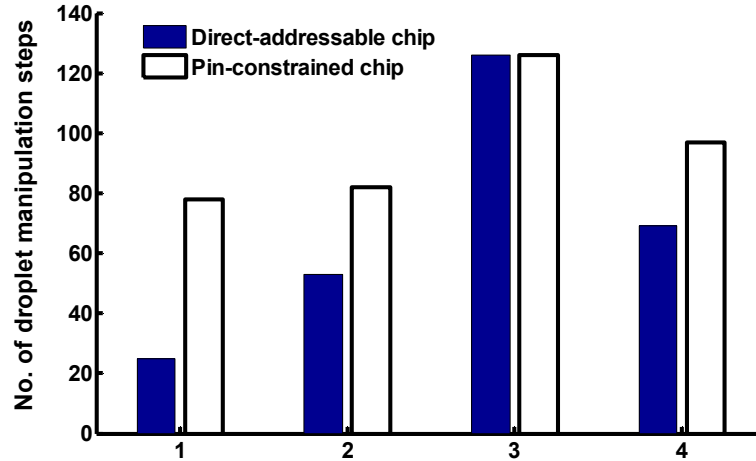


Figure 4.43: Pin-constrained chip designs for functional test evaluation: (a) a linear 5-phase-bus chip for dilution; (b) layout and droplet routes for the multiplexed-assay chip; (c) array-partitioning-based array for multiplexed assay; (d) cross-referencing-based array for multiplexed assay.

partitions, each partition is controlled using a dedicate set of pins. Figure 4.43(d) shows a cross-referencing-based chip design for the same assay. The simple design prototype is composed of 10 row pins and 10 column pins. Finally, for the broadcast-address method, the design shown in Figure 4.38 is used. The chip is designed for executing on-chip polymerase chain reaction (PCR) reactions.

Next we apply the extended functional test method proposed in Section 4.5.4 to these pin-constrained chips. The number of droplet manipulation steps are recorded and shown in Figure 4.44. Figure 4.44 also presents the number of manipulation steps required by the functional test assuming the chip is direct-addressable. Functional test on a cross-referencing-



- 1: *n*-phase-bus chip**
- 2: array-partitioning-based chip**
- 3: cross-referencing-based chip**
- 4: broadcast-addressing-based chip**

Figure 4.44: No. of droplet-manipulation steps for functional test of direct-addressable and pin-constrained chips.

based chip requires the same test time as for a direct-addressable chip. For other pin-constrained designs, the function test can still be carried out effectively, though with an increase in test application time. Considering the significant reduction in the number of control pins, such a compromise is acceptable. Moreover, the reduction in test concurrency can be avoided by increasing the test frequency.

Another important critical quality to measure the extended functional test method for pin-constrained designs is testability. As mentioned in Section 4.5.4, due to the connection-constraints, not all the electrodes on a pin-constrained array can be tested. Therefore, we define *testability* as the ratio of the number of testable electrodes over the total number of electrodes in the array. High testability indicates that the test method can probe the functionality of the chip thoroughly and identify a large number of qualified regions for a target application, which in turn contributes to increased flexibility for design and fault tolerance. Here we calculate testability for the four pin-constrained designs. Results are shown in Figure 4.45.

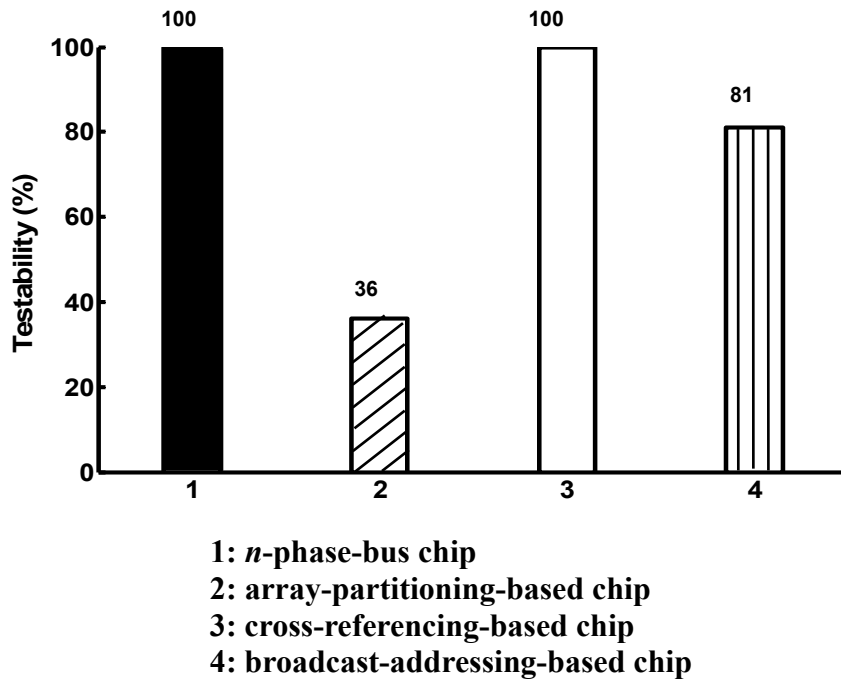


Figure 4.45: Functional testability of pin-constrained chips.

Figure 4.45 shows that the proposed functional test method achieves high testability (> 80%) on the *n*-phase chip, the cross-referencing-based array, and the broadcast-addressing-based array. The testability for the array-partitioning-based chip appears to be low. This is because of the fact that on an array-partitioning-based chip, the mixing and splitting operations are only allowed on the boundary cells. If we take this restriction into account, the testability is as high as 100%.

4.7 Chapter Summary and Conclusions

We have proposed an efficient test and diagnosis method for digital microfluidic biochips. The proposed method enables parallel testing using multiple test droplets for both on-line and off-line testing. We have also identified a number of common defects and defect types. These causes of defects have been related to fault models and observable errors. The proposed test and diagnosis method have been evaluated using complexity analysis and a fabricated chip

example. The results obtained demonstrate significant improvement over prior work on the testing and diagnosis of digital microfluidic biochips. We have also presented several techniques for the functional testing of droplet-based microfluidic biochips. These techniques address fundamental biochip operations such as droplet dispensing, droplet transportation, mixing, splitting, and capacitive sensing. Functional testing is carried out using parallel droplet pathways, and it leads to qualified regions where synthesis tools can map microfluidic functional modules. We have demonstrated functional testing for a fabricated biochip used for PCR. We have also presented simulation results for a protein assay, and quantified the small increase in assay completion time that is needed to achieve 100% coverage of the target defects and malfunctions with functional testing. Finally, we described the application of these functional test methods to pin-constrained chips.

Chapter 5

Design-for-Testability for Digital Microfluidic Biochips

Testing is essential for digital microfluidic biochips that are used for safety-critical applications such as point-of-care health assessment, air-quality monitoring, and food-safety testing. Recent research work has proposed a number of test methods for digital microfluidic biochips [28,34,36]. However, due to the fact that current design methods do not consider testability, the effectiveness of these test techniques is limited.

To tackle the above testability problem, we introduce the concept of design-for-testability (DFT) for biochips. The motivation of DFT for biochips is analogous to that for integrated circuits (ICs). In the early days of IC design, chip area and performance were the primary concerns for chip designers, and testing was only an afterthought. However, as chip complexity increased, test problems were greatly exacerbated and DFT became essential. Compared to the IC industry, digital microfluidics technology is still in its infancy. However, tremendous growth has been predicted for this technology and biochips for clinical diagnostics and cell sorting are now appearing in the marketplace [24,67]. As these devices become more complex, the need for DFT will be increasingly felt.

In this chapter, we provide a DFT solution to facilitate the testing of digital microfluidic biochips. We propose a test-aware design method that incorporates test procedures into the fluidic manipulation steps in the target bioassay protocol. By applying pin-constrained design to the testability-aware bioassay protocol, the proposed method ensures that the resulting chip layout supports the effective execution of test-related droplet operations for the entire chip. Therefore, the proposed DFT method allows design of pin-constrained biochips with a high level of testability with negligible overhead in terms of the number of control pins and assay completion time.

5.1 Testability of a Digital Microfluidic Biochip

Most recently proposed methods assume that the chip under test is a rectangular array controlled using a direct-addressing scheme, i.e., each electrode on the array is connected to an independent control pin. This method provides the maximum freedom for test-droplet manipulation, but it requires a large number of control pins.

To reduce production cost, unused electrodes are often removed from the rectangular array, resulting in an irregular chip layout; see Figure 5.1. To further reduce the number of control pins, pin-constrained design techniques are used in practice as discussed in Chapter 3, whereby multiple electrodes are connected to a single control pin. These design methods achieve a significant reduction in the number of input pins needed for controlling the electrodes. However, as a trade-off, droplet manipulation steps must satisfy additional constraints. These constraints can result in test procedures being either completely ineffective or effective only for a small part of the chip.

To evaluate the effectiveness of a test procedure, we define a parameter referred to as testability. Given a specific test method, the *testability* of a chip design is defined as the ratio of testable electrodes/functional units to the total number of electrodes/functional units on the chip, where a functional unit is defined as a cluster of adjacent electrodes that can carry out a specific type of fluidic operation. Depending on the test method and chip functionality, chip testability can be classified into two categories, namely structural testability and functional testability.

Structural testability is defined as the percentage of testable electrodes on the chip during a structural test. An electrode is considered “testable” if it can be traversed by the test droplet. Note that for most biochips, including pin-constrained chips, any on-chip electrode has to be traversed by at least one droplet in order to carry out the fluidic operations mapped on it. This means it can also be traversed by the test droplet. Therefore, most chip designs can achieve a structural testability of 100%.

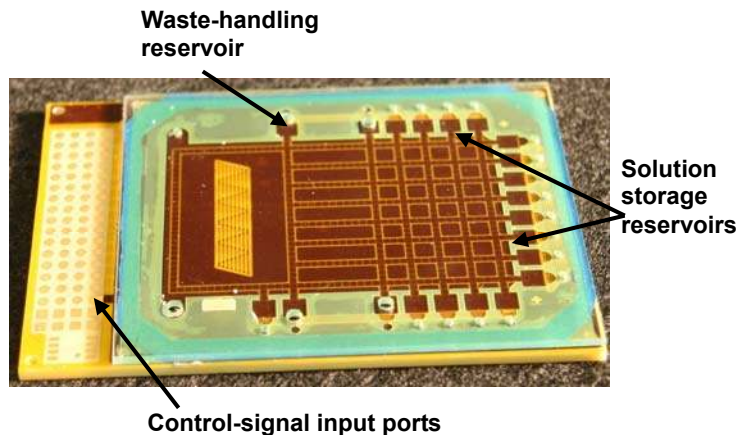


Figure 5.1: A fabricated digital microfluidic biochip [24].

Functional testability is defined as the percentage of testable functional units on a chip in a functional test procedure. High testability indicates that the test method can probe the functionality of the chip thoroughly and identify a large number of qualified regions for a target application, which in turn contributes to increased flexibility for design and fault tolerance. A functional unit is considered to be “testable” if the test-related droplets can be manipulated to carry out the target fluidic operations on it. These fluidic operations in the functional mode are always possible on a direct-addressable chip. However, for a pin-constrained chip, due to constraints introduced by the sharing of input control pins by electrodes, carrying out these functional-test operations on some functional units can result in the problem of electrode interference described in Section 3.1.

Figure 5.2 shows a pin-constrained chip design for a representative protein-dilution assay. The functional test procedure requires a splitting operation to be executed on the highlighted functional unit. To do this, we first activate Pin 13 to hold a test droplet at E_2 . Next, we deactivate Pin 13 and activate Pin 12 and Pin 14 to split the test droplet into two small droplets seated on E_1 and E_3 . However, E_4 is also charged by activating Pin 12. As a result, the split droplet that is supposed to be seated on E_2 will be moved unintentionally to the boundary of E_4 and E_3 .

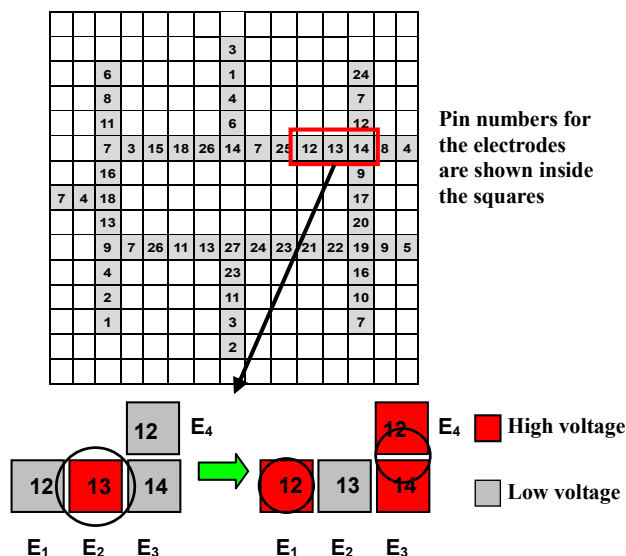


Figure 5.2: An example of an untestable functional unit on a pin-constrained chip for multiplexed assay.

Due to the above electrode interference problem, functional test cannot be applied to all the functional units in a pin-constrained chip design. Therefore, functional testability for a pin-constrained chip is usually less than 100%. Note that the reduction in testability is due to the conflicts between the fluidic operation steps required by functional test and the constraints on droplet manipulations introduced by the mapping of pins to electrodes. Different mappings for a pin-constrained chip lead to different untestable functional units, thereby different levels of chip testability. For example, the untestable functional unit shown in Figure 5.2 can be made testable by connecting electrode E_4 to a different control pin, e.g. Pin 15. Therefore, we can conclude that the key to increasing functional testability is to generate a test-friendly pin assignment that results in a small number of untestable functional units. To do this, test procedures must be considered early during chip design.

5.2 Testability-Aware Pin-constrained Chip Design

In this section, we propose a DFT solution to the functional testability problem described in Section 5.1.

5.2.1 Design Method

Our key idea is to incorporate fluidic operations required by functional test into the fluidic manipulation steps for the bioassay. Since these test-aware fluidic manipulation steps are provided as input to a pin-assignment design method, the resulting test-aware pin-constrained chip design guarantees a test-friendly pin assignment that supports all the fluidic operations required for functional test, thereby ensuring full testability.

The pin assignment in the proposed test-aware design method is based on the broadcast-addressing pin-constrained chip design technique presented in Section 3.3. In this design method, fluidic manipulation steps in a target bioassay, represented by the droplet schedule and droplet-routing steps, are stored in the electronic controller in the form of electrode-activation sequences. Each bit of the sequence represents the status of an electrode at a specific time-step. The status can be either “1” (activate), “0” (deactivate) or “X” (don’t care), which can be mapped to either “1” or “0”. For each electrode on the chip, its activation sequence can be represented using the above three values. Each sequence can contain several don’t-care terms, which can be replaced by “1” or “0”. By careful replacing these don’t-care terms, multiple activation sequences can be made identical. Therefore, the corresponding electrodes can be connected to a single control pin. The broadcast-addressing method achieves a significant reduction in the number of control pins, and the resulting pin assignment ensures the correct execution of all fluidic operations in the bioassay.

We first consider the functional test procedure as a separate bioassay. The fluidic operations required by the test procedure are derived from the scheduling and routing steps related to the test droplets. Next, we merge these fluidic operations with the droplet manipulation steps needed for the target bioassay. The merging can be carried out by attaching the electrode-activation sequences for the test procedure to the electrode-activation sequences for the target bioassays. For each electrode in the array, its activation sequence during the test procedure is added to that for the target bioassay to form a longer sequence. If these longer

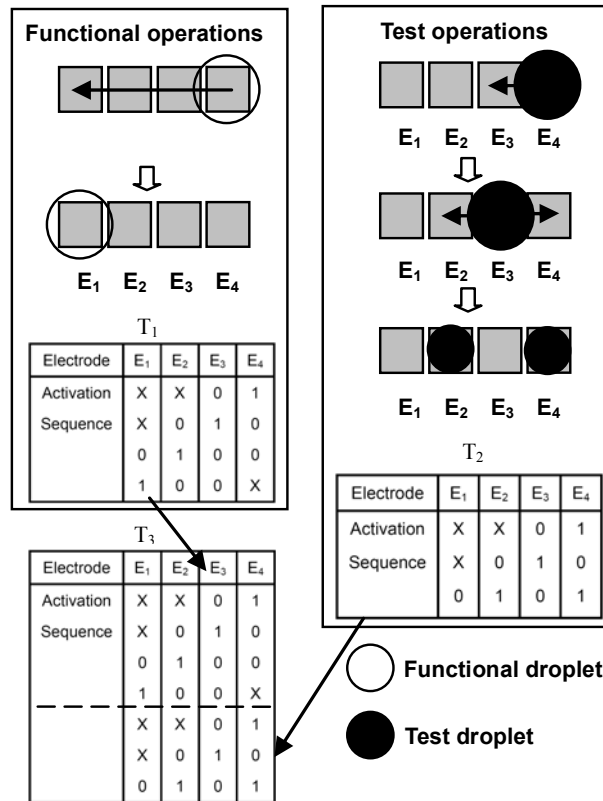


Figure 5.3: Illustration of the influence by adding test operations to the bioassay.

electrode-activation sequences are provided as input to the broadcast-addressing method, the resulted chip design will support not only the target bioassay but also the test operations.

We use an example to illustrate the details of the above DFT method. Figure 5.3 shows a linear array consisting of four electrodes. A simple “routing assay” is mapped to the array, where a droplet is to be routed from E₄ to E₁, one electrode per step. We first list the activation sequence for each electrode (Table T₁) in Figure 5.3. Next we add a splitting test on E₃. The electrode-activation sequences for the splitting test are shown in Table T₂ of Figure 5.3. These activation sequences are then combined with the activation sequences in T₁. The resulted longer activation sequences are listed in table T₃. The broadcast-addressing method is then applied to replace the don’t-care terms in T₃ and generate the eventual pin assignment.

Note that the addition of test operations into the bioassay may result in an increase number of control pins, compared to a test-unaware design using broadcast addressing. As shown in

Figure 5.3 (Table T₁), before the splitting test is added, we can map the two don't-cares in the activation sequence for E₁ with "10" and map the two don't-cares in the activation sequence for E₄ with "01" to make the two sequences identical. Therefore, the corresponding electrodes E₁ and E₄ can be connected to a single control pin. As a result, only three control pins are needed to control the linear array. However, when the splitting test is added, activation sequences in Table T₃ become incompatible. Therefore, they have to be controlled independently. The linear array now requires four control pins.

5.2.2 Euler-Path-Based Functional Test Method for Irregular Chip Layouts

The test operations used in the above testability-aware pin-constrained design method can be determined using the functional test method in Section 4.5. However, this approach requires a rectangular array structure for the chip under test. As discussed in Section 5.1, to reduce cost in practical designs, unused electrodes are often removed from the array, resulting in an irregular chip layout. Irregular layouts also result from the need for allocating routing tracks under the fluidic layer for connecting the electrodes to chip pins. In this subsection, we propose an Euler-path-based method for the functional testing of such irregular chip layouts.

For simplicity, we focus on the functional testing of two widely used microfluidic modules—mixers and splitters. According to the functional test method described in Section 4.5, a mixing functional test can be reduced to a droplet-merging test, which actuates a series of three adjacent electrodes to determine whether two droplets can be merged on them. A split operation can be viewed as the reverse of droplet merging. Consequently, these two tests can be combined into a unified splitting-and-merging test application procedure.

In a splitting-and-merging test for a single functional unit, a test droplet is routed to the center electrode of the three-electrode cluster, split, merged, and finally routed back to a detection site for test readout. To carry out mixing and splitting functional test for a chip, this

basic splitting-and-merging test needs to be carried out on every three-electrode cluster on the chip.

For a rectangular array, multiple splitting-and-merging tests can be carried out in parallel on a row/column of electrodes, as shown in Figure 5.5. However, parallel testing is not always feasible on an irregular-shape chip layout. Instead, the splitting-and-merging steps have to be carried out one at a time. All the functional units need to be targeted for full testability. However, overtesting must be avoided, i.e., a functional unit should not be tested repeatedly. To meet these criteria, we carry out the splitting-and-merging test along the Euler path of the chip layout.

Given an irregular chip layout, the Euler-path-based functional test method first maps it to an undirected graph and extracts the Euler path [36]. An Euler path traverses every edge in the graph exactly once. Next the mixing-and-splitting test is applied to the functional units along the Euler path, one at a time, until all the functional units are covered, as shown in Figure 5.6(a). Note that by following the above steps, the test droplet will traverse all the electrodes on the chip. Therefore, structural test is also accomplished.

To reduce the test-completion time, the functional units can be tested in groups. After the splitting-and-mixing test for a target functional unit is completed, the test droplet can be used to test the adjacent functional units instead of being routed to the source reservoir. Therefore, the test droplet is routed back to the source reservoir for test readout (Figure 5.6(b)) only after a group of functional units is tested.

The test efficiency depends on the size of a group of functional units being targeted by the same test droplet. A large group can be targeted to reduce test time. As a trade off, this group-testing scheme leads to reduced resolution for diagnosis. We can appropriately select the size of these groups to meet different test requirements.

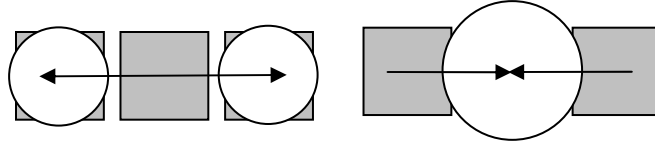


Figure 5.4: Mixing and splitting test for a functional unit.

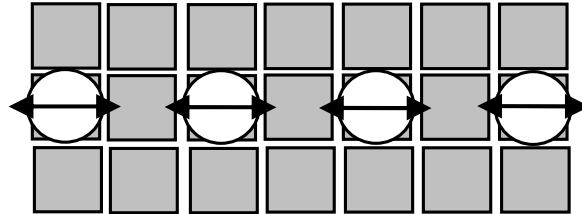


Figure 5.5: Parallel mixing and splitting test for a row of electrodes.

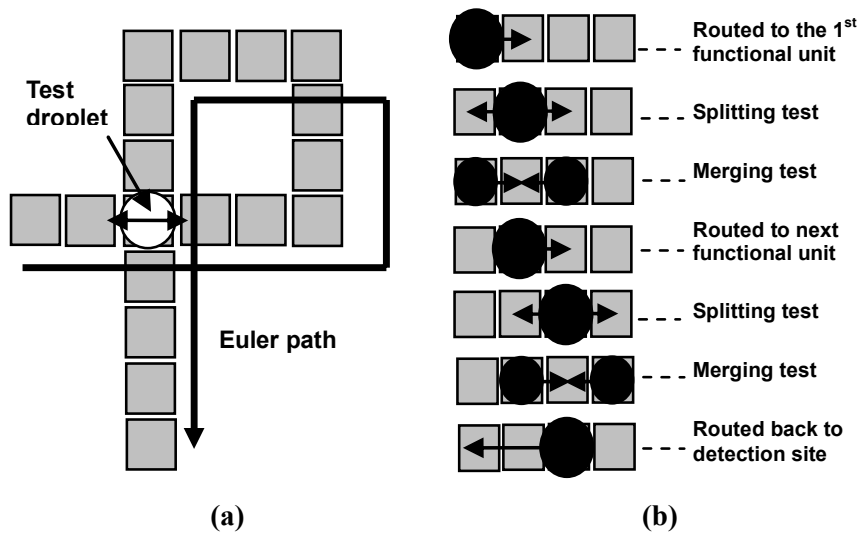


Figure 5.6: (a) Mixing-and-splitting test of functional units along the Euler path of a chip (b) Testing functional units in groups of two.

5.3 Simulation Results

In this section, we evaluate the proposed Euler-path-based functional testing method and the testability-aware design method by applying them to two target applications: a multiplexed immunoassay and the polymerase chain reaction (PCR) procedure.

Each assay is first mapped to a 15×15 electrode array controlled using the direct-addressing scheme. Unused electrodes are removed from the array, resulting in irregular chip layouts.

Next, the proposed Euler-path-based functional test method is applied to obtain a test plan for the chip. Finally, the test-aware design method is used to generate a pin-constrained design with a high level of testability.

5.3.1 Multiplexed Assay

We first map the multiplexed biochemical assay described in Section 3.4. This assay is used for in-vitro measurement and other antigens in human physiological fluids, which is of great importance for clinical diagnosis. For instance, a change in regular metabolic parameters in a patient's blood can signal organ damage or dysfunction prior to observable microscopic cellular damages or other symptoms. A portable, inexpensive biochip can be used for carrying out multiplexed bioassays for rapid and point-of-care diagnosis of such disorders. Flowchart for the multiplexed assays has been shown in Figure 3.34 in the form of a sequencing graph. A depiction of the droplet pathways for multiplexed glucose and lactase assays has been given in Figure 3.11.

Next we apply the proposed Euler-path-based functional test method to the above chip layout. To investigate the influence of the number of electrodes in each test group on the test frequency, five iterations of Euler-path-based functional test are carried out. Electrodes are tested in groups of 1-5, respectively. The test completion times (assuming test-droplet routing frequency of 10 Hz) are shown in Figure 5.7.

Figure 5.7 shows that a significant reduction in test completion time is achieved by increasing the number of electrodes in each test group. For example, by testing the electrodes in groups of five instead of testing one by one, the test completion time drops sharply from 332.8 seconds to 98.2 seconds, i.e., a 71% reduction. Note that as a trade off, whenever an error is observed, we can only determine a group of five candidate defective functional units. Nevertheless, even such coarse-grained diagnostic information is useful in practice for dynamic reconfiguration.

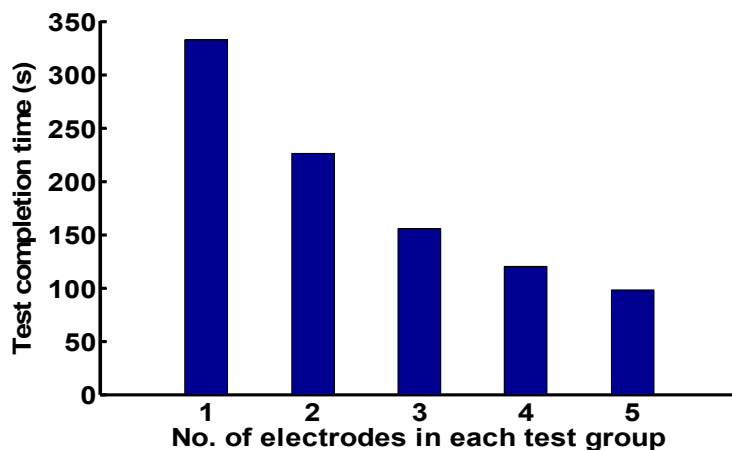


Figure 5.7: Comparison of test completion time for the Euler-path-based functional test method.

								12	2								
								18	17								
2	7	18						10	19						18	7	2
		9						9	6	18					9		
		3							21						3		
			25	2	24	6	7	8	26	16	19	1	5				
								1	15	14							
									26	6	7						
										21							
			11	20	10	22	2	23	5	20	19	22	25				
			5														17
			12														13
2	7	18													18	7	2

Figure 5.8: Pin assignment for the multiplexed assay chip obtained using the testability-aware design method.

Next we apply the test-aware design method to generate a pin assignment for the chip layout in Figure 3.11. The test-droplet routing sequences derived from the Euler-path-based functional test are combined with the bioassay schedule. The pin assignment results are generated as shown in Figure 5.8. We use the pin assignment generated obtained using the test-oblivious broadcast-addressing method for comparison, as has been shown in Figure 3.35.

As shown in Figure 5.8, the pin assignment resulting from the test-aware design method uses 26 control pins, i.e., only one more control pin than test-oblivious pin assignment (Figure 3.35). The test-aware design achieves 100% functional testability while the test-oblivious result achieves functional testability of only 76%.

5.3.2 Polymerase Chain Reaction (PCR)

For the second assay, we use the mixing stages of the PCR. These stages are used for rapid enzymatic amplification of specific DNA strands. Its assay protocol has been modeled by a sequencing graph in Chapter 3, as shown in Figure 3.37. Mapping the protocol on to the array, chip layout and schedule has been obtained as shown in Figure 3.38 and Figure 3.39, respectively.

Next we apply the proposed Euler-path-based functional test method to the above chip layout. Again, five iterations of Euler-path-based functional test are carried out. The test completion times are shown in Figure 5.9. Like in Figure 5.7, a significant reduction of test completion time is achieved as the number of electrode in each test group increases.

Next we apply the test-aware pin-constrained design method to generate pin-assignment for the chip layout in Figure 3.38. Pin-assignment results are generated as shown in Figure 5.10. We use the pin-assignment result obtained from the test-oblivious broadcast-addressing method for comparison; see Figure 3.40.

In Figure 5.10, the pin-assignment result generated from the test-aware design uses the same number of control pins as the test-oblivious pin-assignment result from the test-oblivious design method. However, test-oblivious result only allows a functional testability of 84%, while the test-aware achieves 100%. Combining these results with those from the multiplexed assay, we can conclude that by carefully rearranging the pin-assignment, the test-aware design method achieves a significant improvement in chip testability with a trivial increase in the number of control pins. Moreover, the pin-assignment result generated from the test-aware method supports the execution of the fluidic operations in the bioassay following the original droplet scheduling and routing result before testing is considered. Therefore, it guarantees the same bioassay completion time as that achieved by the test-oblivious pin-assignment result.

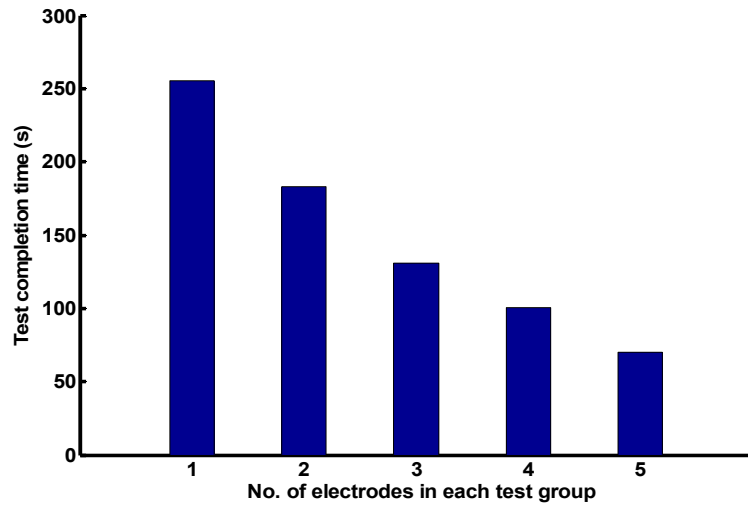


Figure 5.9: Comparison of test completion time for the Euler-path-based functional test method.

								5											
								12											
5	12	10						10						10	12	5			
		9						7						9					
		3						1						3					
		11	5	10	9	6	5	8	13	14	2	1							
		14												10					
5	4	1												4	8	5			
		3												13					
		12	7	9	2	8	1	6	9	11	12	1							
		5						4						17					
		8						11						7					
5	12	10						10						10	12	5			
								12											
								5											

Figure 5.10: Pin-assignment result for the PCR assay chip obtained using the test-aware design method.

5.4 Chapter Summary and Conclusions

In this chapter, we have introduced the concept of design-for-testability for microfluidic biochips. We have presented proposed a test-aware design methodology that allows the design of a pin-constrained biochip with 100% testability. An Euler-path-based functional test method, which extends the application of functional test on irregular chip layouts, has also been

presented. The proposed functional test method and the test-aware method have been evaluated by applying them to design and test biochips for a multiplexed bioassay and a PCR procedure. These DFT techniques form a bridge between biochip design and testing. They not only allow the design of easy-to-test biochip platforms but also broaden the application of the recently proposed test methods to cover more generic chip layouts.

Chapter 6

Application to Protein Crystallization

Protein crystallization is a commonly used technique for protein analysis and subsequent drug design. It predicts the three-dimensional arrangement of the constituent amino acids, which in turn indicates the specific biological function of a protein. As discussed in Section 1.3, protein crystallization experiments are typically carried out manually in well-plates in the laboratory. As a result, these experiments are slow, expensive, and error-prone due to the need for repeated human intervention.

In this chapter, we present the design of a multi-well plate microfluidic biochip for protein crystallization; this biochip can transfer protein samples, prepare candidate solutions, and carry out crystallization automatically [69]. To reduce the manufacturing cost of such devices, we adopt the “Connect-5” algorithm from Section 3.1 to generate a pin-assignment plan for the proposed design. The resulting biochip enables control of a large number of on-chip electrodes using only a small number of pins. Based on the pin-constrained chip design, we present an efficient shuttle-passenger-like droplet manipulation method to achieve high-throughput and defect-tolerant well loading. To facilitate preparation of crystallizing reagent solutions required by the assay, we also present an efficient algorithm to generate a preparation plan that lists the intermediate mixing steps needed to generate target solutions with the required concentrations [70].

6.1 Chip Design and Optimization

In this section, we present a multi-well plate design prototype for protein crystallization. As discussed in Section 1.3, to “hit” on the correct parameters for the crystallization of proteins, typically a very large number of experiments (10^3 - 10^4) are required. To achieve high

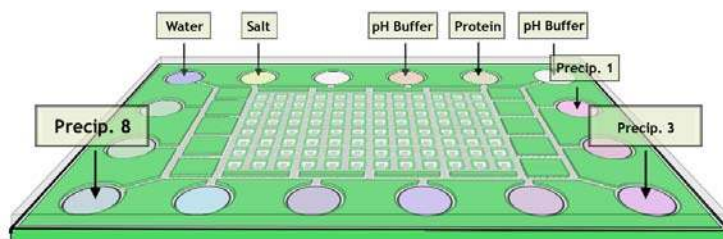


Figure 6.1: Schematic view of a 96-well chip that automatically sets up 96 reagent condition solutions [24].

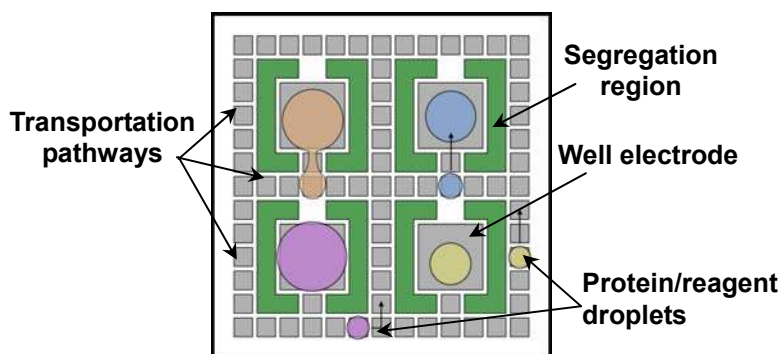


Figure 6.2: Schematic top-view of four wells and the surrounding electrodes [24].

efficiency, we use a multi-well plate design for parallel processing, as in microbatch crystallization. The schematic for the design is shown in Figure 6.1. The overall chip size is the same as that of a standard Society for Biomolecular Screening (SBS) multi-well plate. The chip has 96 wells and there are electrode pathways to connect these wells to reagent-loading and protein-loading loading ports.

Figure 6.2 shows the specific configuration of the wells. Note that unlike microbatch crystallization, where reagents and proteins are preloaded either manually or by robotics, here reagent and protein droplets are automatically transported along the pathways from their input loading ports to the wells. The rest of the chip real estate is used for accommodating the reagent and protein input wells. In addition to the protein reservoir that a user loads, there are two additional reservoirs that the user can load. These additional reservoirs can be loaded with any user-selected additives such as glycerol or detergents. Additives can stabilize the proteins and

there are numerous reports on the use of additives to improve the quality and size of protein crystals [71]. As we gain a better understanding of scaling issues, we will increase the number of wells on-chip as part of future work, since space (real estate) is available in the prototype design.

6.1.1 Pin-Constrained Chip Design

Next we assign control pins to address the electrodes in the proposed design. There are a total of 1284 electrodes in the chip, including electrodes in wells, transportation pathways, and reservoirs. If direct addressing is used, i.e., each cell of the patterned electrodes is accessed directly and independently via a dedicated control pin, a total of 1284 pins will need to be wired. However, the large number of electrodes required leads to a cumbersome wiring problem for control pins, especially when fabricated using PCB technology. In PCB technology, the diameter of the via hole is usually comparable to the electrode pitch size. Therefore, there is only a limited number of control lines that can be routed on one layer of PCB. As shown in Figure 6.3, the via hole diameter is 40% of the electrode pitch. Therefore, only four control pin can be wired in any row. To route a large number of control pins, a multi-layer PCB design is needed, which is prohibitively expensive. Therefore we adopt a “correlated” pin-assignment method, which allows a control pin to be connected to multiple electrodes, thereby reducing the total number of pins.

Next we address the problem using the efficient and easy-to-implement “Connect-5” algorithm presented in Section 3.1. For simplicity, we first look at the application of the algorithm for a two-dimensional array of electrodes without wells. As shown in Section 3.1, five copies of Bagua repetitions are sufficient to cover a biochip array of any size. This is because of the following property of a Bagua repetition: vertices connected to the same (shared) pin appear after exactly five cells in the same row or column of the array, as shown in Figure 3.8.

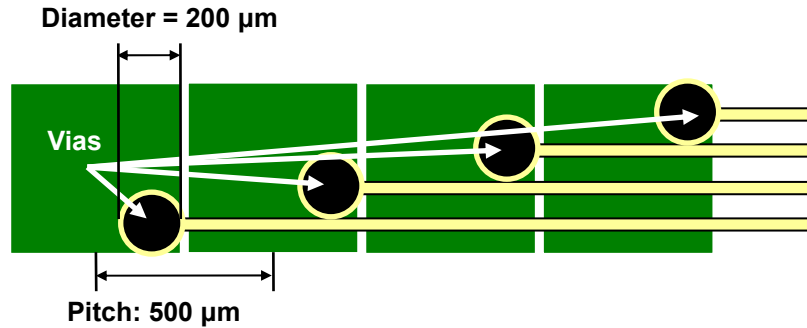


Figure 6.3: Illustration of wire routing limits on a PCB layer.

It has been shown in Figure 3.9 that control pins assigned to the electrodes using this method in a microfluidic array allow free movement of a single droplet, i.e., the “cross constraint” is met.

We modify the above pin-assignment procedure above to make it applicable for our well-plate design. Note that the well-plate design can be viewed as a special case of the two-dimensional array where parts of the array are occupied by wells and segregation walls. Unoccupied electrodes between wells can be used as transportation pathways. Therefore, the pin-assignment for these electrodes does not need to be changed. The overall pin-assignment procedure is as follows.

Start with a two-dimensional electrode array of the same size as the target well-plate design, but with no cells reserved as wells or segregation regions. Apply the Connect-5 algorithm to generate a preliminary pin-assignment result. For example, to generate a pin-assignment result to the multi-well chip in Figure 6.1, a preliminary result is first derived, as shown in Figure 6.4(a).

Next, consider the electrodes that will make up the segregation regions and wells in the multi-well design. Disconnect these electrodes from their control pins, see Figure 6.4(b).

Finally, group the electrodes occupied by each well and connect each group to a single control pin. For independent control of each well, the group control pins must be different not only from each other but also from the pins assigned to the electrodes on the transportation

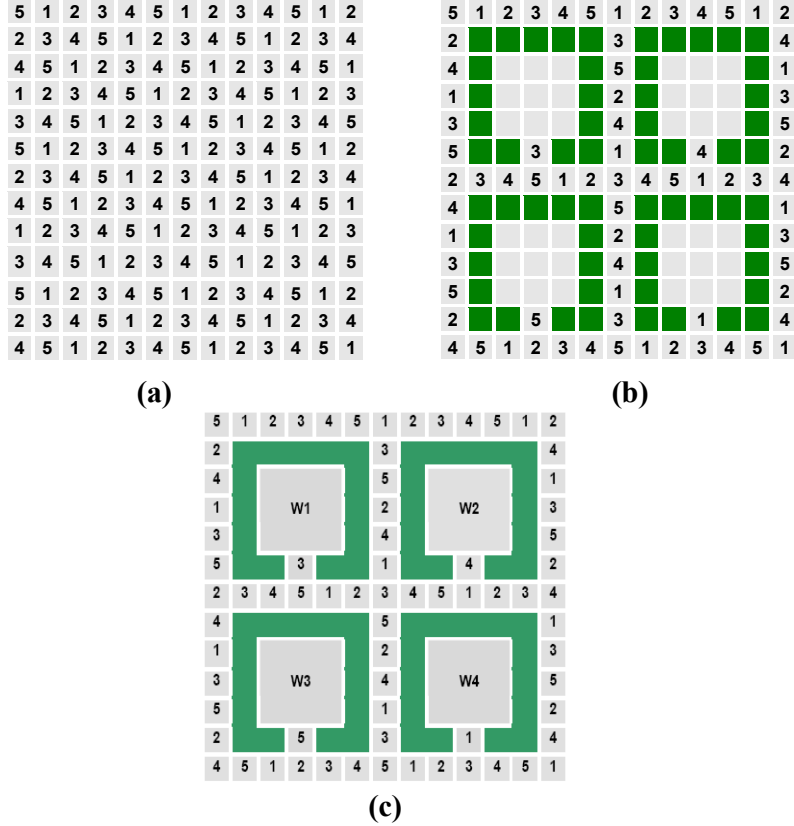


Figure 6.4: Example of pin-assignment example for a 4-well-plate design.

pathway. The modified pin-assignment result is shown in Figure 6.4(c).

Note that in Figure 6.4(c), the same patterns of pin assignment repeat in both column and row directions with a period of 6. Based on this observation, we can adjust the size of the unit well to obtain a more regular pin-assignment result. Here define a *well unit* as a single well and the routing pathways round it. In the design in Figure 6.4(c), the size of the well unit is 7×7 . We first shrink the size of the unit well from 7×7 to 6×6 (since the period of the repetitive pin-assignment patterns is 6) electrodes, as shown in Figure 6.5. Next we apply the Connect-5 algorithm to get a pin assignment for the 96-well chip with the adjusted unit well size, see Figure 6.6.

For a 96 well plate design with well unit size of 5×5 , there are a total of 1284 electrodes in the chip, including electrode in wells, transportation pathways and reservoirs. Therefore, a total

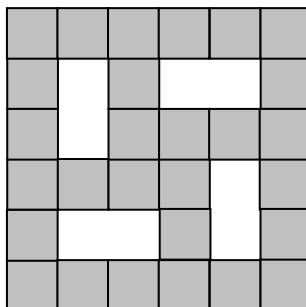


Figure 6.5: Illustration of a 6×6 electrode well unit.

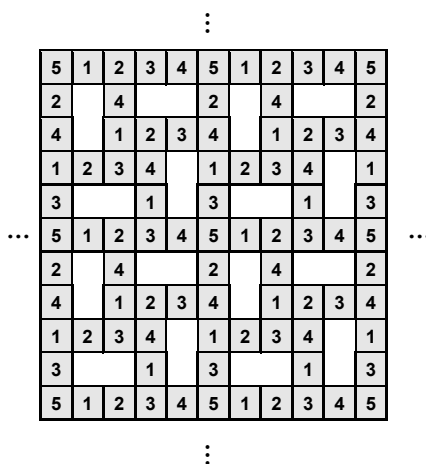


Figure 6.6: Pin assignment using 5 pins for the 96-well chip (unit well size = 6×6 electrodes).

of 1284 control pins are needed for direct addressing. In contrast, the design in Figure 6.6 only needs 5 pins to control all the electrodes on the transportation pathways, thereby significantly reducing the total number of control pins to 181.

The pin-constrained design using Connect-5 method not only significantly reduces the number of control pins but it also provides an easy wiring solution. According to Section 3.1, electrodes sharing the same pin in the pin-assignment result from Connect-5 algorithm are diagonally aligned. Therefore they can be easily wired diagonally, as shown in Figure 6.7. Moreover, the diagonal wiring allows the diameter to be almost the same as the electrode pitch size, as illustrated in Figure 6.8. This efficient wiring plan allows the 181 pins to be wired on a 2-layer PCB. Recall that the direct-addressing method needs 1284 control pins, which requires

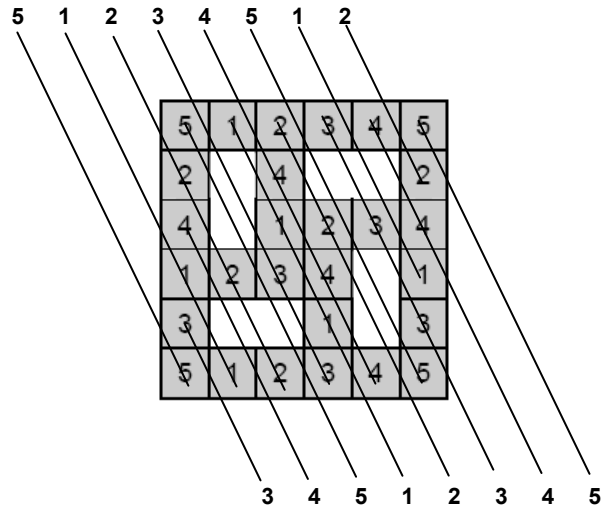


Figure 6.7: Wiring of a well unit.

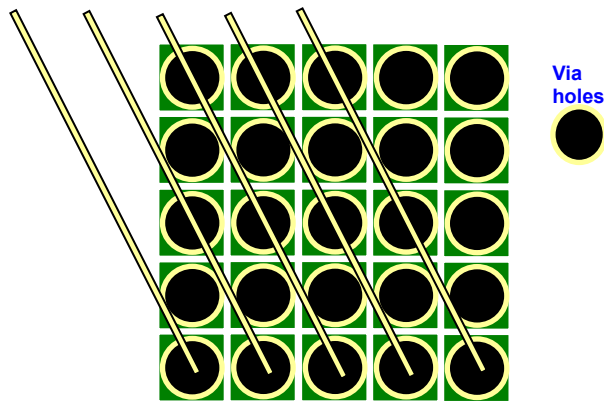


Figure 6.8: Wiring of a well unit with large vias.

a 4-layer PCB and thereby increases the fabrication cost by a factor of 1.6~2 [72]. Moreover, the 181 pins can be easily incorporated using standardized 3 mil feature size technology. In contrast, to fit the 1284 pins in the direct-addressing-based design, 2 mil technology, which usually cost 3-5x times more than 3 mil technology, has to be used. Therefore, the pin-constrained design achieves a reduction of fabrication cost by a factor of 5-10x. The reduction is more significant when the wiring-plan design cost is considered.

In Figure 6.6, every well unit has the same pattern of pin-assignment. This is because the dimension of the unit well is the same as the period of pin-assignment patterns form Connect-5

algorithm. This regular pin-assignment result facilitates the use of an efficient well-loading algorithm, discussed in Section 6.1.2.

6.1.2 Shuttle-Passenger-Like Well-Loading Algorithm

In this section, we focus on the problem of loading the wells with sample and reagent droplets on the pin-constrained chip. The goal is to efficiently route the sample and reagent droplets to their destination wells. Note that in the 96-well chip design in Figure 6.6, every 6×6 well unit has the same pattern of pin-assignment. Therefore, any sequence of manipulations in a single well unit will cause the same manipulations in all the other well units. Although this “synchronizing” property leads to reduced freedom of droplet manipulations, it allows the concurrent manipulation of multiple droplets. Based on this observation, we propose a parallel shuttle-passenger-like routing method for high-throughput well loading.

We illustrate the well-loading algorithm using an example. Figure 6.9 shows a pin-constrained chip which consists of four 6×6 well units. A dispensing reservoir is located at the top right corner on the chip. Three droplets D_1 , D_2 , and D_3 are to be dispensed and routed to three destination wells. If the droplets are placed on the start points as indicated in Figure 6.9, the routing can be carried out simultaneously by applying the control-pin actuation sequence $5 \rightarrow 2 \rightarrow 4 \rightarrow 1 \rightarrow 3 \rightarrow 5 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 1$. The actuation sequence will route all the droplets (if any) at the upper left corner of the well units to the well within the same unit, just as synchronized shuttles that carry passengers from fixed start points to fixed destinations. The shuttles run regularly irrespective of whether there is any passenger. To go to a specific destination, a passenger needs to get to the correct starting point and wait for the shuttle (pin actuation sequence) for pick-up and routing to the destination (well).

Routing of droplets to the starting point can also be carried out using the shuttle-passenger-like method. As in the example in Figure 6.9, the routing step can be carried out using the shuttle (pin activation sequence) as shown in Figure 6.10.

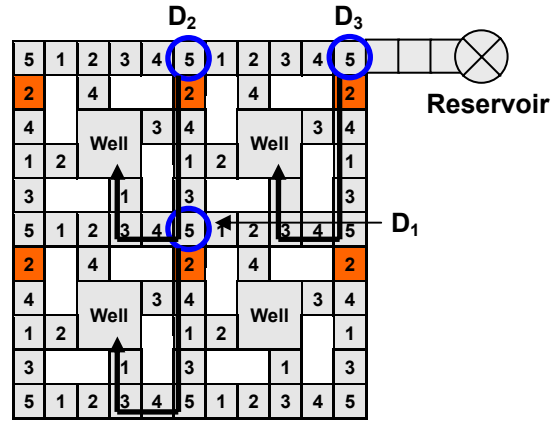


Figure 6.9: Loading of three droplets using shuttle-passenger-like method.

	D ₁ dispensed					D ₂ dispensed			D ₃ dispensed		
Step	1	2	3	4	5	6	7	8	9	10	11
Activated pin	5	2	4	1	3	5	4	3	2	1	5

Figure 6.10: Activation sequence and dispensing time instances for routing droplets to corresponding starting point in Figure 6.9.

Therefore, the proposed well-loading method contains two steps. In the first step, droplets to be routed are transported to the corresponding start points in their destination well units. This step is carried out as follows:

- Calculate the electrode activation sequence to route the droplet to the farthest starting point away from the source reservoir.
- Select a subsequence from the sequence from a) for each droplet that can be route d to its starting point.
- Applying the electrode-activation sequence from a), and dispense each droplet at a specific time corresponding to the start of its subsequence.

Next, a second pin-actuation sequence is applied to route droplets to their target wells. The overall routing steps take little time because all the wells can be filled using only two pin-actuation sequences.

6.1.3 Defect Tolerance

The design proposed in Section 6.1.1 and Section 6.1.2 may suffer from fabrication defects or operational faults. In this section, we propose a “cross loading” based method to achieve defect tolerance for the proposed chip design. We rely on the use of testing and diagnosis methods in Chapter 4 to locate defect sites.

We first classify defects into three categories based on their locations on the chip. Note that in the well-loading algorithm proposed in Section 6.1.2, wells are loaded from one side, i.e., right side or left side. Therefore, not all the electrodes are used. If a defect occurs in these unused electrodes, then it will not affect droplet manipulations on the chip. We refer to this type of defects as *benign defects*. In the design proposed in Section 6.1.1, benign defects include all the defects in the unused entrance electrodes for the well and all the electrodes between the bottom entrance electrodes and the left/right routing pathways if all the wells are loaded from the right/left side. For these benign defects, no defect tolerance is needed.

The second category of defects occurs on the electrodes used by the well-loading algorithm on the electrode rows but not on the routing pathways. These defects are referred to as *loading pathway defects*, as shown in Figure 6.11. These defects can be bypassed by simply changing the side from which the well is loaded.

The third category includes all the defects on the routing pathways. Therefore, we refer to them as *routing pathway defects*. Unlike loading pathway defects, these defects affect the loading operations for more than one well unit. They cannot be bypassed by simply changing the side from which the well is loaded. Instead, we use a “cross loading” method for defect tolerance. Two iterations of well-loading operations are carried out, one in the column direction and one in the row direction. If the defects occur on the routing pathways in the well-loading operation in the column direction, the loading of all the wells within the same column with the defects will be skipped. The skipped wells will then be loaded in the well-loading operation in the row direction and vice versa. An example is shown in

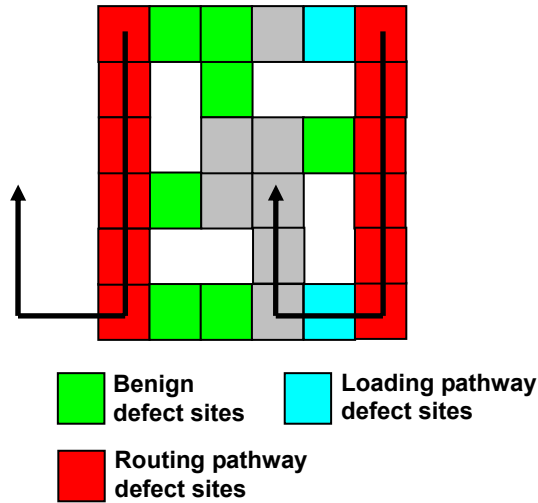


Figure 6.11: Three categories of defects.

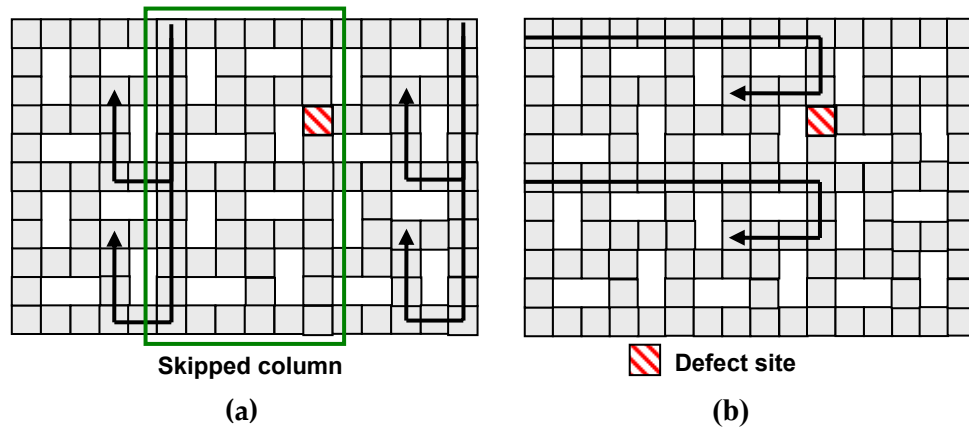


Figure 6.12: Illustration of “cross loading” method (a) loading in column direction. (b) loading in row direction.

Figure 6.12. There is a defect in the routing pathway in the first well unit in the second column. In the column-loading step, the entire second row is skipped. The skipped wells are then loaded in the second iteration of loading in the row direction.

6.1.4 Evaluation of Well-Loading Algorithm and Defect Tolerance

In the section, we evaluate the proposed pin-constraint design and the shuttle-passenger-like well-loading algorithm.

Loading Time

We first calculate the time needed for loading the wells on a pin-constrained chip and a chip with independent pins (direct-access). In a direct-access chip, the time required to load all the wells is determined by the time taken by a droplet to traverse the critical path, i.e., from the dispensing reservoir to the farthest well, as shown in Figure 6.13. For an $N \times N$ array, the routing time for the critical path is $2N - 3$ clock cycles. The proposed pin-constrained chip has the same critical path. Using the well loading algorithm from Section 6.1.2, a droplet can be routed along the critical path one electrode per clock cycle with no stalled cycles. Therefore, the routing time is also $2N - 3$ clock cycles. Thus we conclude that the pin-constrained design provides the same routing efficiency as the direct-access design, while it achieves a significant reduction in the number of control pins.

Defect Tolerance

Next we examine the defect tolerance of the proposed pin-constrained design by injecting random defects. A design is deemed to be robust if the injected defect can be bypassed using the defect-tolerance methods proposed in Section 6.1.3. Some defects may block all the routing pathways to one or more wells, and these wells cannot be loaded. In this case, a failure occurs on the chip.

Next we define a parameter referred to as “failure rate”. Let N_t be the total number of biochips in a representative sample, and let N_f be the number of defective chips that suffers from a failure. Then the *failure rate* f is defined by the equation $f = N_f / N_t$.

We run the simulations with different defect occurrence probabilities for the pin-constrained chip and record the failure rates. As a baseline, we also carry out defect injection for a direct-access chip. Results are shown in Figure 6.14. Note that if we do not set any upper limit on the well-loading time, any defect that can be bypassed in the direct-access chip can also be bypassed in the pin-constrained chip. This is because we can manipulate

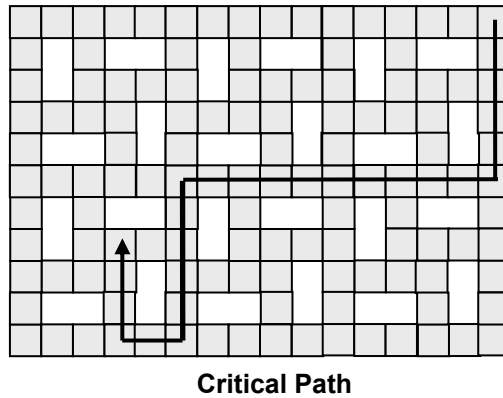


Figure 6.13: Critical path for the multi-well chip (for both the direct-access and pin-constrained chips).

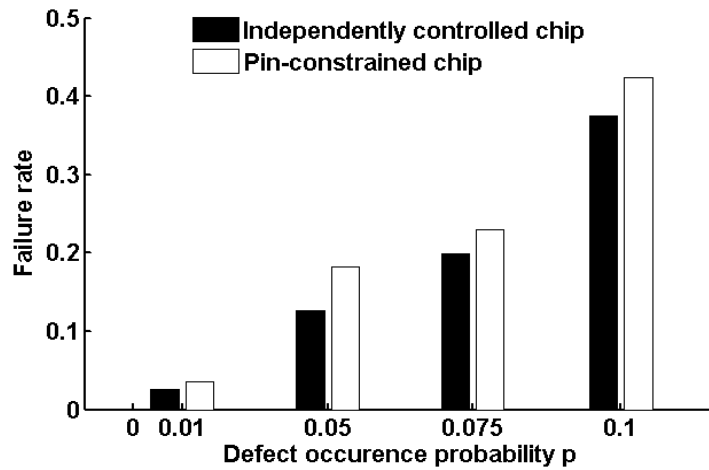


Figure 6.14: Evaluation of failure rates for pin-constrained chip and independently controlled chip.

only one droplet to load only one well in any iteration of shuttle-passenger-like routing, which allows the same degree of freedom as in the direct-access chip. However, this scheme results in a significant increase in the well-loading time. Therefore, in our evaluation, we use a restricted definition of failure for the pin-constrained design; it refers to the case that the injected defects cannot be bypassed using the “cross loading” method.

Figure 6.14 shows that, as expected, the introduction of pin constraints leads to a slightly higher failure rate compared to the direct-access chip. However, this increase is acceptable in

practice due to the significant reduction in the number of control pins for the proposed design.

6.2 Automated Solution Preparation

A critical step in protein crystallization is to generate various mixed crystallizing solutions with desired concentration levels using fixed volume of stock solutions. In the laboratories, these solutions are usually prepared manually, which is very time-consuming, imprecise, and it requires large volumes of liquid. The digital microfluidic biochip design proposed in the previous section provides a hardware platform that allows automated solution preparation based on on-chip droplet transportation and mixing. In this section, we propose an efficient solution-preparation algorithm to generate a preparation plan that lists the intermediate mixing steps needed to generate the target solutions with the required concentrations. It determines the type, concentration, and the number of dispensed droplets of stock solutions. The derived preparation plan is then programmed into a microcontroller that controls the biochip platform to carry out the solution-preparation process.

6.2.1 Efficient Solution-Preparation Planning Algorithm

For a given bioassay, we refer to the set of solutions to be prepared as *target solutions*. In this section, we present an efficient algorithm for preparation of the target solutions.

Concentration Manipulation using Mixing and Dispensing

Suppose that we have to prepare a target solution of reagent A with concentration of 0.2 M. Assume that we only have a stock solution of A with concentration 0.4 M. Therefore, we have to dispense droplets from the stock-solution reservoir and dilute them appropriately, by mixing with equal volumes of a diluent in a reservoir. Note that for better mixing, the reservoir must be filled to its capacity. The dilution of one droplet must be followed by

dispensing of a droplet of 0.2 M concentration of A from the large mixed droplet in that reservoir. By varying the number of droplets routed from the stock solution reservoir to the mixing reservoir, stock solution droplets can be diluted to different concentrations.

However, since digital microfluidic biochip can only handle discrete droplets. The number of droplets routed from the solution reservoirs can only be integers. As a result, the stock solution droplets cannot be diluted to an arbitrary concentration using one iteration of the mixing-and-dispensing operation. Instead, only a set of discrete concentrations are feasible.

For the above example, assuming the dilution is performed in a reservoir whose capacity is 4 times the volume of a unit droplet. We can only dispense 1, 2, or 3 droplets from the stock solution whose concentration is 0.4 M. The concentrations of the dispensed droplet after the dilution can only be 0.1 M, 0.2 M or 0.3 M. Here we define the difference between the outcome concentrations caused by dispensing one more (or less) droplet into the mixing reservoir as *modulation resolution*.

By definition, the modulation resolution can be determined using the equation:

$$\begin{aligned} & \textit{Modulation resolution} \\ & = \frac{\textit{concentration of stock solution} \times \textit{volume of a unit droplet}}{\textit{capacity of mixing reservoir}} \end{aligned} \quad (6.1)$$

In one iteration of mixing-and-dispensing, only concentrations of the dispensed droplet from the mixed large droplet can only be multiples of the modulation resolution. To obtain other concentrations, extra dilution steps are needed to obtain intermediate stock solutions with reduced concentration, which in turn yields finer module resolution according to Equation (6.1).

Solution-Preparation Algorithm

In this subsection, we focus on the problem of generating target solutions with required sample concentrations using the basic dispense-mix-dispense operation as described above.

We refer to this process as “solution-preparation planning”.

Given a set of target solutions, the solution-preparation planning algorithm determines,

1. The types of stock solutions that are needed.
2. The concentration of each stock solution.
3. The number of dispensed droplets of the stock solutions.
4. How these droplets must be mixed so that we can derive the target solutions using smallest number of droplet manipulation steps (routing, mixing and dilutions).

Next we use an example to illustrate the algorithm. In this example, we plan to generate a set of 24 target solutions from a protein crystallization assay, as shown in Table 6.1. To start, the algorithm first determines the types of stock solutions needed. Here we use one stock solution for each type of reagent included in the set of target solutions. Therefore, the number of stock solutions is the same as the total types of reagents included in the target solutions. For the example in Table 6.1, 17 different types of reagents are included in the target solutions. Thus 17 types of stock solutions are needed, as listed in Table 6.2. These stock solutions are stored in on-chip reservoirs.

Next the algorithm determines the concentration of each type of stock solution. For each type of stock solution, the algorithm identifies all the target solutions that contain the corresponding reagent in the stock solution.

For the example shown in Table 6.1 and Table 6.2, stock solution S_5 contains the reagent *polyethylene glycol 4000*. There are six target solutions that contain this reagent, i.e., *MembFac_02* and *MembFac_03*, *MembFac_05*, *MembFac_13*, *MembFac_17*, *MembFac_23*, as listed in Table 6.3. The concentrations of the reagent *polyethylene glycol 4000* in these target solutions are 12 %w/v, 10 %w/v, 12 %w/v, 12%w/v, 12 %w/v, and 12 %w/v, respectively.

Table 6.1: Target solution list for protein crystallization

Condition ID	Reagent_ID	*	Condition ID	Reagent_ID	*
MembFac_01	sodium chloride	0.1 M	MembFac_13	polyethylene glycol 4000	12 %w/v
MembFac_01	sodium acetate trihydrate	0.1 M	MembFac_13	lithium sulfate monohydrate	0.1 M
MembFac_01	2-methyl-2,4-pentanediol (MPD)	12 %v/v	MembFac_13	tri-sodium citrate dihydrate	0.1 M
MembFac_02	zinc acetate dihydrate	0.1 M	MembFac_14	iso-propanol (IPA)	10 %v/v
MembFac_02	sodium acetate trihydrate	0.1 M	MembFac_14	tri-sodium citrate dihydrate	0.1 M
MembFac_02	polyethylene glycol 4000	12 %w/v	MembFac_14	tri-sodium citrate dihydrate	0.1 M
MembFac_03	ammonium sulfate	0.2 M	MembFac_15	2-methyl-2,4-pentanediol (MPD)	12 %v/v
MembFac_03	polyethylene glycol 4000	10 %w/v	MembFac_15	sodium chloride	0.1 M
MembFac_03	sodium acetate trihydrate	0.1 M	MembFac_15	tri-sodium citrate dihydrate	0.1 M
MembFac_04	sodium chloride	0.1 M	MembFac_16	magnesium sulfate heptahydrate	1 M
MembFac_04	iso-propanol (IPA)	12 %v/v	MembFac_16	tri-sodium citrate dihydrate	0.1 M
MembFac_04	sodium acetate trihydrate	0.1 M	MembFac_17	tri-sodium citrate dihydrate	0.1 M
MembFac_05	sodium acetate trihydrate	0.1 M	MembFac_17	sodium chloride	0.1 M
MembFac_05	polyethylene glycol 4000	12 %w/v	MembFac_17	polyethylene glycol 4000	12 %w/v
MembFac_06	ammonium sulfate	1 M	MembFac_18	lithium sulfate monohydrate	0.1 M
MembFac_06	sodium acetate trihydrate	0.1 M	MembFac_18	tri-sodium citrate dihydrate	0.1 M
MembFac_07	magnesium sulfate heptahydrate	1 M	MembFac_18	polyethylene glycol 6000	12 %w/v
MembFac_07	sodium acetate trihydrate	0.1 M	MembFac_19	magnesium chloride hexahydrate	0.1 M
MembFac_08	sodium acetate trihydrate	0.1 M	MembFac_19	2-methyl-2,4-pentanediol (MPD)	4 %v/v
MembFac_08	magnesium chloride hexahydrate	0.1 M	MembFac_19	tri-sodium citrate dihydrate	0.1 M
MembFac_08	polyethylene glycol 400	18 %v/v	MembFac_20	sodium chloride	0.1 M
MembFac_09	ammonium dihydrogen phosphate	1 M	MembFac_20	tri-sodium citrate dihydrate	0.1 M
MembFac_09	sodium acetate trihydrate	0.1 M	MembFac_21	polyethylene glycol 400	4 %v/v
MembFac_09	lithium sulfate monohydrate	0.1 M	MembFac_21	tri-sodium citrate dihydrate	0.1 M
MembFac_10	polyethylene glycol 6000	12 %w/v	MembFac_21	lithium sulfate monohydrate	0.1 M
MembFac_10	sodium chloride	0.1 M	MembFac_22	ADA	0.1 M
MembFac_10	sodium acetate trihydrate	0.1 M	MembFac_22	ammonium sulfate	1 M
MembFac_11	sodium acetate trihydrate	0.1 M	MembFac_23	ADA	0.1 M
MembFac_11	magnesium chloride hexahydrate	0.1 M	MembFac_23	polyethylene glycol 4000	12 %w/v
MembFac_11	polyethylene glycol 6000	12 %w/v	MembFac_23	lithium sulfate monohydrate	0.1 M
MembFac_12	sodium chloride	0.1 M	MembFac_23	iso-propanol (IPA)	2 %v/v
MembFac_12	polyethylene glycol 400	18 %v/v	MembFac_24	di-ammonium hydrogen phosphate	1 M

*Reagent concentration

Recall that in the previous subsection, we have shown that these concentrations must be multiples of the modulation resolution, i.e., the module resolution must be a common factor of these concentrations.

Table 6.2: Stock solutions needed to prepare the target solutions in Table 6.1.

S1	sodium chloride	1 M	S10	polyethylene glycol 400	36 %v/v
S2	sodium acetate trihydrate	1 M	S11	ammonium dihydrogen phosphate	10 M
S3	2-methyl-2,4-pentanediol (MPD)	120 %v/v	S12	lithium sulfate monohydrate	1 M
S4	zinc acetate dihydrate	1 M	S13	polyethylene glycol 6000	120 %w/v
S5	polyethylene glycol 4000	20 %w/v	S14	tri-sodium citrate dihydrate	1 M
S6	ammonium sulfate	2 M	S15	ADA	1 M
S7	iso-propanol (IPA)	100 %v/v	S16	Ammonium sulfate	10 M
S8	magnesium sulfate heptahydrate	10 M	S17	di-ammonium hydrogen phosphate	10 M
S9	magnesium chloride hexahydrate	1 M			

In our algorithm, we pick the great common divisor (GCD) of these concentrations as the modulation resolution. This is because smaller modulation resolution indicates more droplets need to be routed into the mixing reservoir to obtain the target concentration. As a result, more droplet manipulation steps are needed.

In the above example, the modulation resolution = $\text{GCD}(12, 10, 12, 12, 12, 12) = 2\%w/v$. Assuming the volume of a unit droplet is 20 nl and the capacity of the mixing reservoir is 200 nl, we can use the transformation of Equation (6.1) to calculate the concentration of stock solution containing reagent *polyethylene glycol 4000* = $2\%w/v \times 200\text{nl} / 20\text{nl} = 20\%w/v$.

$$\begin{aligned} & \text{Concentration of stock solution} \\ &= \frac{\text{Modulation resolution} \times \text{capacity of mixing reservoir}}{\text{volume of a unit droplet}} \end{aligned} \quad (6.2)$$

Next we calculate the number of droplets that are needed to be routed from the stock solution to the mixing reservoir in generating a target solution. It can be obtained using the following equation:

$$\begin{aligned} & \# \text{ of droplets routed from stock solution} \\ &= \frac{\text{Concentration of the reagent in the target solution}}{\text{Modulation resolution}} \end{aligned} \quad (6.3)$$

For the above example, the numbers of droplets from stock solution S_5 for target solutions are listed in Table 6.3. We can obtain the concentration of all the other stock solutions and the

Table 6.3. Target solutions containing reagent polyethylene glycol 4000.

Condition ID	Reagent_ID	Reagent concentration		*
MembFac_02	polyethylene glycol 4000	12	%w/v	6
MembFac_03	polyethylene glycol 4000	10	%w/v	5
MembFac_05	polyethylene glycol 4000	12	%w/v	6
MembFac_13	polyethylene glycol 4000	12	%w/v	5
MembFac_17	polyethylene glycol 4000	12	%w/v	5
MembFac_23	polyethylene glycol 4000	12	%w/v	6

of droplets routed from the stock solution to the mixing reservoir

Table 6.4. Preparation plan for target solution MembFac_02.

Reagent_ID	Stock solution #	Concentration	# of droplets
zinc acetate dihydrate	S4	1 M	1
sodium acetate trihydrate	S2	1 M	1
polyethylene glycol 4000	S3	20 %w/v	6
diluent	—	—	2
note: total number of droplets = 1+1+6+2 = 10 unit droplets = capacity of mixing reservoir			

number of droplets to be routed from each stock solution in a similar manner. Now we can generate the preparation plan for a target solution, an example is shown in Table 6.4.

In Table 6.4, the total number of droplets from different types of stock solutions is $1+1+6 = 8 < 10$ unit droplets (the capacity of mixing reservoir). Two droplets of diluent are routed to fill up the mixing reservoir. For some other target solutions, the total number of droplets from different stock solutions may exceed the capacity of the mixing reservoir. In this case, we identify the stock solution that dispenses the largest number of droplets and double its concentration. By this means, fewer droplets are needed to generate the target solution. As a tradeoff, the modulation resolution is lowered. To obtain finer resolution, extra dilution iterations are needed.

Next we can list the steps involved in the solution-preparation planning algorithm.

1. Given a set of target solutions T_s ($T_{s_1}, T_{s_2}, T_{s_3}, \dots, T_{s_m}$), identify the types of reagents contained R ($R_1, R_2, R_3, \dots, R_n$).
2. Determine the set of stock solutions S_s ($S_{s_1}, S_{s_2}, S_{s_3}, \dots, S_{s_n}$) following the mapping:

$$R_i \leftrightarrow Ss_i$$

3. For each type of stock solution Ss_i , identify the set of target solutions TsR_i that contain the corresponding reagent R_i .
4. Determine the modulation resolution for stock solution Ss_i using the equation $modulation\ resolution = GCD(TsR_i)$, when GCD refers to the greatest common divisor. The Euclidean algorithm is used to compute the GCD [73].
5. Determine the concentration for each solution using Equation (6.2).
6. Calculate the number of droplets dispensed from each stock solution reservoir for each type of target solution using Equation (6.3).
7. Check if the total number of droplets dispensed from different stock solution reservoirs exceeds the capacity of the mixing reservoir. If yes, go to step 8, otherwise, the algorithm terminates.
8. Identify the stock solution that dispenses the most droplets into the mixing reservoir in preparing the target solution. Double the concentration of that stock solution. Then go to step 6.

Assuming the total number of target solutions is m and n types of reagents are contained. Step 1 scans the target solution set and record different reagents. It takes $O(n+m)$ time. Step 2 carries out one-to-one mapping and takes $O(n+m)$ time. For step 3, in the worst case, each target solution contains all types of reagents. Therefore, this step takes $O(nm)$ time. According to the Euclidean algorithm, Step 4 takes $O(n \log_{10} C_{max})$ time where C_{max} is the largest value of the concentrations [73]. For most bioassays, the concentrations can be represented using 4 digits, i.e., $k < 10000$. Therefore, Step 4 takes $O(4n) = O(n)$ time. Step 5 takes $O(n)$ time. For Step 6, in the worst case, each target solution contains all types of reagents. Thus this step takes $O(nm)$ time. Step 7 takes $O(n+m)$ time. Step 8 looks for the stock solution that dispenses the most droplets into the mixing reservoir in preparing the target solution, which takes $O(n)$ time.

For the entire algorithm, in the best case, Step 8 is never reached. The algorithm takes $O(n+m+n+m+nm+n+n+n+m+nm+n) = O(nm)$ time. In the worst case, for every target solution, the total number of droplets dispensed from different stock solution reservoirs exceeds the capacity of the mixing reservoir. Steps 6-8 are executed for each target solution, i.e., m times. Therefore, the entire solution-preparation planning algorithm takes $O(nm^2)$ time.

We next address the complexity of the fluidic operation; in the best case, the preparation a single target solution requires only one iteration of mixing-and-dispensing operation. No extra dilution is needed. The entire preparation plan takes m mixing-and-dispensing operations. In the worst case, the preparation of each target solution requires an extra dilution step. The preparation plan requires $2m$ mixing-and-dispensing operations.

6.2.2 Experimental Results and Comparison

Next we use the planning algorithm to carry out solution-preparation for protein crystallization.

For simplicity, we extract 24 target solutions from the thousands of solutions for the experiment as listed in Table 6.1. From Table 6.1, we can see that 17 types of reagents are used, as listed in Table 6.2. After applying the solution-preparation planning algorithm, 17 source solutions with certain concentrations are chosen corresponding to the 17 types of reagents and stored in on-chip reservoirs.

Next we prepare these target solutions. First, manual operation is used. A pipette that can handle a minimum volume of 20 μ l is used. Preparing the target solution consumes 22 ml of reagent stock solutions and takes 1.5 hours. In contrast, proposed chip design and the solution-preparation planning algorithm take only 18 minutes and 12 μ l of reagent solutions.

For protein crystallization, reagent concentration is very important. Therefore, we need to guarantee a high level of accuracy of concentration while preparing the target solutions. For a

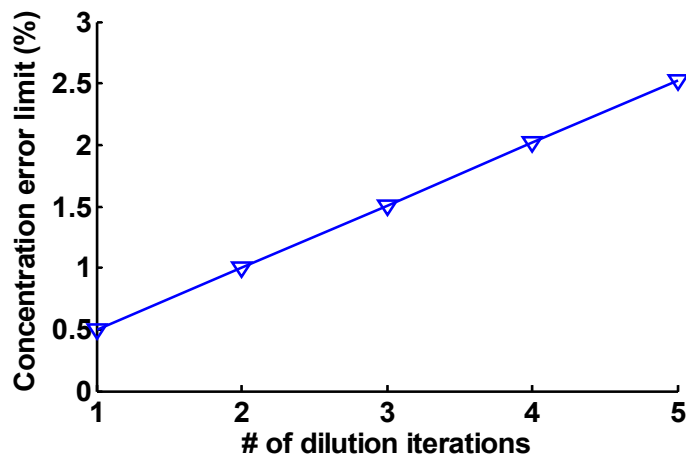


Figure 6.15: Concentration error limit vs number of mixing-dispensing iterations

digital microfluidic biochip, the key to generating solutions with precise concentration is to maintain constant volume of the dispensed droplets.

Experiments have shown that our chip design achieves a high level of consistency in the volume of dispensed droplets (variation $< 0.5\%$), which indicates high accuracy in the concentration of the prepared target solutions. Note that the accuracy will degrade when multiple iterations of dilution are carried out. However, results also show an error limit of less than 2.5% even when five iterations of mixing-dispensing operations are used in preparing the target solutions; see Figure 6.15.

6.3 Chapter Summary and Conclusions

We have presented a multi-well plate based digital microfluidic biochip design for protein crystallization. The proposed biochip is capable of concurrently setting up 96 conditions, thereby achieves high throughput. We have also applied an efficient algorithm to generate a pin-assignment plan for the proposed design, which enables control of the biochip with only a small number of pins. Compared to a directly addressable biochip, the proposed pin-constrained design achieves a significant reduction in fabrication cost. We have also

described efficient droplet-routing algorithms for defect-tolerant well-loading. An efficient solution-preparation planning algorithm has also been presented to facilitate the generation of crystallizing solutions. Given a set of target solutions, the algorithm determines the type, concentration, and the number of dispensed droplets of the stock solutions. The propose chip design and associated algorithms will pave the way for increased use of digital microfluidic biochips in high-throughput, highly automated, and affordable protein crystallization systems.

Chapter 7

Conclusions and Future Work

7.1 Thesis Contributions

This thesis has presented advanced optimization techniques for automated synthesis, testing, and pin-constrained biochip design. In contrast to previous conceptual methods, the proposed techniques address practical issues that arise in the design, fabrication, and use and maintenance of digital microfluidic biochips. By bridging the gaps between theory and realistic applications, these techniques provide powerful, practical and fully-automated design tools for of digital microfluidics.

Droplet routing and defect-tolerance issues have been considered during chip synthesis, and we have presented a defect-tolerant, routing-aware, PRSA-based synthesis method. Droplet routability, defined as the ease with which droplet pathways can be determined, has been estimated and integrated in the synthesis flow. The proposed method increases the likelihood that feasible droplet pathways can be found for area-constrained biochip layouts. The synthesis tool also implements anticipatory defect-tolerance to guarantee system robustness for the synthesized design.

To reduce fabrication cost, we have developed three techniques for pin-constrained biochip design. The droplet-trace-based array-partitioning method utilizes the concept of droplet trace, which is extracted from the scheduling and droplet-routing results produced by a synthesis tool. An efficient “Connect-5” pin-assignment scheme is also developed and combined with the array-partitioning algorithm to control a large number of electrodes with a small number of control pins. Another pin-constrained method is based on a “cross-referencing” addressing structure that uses “rows” and “columns” to access electrodes in digital microfluidic arrays. A clique-partitioning-based droplet manipulation algorithm has

been developed for the “cross-referencing” biochip which allows simultaneous movement of a large number of droplets. A broadcast-addressing method has been presented. The concept of “don’t-care” status in the control of electrodes has also been introduced for the first time. By combining “compatible” electrodes, the broadcast-addressing method achieves low input bandwidth while providing high throughput. The above three methods provide a comprehensive framework for the automated design of pin-constrained digital microfluidic biochips

We have also developed a comprehensive fault-model library, which consists of not only physical defects and but also malfunctions. Efficient structural test and diagnosis methods have been proposed based on parallel manipulation of multiple test droplets in a scan-like manner. The proposed method can be used in both on-line and off-line scenarios. We have also introduced the concept of functional testing for digital microfluidic biochips. A set of testing techniques have been presented to verify the functionality of on-chip fluidic modules, such as mixers, splitters and dispensing reservoirs.

This thesis has also identified the need of design-for-testability techniques for digital microfluidic biochips. Testability considerations have been addressed in the synthesis flow. Appropriate modifications have also been made in testing methods to increase their effectiveness.

We have applied the proposed design and optimization methods to a real-life protein crystallization assay. The successful design of a low-cost, easily manufacturable, high-throughput, and robust chip for protein crystallization has resulted from the optimization algorithms developed in this thesis. This thesis has therefore led to powerful design tools for application- and-technology-guided chip design and a bridge between synthesis theory and realistic applications.

7.2 Future Work

Despite the progress that we have made, numerous challenges remain to be tackled for real-chip design automation. Section 7.2.1 presents a physical-constrained-guided synthesis tool based on the framework described in Chapter 2. This section describes how design parameters and physical constraints, as derived from the fabrication process, can be incorporated into the synthesis flow. It also proposes a synthesis flow that is guided by physical constraints. Section 7.2.2 describes a research plan to handle errors that occur during bioassay execution. To ensure system dependability, many bioassays must be monitored during execution at several “checkpoints” using sensor-based feedback. When a malfunction is detected or the outcome of an assay step deviates from the expected outcome, certain fluidic operations must be re-executed. A future research for developing a feedback-based synthesis tool that integrates control flow and conditional “if-then-else” operations is outlined.

7.2.1 Synthesis Based on Physical Constraints

The unified synthesis method described in Chapter 2 provides a powerful tool for the automated design of digital microfluidic biochips. It combines geometry-level synthesis with architectural synthesis and generates a comprehensive design with detailed resource binding, operation scheduling and module placement information. However, this synthesis approach is oblivious to constraints imposed by the manufacturing process. Resource-binding, operation-scheduling and module-placement decisions are made without any consideration of physical constraints, such as transportation speed limit, maximum switching frequency, and reservoir capacity. A chip design that disregards these physical constraints can suffer from a severe “mismatch problem”. An experimental example is shown in Figure 7.1, where a dispensing operation has been mapped onto a reservoir on a PCB chip. The synthesis result requires four droplets to be dispensed from the reservoir. However, the capacity of the

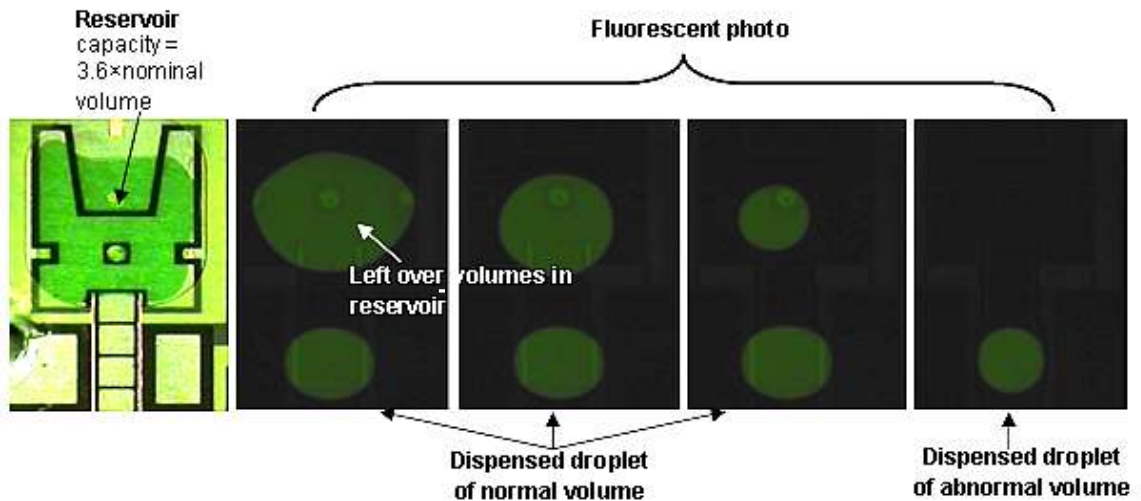


Figure 7.1: An example of mismatch problem – reservoir capacity overflow.

reservoir is only 3.6 times the nominal volume of a standard droplet. By carrying out four iterations of dispensing, we obtained three droplets of normal volume and one droplet of smaller volume as shown in Figure 7.1. This “shrunk” droplet may be too small to overlap with an adjacent electrode and therefore cannot be moved using electrowetting. This type of mismatch problems can be catastrophic for bioassay execution and must be avoided.

Mismatch Problems

Mismatch problems can appear in various forms. Here we discuss two of the most common problems, namely incorrect scheduling and undesirable electrode charging. Incorrect scheduling can result when the desired operation frequency, i.e., the frequency for the control signals for the synthesized design exceeds the transportation speed limit of the chip. In this case, the target synthesized bioassay must be executed at lower frequency. The reduction in clock frequency leads to performance degradation for some fluidic modules. For example, the operation time for a 2×3 mixer increases from 3s to 10s when operation frequency is reduced from 10Hz to 3Hz (the mixing time of 3s for 10Hz frequency is taken from [49]). Some other modules, however, are insensitive to a change in operation frequency. An optical detector has the same detection time at

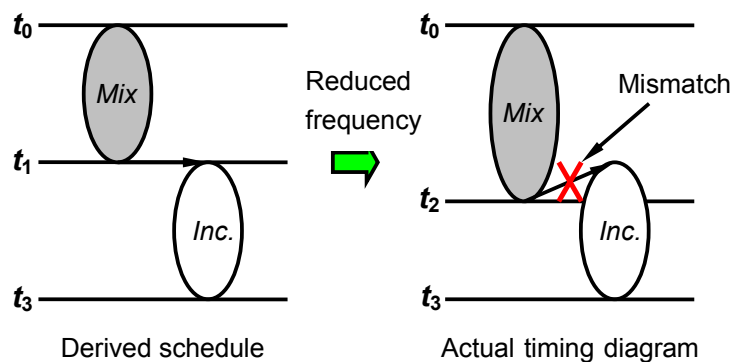


Figure 7.2: An example of scheduling error (*Inc.* refers to incubation operation).

all clock frequencies and a thermal incubator requires a fixed incubation time. These differences in the performance degradation for different types of modules can invalidate the scheduling results derived from the synthesis procedure. The resulting timing mismatch in bioassay execution is referred to as *scheduling error*. Figure 7.2 shows an example of scheduling error for an assay that involves a mixing step followed by an incubation operation. In the schedule derived from the synthesis flow, incubation starts at time t_1 . However, when the operation frequency is reduced, mixing finishes at a later time t_2 , which leads to a timing mismatch between the mixer and the incubator.

A scheduling error can be catastrophic for a bioassay that is mapped to a microfluidic array. For example, for on-chip dilution, a splitting operation is scheduled right after a mixing operation. A scheduling error may cause a splitting of a droplet which is not fully mixed, which can lead to two droplets with different sample concentrations. The solution to this incorrect scheduling problem requires complete resynthesis, which is inefficient and undesirable for the end-user. Therefore, research is needed to avoid it in the synthesis procedure.

Electrode charging is also a common problem in the execution of a synthesized bioassay. In a synthesis result, an electrode can be activated for an arbitrarily long period of time. However, in real chips, long activation duration for an electrode may lead to charge

accumulation on the electrode, and thereby result in electrode breakdown. When that happens, the chip is permanently damaged. Therefore, such problems must be avoided during synthesis.

Synthesis Guided by Physical Constraints

In this subsection, we describe how the above mismatch problems can be avoided by incorporating physical constraints in synthesis. The synthesis method presented in Chapter 2 serves as a starting point for this approach. To avoid scheduling errors caused by a change in the operation frequency, we propose to add chip frequency to the design specification and expand the module library. The expanded module library will consist of several sub-libraries, with each sub-library storing the operation time for all the modules at a certain operation frequency. When an operation frequency of the target chip is specified, the synthesis tool will choose the corresponding sub-library to generate resource binding, scheduling, and the placement plan.

To avoid the problem of electrode charging, we can add a “successive activation” time limit as a design rule for synthesis. *Successive activation* is defined as the number of clock cycles for which the electrode is active. In the PRSA-based synthesis flow in Chapter 2, for each candidate synthesis result, we can calculate the maximum successive activation time for each electrode and add it to the fitness function. Candidate designs with high successive activation time must be discarded during evolution.

7.2.2 Control-Path Design and Synthesis

The synthesis method of Chapter 2 also suffers from the drawback that it assumes no control or feedback mechanism during bioassay execution. The fluidic operations are carried out following the pre-determined schedule without any feedback. Therefore, the only way to ascertain the correctness of such a synthesized biochip is to examine parameters such as the volume of the

product droplet, sample concentration in the product, detector readout, etc. If an error is detected at the end of the assay, the entire bioassay must be repeated. For example, in the protein dilution assay described in Chapter 2, a sample droplet is diluted by buffer droplets using multiple hierarchies of binary mixing/splitting phases. If an error occurs in the dispensing reservoir and leads to a sample droplet of abnormal volume, the concentrations of all the product droplets are affected. As a result, the entire assay (133 operations including droplet dispensing, mixing/splitting, and detection) must be re-executed. Such repetitive executions can potentially lead to wastage of samples and an undue increase in the assay time. Therefore, a monitoring and an appropriate feedback control mechanism must be implemented. During bioassay execution, a monitoring program can determine the status of the assay and the quality of intermediate products at several checkpoints. If a malfunction is detected or the quality of an intermediate product fails to meet predetermined requirements, only a fragment of the bioassay is re-executed. Here we refer to monitoring and control mechanisms as “control path” for the digital microfluidic biochip. We next outline an automated design tool for the synthesis of control paths.

Control-Path Design Based on Error-Propagation

The synthesis of a control path consists of two segments—control-path design and control-path synthesis. Given a bioassay sequencing graph, control-path design determines which operations need to be monitored, and thereby where the checkpoints are needed. For each checkpoint, the control-path design determines which part of the assay must be re-executed in case if an error is detected. However, checkpoint monitoring and re-executions lead to increased completion times. Therefore, careful design is needed to limit the number of checkpoints and the size of the re-execution segment for each checkpoint. We propose an efficient control-path design method based the concept of error-propagation estimates.

In a digital microfluidic biochip platform, each fluidic operation works within a specific error limit, which is defined as worst-case percentage offset of the actual output value from the

nominal value. For example, a dispensing operation with an error limit of 10% implies that the reservoir, in the worst case, can dispense a droplet with a volume of 1.1 or 0.9 times the normal value. In practice, the error limit can be obtained using experiments.

Given a target bioassay protocol, we can collect the error-limit information for every fluidic operation in the protocol. Using error-analysis [74], the error limit of the output of an operation can be derived from the error limit of the input of the operation and the operation's intrinsic error limit. From the start of operations of the protocol, we apply the error-propagation theory and calculate the error-limit for the output of each operation. The value of the error limit is increased as more operations are considered in the protocol. At some point, the derived output error-limit will exceed a predetermined threshold, which is obtained from the precision requirement of the protocol. At this point, a checkpoint must be added. In this way, the error-propagation-based checkpoint-allocation method minimizes the number of checkpoints while maintaining coverage for all the possible failures during assay operation.

After a checkpoint C_1 is determined, a re-execution subroutine needs to be assigned to it. Here we do a "backtrace" operation along the sequencing graph until another checkpoint C_2 is reached. We define the fragment of the bioassay from the upstream checkpoint C_2 to the current checkpoint C_1 as the re-execution subroutine for the C_1 , see Figure 7.3. Note that during bioassay execution, a checkpoint can only be reached when no failure is detected in all its upstream checkpoints. This implies that the error is localized among the operations between the C_1 and C_2 . Therefore, by re-executing the subroutine, i.e., operations O_1 and O_2 , the error can be corrected. Note that there may be multiple backtrace paths from C_1 to C_2 . In this case, it is a challenge to tell which path causes the error. However, since all the operations in these paths are scheduled between C_1 and C_2 , in practice, we can correct the error by re-executing all the operations scheduled between C_1 and C_2 .

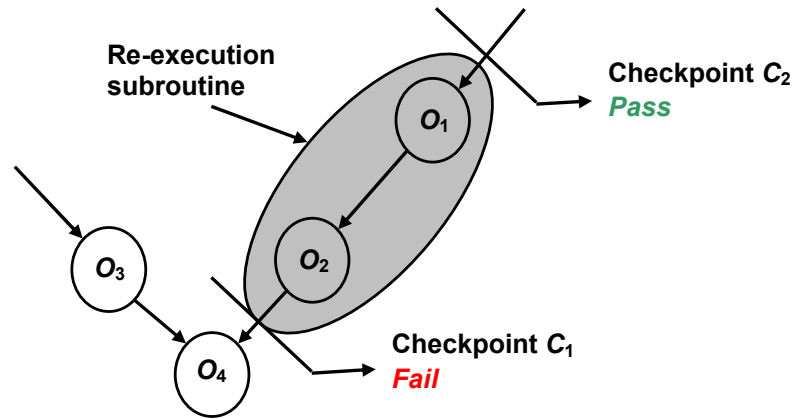


Figure 7.3: Backtrace algorithm and re-execution subroutine.

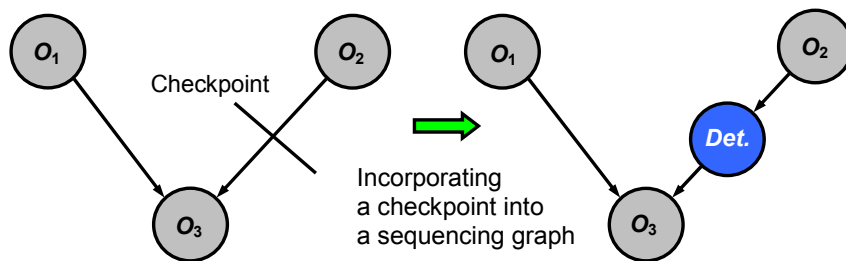


Figure 7.4: The incorporation of a checkpoint to a sequencing graph (*Det.* refers to detection operation).

synthesis. We focus here on the synthesis of checkpoints. The goal is to incorporate checkpoints into the synthesis result for a bioassay without a control path.

Each checkpoint can be mapped to a detection operation and inserted into the sequencing graph at the same location. For instance, a checkpoint located between operations O_2 and O_3 in Figure 7.4 is mapped to a detection operation at the same position. By applying the synthesis method of Chapter 2 to this modified sequencing graph, the control path can be easily incorporated as part of the bioassay protocol. Checkpoints are then mapped to on-chip sensors. The corresponding intermediate product droplets will be routed to the sensors for detection at scheduled time point. The sensor output is sent to the microcontroller that coordinates bioassay execution. The bioassay synthesis results are mapped to a software program and stored in microcontroller memory. Each re-execution subroutine corresponds to

a fragment of the program that can be identified by the starting addresses of these subprograms in memory. If an error is detected at a certain checkpoint, the microcontroller intercepts the program for the bioassay and points the program counter to the starting address of the corresponding subprogram. In this way, control-flow feedback is implemented based on an interrupt-handling mechanism implemented in software.

References

- [1] T. H. Schulte, R. L. Bardell and B. H. Weigl, "Microfluidic technologies in clinical diagnostics", *Clinica Chimica Acta*, vol. 321, pp. 1-10, 2002.
- [2] V. Srinivasan, V. K. Pamula, M. G. Pollack and R. B. Fair, "Clinical diagnostics on human whole blood, plasma, serum, urine, saliva, sweat, and tears on a digital microfluidic platform", *Proceeding of Miniaturized Systems for Chemistry and Life Sciences (μ TAS)*, pp.1287-1290, 2003.
- [3] A. Guiseppi-Elie, S. Brahim, G. Slaughter, and K. R. Ward, "Design of a subcutaneous implantable biochip for monitoring of glucose and lactate", *IEEE Sensors Journal*, vol. 5, no. 3, pp. 345-355, 2005.
- [4] E. Verpoorte and N. F. De Rooij, "Microfluidics meets MEMS", *Proceeding of IEEE*, vol. 91, pp. 930-953, 2003.
- [5] R. B. M. Schasfoort, S. Schlautmann, J. Hendrikse, A. van den Berg, "Field-effect flow control for microfabricated fluidic networks", *Science*, vol. 286, pp. 942-945, 1999.
- [6] Fluidigm Corporation, <http://www.fluidigm.com>.
- [7] Caliper Life Science, <http://www.caliperls.com>.
- [8] Tecan Systems Inc, <http://www.tecan.com>
- [9] R. B. Fair, V. Srinivasan, H. Ren, P. Paik, V. K. Pamula and M. G. Pollack, "Electrowetting-based on-chip sample processing for integrated microfluidics," *Proceeding of IEEE International Electron Devices Meeting (IEDM)*, pp. 32.5.1-32.5.4, 2003.
- [10] M. G. Pollack, R. B. Fair and A. D. Shenderov, "Electrowetting-based actuation of liquid droplets for microfluidic applications", *Applied Physics Letters*, vol. 77 no. 11, 2000.
- [11] S. K. Cho, H. Moon, and C-J. Kim, "Creating, transporting, cutting, and merging liquid droplets by electrowetting-based actuation for digital microfluidic circuits", *Journal of Microelectromechanical Systems*, vol. 12, no. 1, pp. 70-80, 2003.
- [12] M. Abdelgawad and A. R. Wheeler, "Rapid Prototyping in Copper Substrates for Digital Microfluidics". *Advanced Material*, vol. 19, pp.133-137, 2007.
- [13] R. B. Fair, A. Khlystov, T. D. Taylor, V. Ivanov, R. D. Evans, P. B. Griffin, V. Srinivasan, V. K. Pamula, M. G. Pollack and J. Zhou, "Chemical and biological applications of digital-microfluidic devices", *IEEE Design & Test of Computers*, vol. 24, pp. 10-24, 2007.

- [14] F. Su and K. Chakrabarty, "High-level synthesis of digital microfluidic biochips", *ACM Journal on Emerging Technologies in Computing Systems*, vol. 3, Article 16, January 2008
- [15] F. Su and K. Chakrabarty, "Unified high-level synthesis and module placement for defect-tolerant microfluidic biochips", *Proceeding of IEEE/ACM Design Automation Conference*, pp. 825-830, 2005.
- [16] P. -H. Yuh, C. -L. Yang and C. -W. Chang, "Placement of defect-tolerant digital microfluidic biochips using the T-tree formulation", *ACM Journal on Emerging Technologies in Computing Systems*, vol. 3, issue 3, 2007.
- [17] A. J. Ricketts, K. Irick, N. Vijaykrishnan and M. J. Irwin, "Priority scheduling in digital microfluidics-based biochips." *Proceeding of IEEE Design, Automation and Test in Europe (DATE) Conference*, pp. 329-334, 2006.
- [18] A. J. Pfeiffer, T. Mukherjee and S. Huan, "Synthesis of multiplexed biofluidic microchips", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 2, pp.321-333, 2006.
- [19] International Technology Roadmap for Semiconductors, <http://public.itrs.net/Files/2003ITRS/Home2003.htm>.
- [20] F. Su, *Synthesis, Testing, and Reconfiguration Techniques for Digital Microfluidic Biochips*, Ph.D. thesis, Duke University, Durham, NC, USA, 2006.
- [21] X. Chen, D. F. Cui, C. Liu, H. Li and J. Chen, "Continuous flow microfluidic device for cell separation, cell lysis and DNA purification", *Analytica Chimica Acta*, vol. 584, pp.237-243, 2007.
- [22] M. G. Pollack, R. B. Fair and A. D. Shenderov, "Electrowetting-based actuation of liquid droplets for microfluidic applications", *Applied Physics Letters*, vol. 77, pp. 1725-1726, 2000.
- [23] V. Srinivasan, V. K. Pamula and R. B. Fair, "Droplet-based Microfluidic Lab-on-a-chip for Glucose Detection", *Analytica Chimica Acta*, vol. 507, no.1, pp.145-150, 2004.
- [24] Advanced Liquid Logic, Inc., <http://www.liquid-logic.com>.
- [25] J. Gong and C. J. Kim, "Two-dimensional digital microfluidic system by multi-layer printed circuit board", *Proceeding of IEEE MEMS*, 726-729, 2005.
- [26] K. F. Böhringer, "Modeling and controlling parallel tasks in droplet-based microfluidic systems", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, pp.329-339, 2006.

- [27] F. Su, W. L. Hwang and K. Chakrabarty, "Droplet routing in the synthesis of digital microfluidic biochips", *Proceeding of IEEE Design, Automation and Test in Europe (DATE) Conference*, pp. 323-328, 2006.
- [28] F. Su, S. Ozev and K. Chakrabarty, "Testing of droplet-based microelectrofluidic systems", *Proceeding of IEEE International Test Conference*, pp. 1192-1200, 2003.
- [29] F. Su, S. Ozev and K. Chakrabarty, "Ensuring the operational health of droplet-based microelectrofluidic biosensor systems", *IEEE Sensors Journal*, vol. 5, pp. 763-773, 2005.
- [30] H. G. Kerkhoff and M. Acar, "Testable design and testing of micro-electro-fluidic arrays", *Proceeding of IEEE VLSI Test Symposium*, pp. 403-409, 2003.
- [31] A. Dhayni, S. Mir, L. Rufer, A. Bounceur, "Pseudorandom functional BIST for linear and nonlinear MEMS", *Proceeding of IEEE Design, Automation and Test in Europe (DATE) Conference*, pp. 664-669, 2006.
- [32] H. G. Kerkhoff, "Testing of microelectronic-biofluidic systems", *IEEE Design & Test of Computers*, vol. 24, pp. 78-84, 2007.
- [33] H. G. Kerkhoff and H. P. A. Hendriks, "Fault modeling and fault simulation in mixed micro-fluidic microelectronic systems", *Journal of Electronic Testing Theory and Applications*, vol. 17, pp. 427-437, 2001.
- [34] F. Su, S. Ozev and K. Chakrabarty, "Test planning and test resource optimization for droplet-based microfluidic systems", *Journal of Electronic Testing: Theory and Applications*, vol. 22, pp. 199-210, 2006.
- [35] T. H. Schulte, R. L. Bardell and B. H. Weigl, "Microfluidic technologies in clinical diagnostics", *Clinica Chimica Acta*, vol. 321, pp. 1-10, 2002.
- [36] F. Su, W. Hwang, A. Mukherjee and K. Chakrabarty, "Testing and diagnosis of realistic defects in digital microfluidic biochips", *Journal of Electronic Testing: Theory and Applications*, vol. 23, pp. 219-233, 2007.
- [37] W. Hwang, F. Su and K. Chakrabarty, "Automated design of pin-constrained digital microfluidic arrays for lab-on-a-chip applications", *Proceeding of IEEE/ACM Design Automation Conference*, pp. 925-930, 2006.
- [38] V. Srinivasan, V. K. Pamula and R. B. Fair, "An integrated digital microfluidic lab-on-a-chip for clinical diagnostics on human physiological fluids", *Lab on a Chip*, vol. 4, pp. 310-315, 2004.
- [39] S.-K. Fan, C. Hashi and C.-J. Kim, "Manipulation of multiple droplets on $N \times M$ grid by cross-reference EWOD driving scheme and pressure-contact packaging", *Proceeding of*

- IEEE MEMS Conference*, pp. 694-697, 2003.
- [40] J. C. Kendrew, G. Bodo, H. M. Dintzis, R.G. Parrish, H. Wyckoff and D. C. Phillips, “A three-dimensional model of the myoglobin molecule obtained by x-ray analysis”, *Nature*, vol. 181, pp. 662-666, 1958.
- [41] A. McPherson, “Crystallization of macromolecules – general principles”, *Methods in Enzymology A*, vol. 114, pp. 112-120, 1985.
- [42] N. E. Chayen, P. D. Shaw Stewart, D.L. Maeder and D.M. Blow, “An automated system for micro-batch protein crystallization and screening”, *Journal of Applied Crystallography*, vol. 23, pp. 297-302, 1990.
- [43] J. R. Luft, D. M. Rak and G. T. DeTitta, “Microbatch macromolecular crystallization in micropipettes”, *Journal of Crystal Growth*, vol. 196, pp. 450-455, 1999.
- [44] <http://www.douglas.co.uk/oryx8.htm>.
- [45] <http://www.gilson.com/Applications/autoLiquidHandling.asp>.
- [46] <http://www.syrrx.com>.
- [47] R. C. Stevens, “High-throughput protein crystallization”, *Current Opinion in Structural Biology*, vol. 10, pp. 558-563, 2000.
- [48] H. I. Krupka, B. Rupp, B. W. Segelke, T. P. Lakin, D. Wright, H. C. Wu, P. Todd and A. Azarani, “The high-speed Hydra-Plus-One system for automated high-throughput protein crystallography”, *Acta Crystallographica*, vol. 58, pp. 1523-1526, 2002.
- [49] P. Y. Paik, V. K. Pamula, M. G. Pollack, and R. B. Fair, “Rapid droplet mixers for digital microfluidic systems”, *Lab on a Chip*, vol. 3, pp. 253-259, 2003.
- [50] T. Xu and K. Chakrabarty, “Integrated droplet routing and defect tolerance in the synthesis of digital microfluidic biochips”, *ACM Journal on Emerging Technologies in Computing Systems*, vol. 4., no. 3, article 11, 2008.
- [51] T. Xu, K. Chakrabarty and F. Su, “Defect-aware synthesis of droplet-based microfluidic biochips”, *Proceeding of IEEE International Conference on VLSI Design*, pp. 647-652, 2007.
- [52] F. Su and K. Chakrabarty, “Module placement for fault-tolerant microfluidics-based biochips”, *ACM Transactions on Design Automation of Electronic Systems*, vol. 11, pp. 682-710, 2006.
- [53] T. Xu and K. Chakrabarty, “Parallel scan-like test and multiple-defect diagnosis for

- digital microfluidic biochips”, *IEEE Transactions on Biomedical Circuits and Systems*, vol. 1, pp. 148-158, June 2007.
- [54] T. Xu and K. Chakrabarty, “Integrated droplet routing and defect tolerance in the synthesis of digital microfluidic biochips”, *ACM Journal on Emerging Technologies in Computing Systems*, vol. 4., no. 3, article 11, August 2008.
- [55] T. Xu, W. Hwang, F. Su and K. Chakrabarty, “Automated design of pin-constrained digital microfluidic biochips under droplet-interference constraints”, *ACM Journal on Emerging Technologies in Computing Systems*, vol. 3, article 14, 2007.
- [56] R. Diestel, *Graph Theory*, Berlin: Springer, 2005.
- [57] C. H. Papadimitriou, *Computational Complexity*. Reading, MA: Addison Wesley, 1993.
- [58] M. A. Breuer, “A class of min-cut placement algorithms”, *Proceeding of IEEE/ACM Design Automation Conference*, pp. 284-290, 1977.
- [59] A. Dunlop and B. Kernighan, “A procedure for placement of standard cell VLSI circuits”, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 1, pp. 92-98, 1985.
- [60] A. B. Kahng, I. I. Mandoiu, S. Reda, X. Xu and A. Zeilikovsky, “Evaluation of placement techniques for DNA probe array layout”, *Proceeding of IEEE/ACM International Conference on Computer-Aided Design*, pp. 262-269, 2003.
- [61] Connect5 strategies, www.springfrog.com/games/gomoku.
- [62] T. Xu and K. Chakrabarty, “A droplet-manipulation method for achieving high-throughput in cross-referencing-based digital microfluidic biochips”, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, pp. 1905-1917, 2008.
- [63] E. J. Griffith, S. Akella and M. K. Goldberg, “Performance characterization of a reconfigurable planar-array digital microfluidic system”, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, pp. 340-352, 2006.
- [64] T. Xu and K. Chakrabarty, “Broadcast electrode-addressing for pin-constrained multi-functional digital microfluidic biochips”, *Proceeding of IEEE/ACM Design Automation Conference*, pp. 173-178, 2008.
- [65] T. Xu and K. Chakrabarty, “Parallel scan-like test and multiple-defect diagnosis for digital microfluidic biochips”, *IEEE Transactions on Biomedical Circuits and Systems*, vol. 1, pp. 148-158, 2007.

- [66] T. Xu and K. Chakrabarty, "Functional testing of digital microfluidic biochips", *Proceeding of IEEE International Test Conference*, 2007.
- [67] M. G. Pollack, *Electrowetting-Based Microactuation of Droplets for Digital Microfluidics*, Ph.D thesis, Duke University, 2001.
- [68] Silicon Biosystems, <http://www.siliconbiosystems.com/applications/webwork/DEPArray.page>.
- [69] T. Xu, K. Chakrabarty and V. K. Pamula, "Design and optimization of a digital microfluidic biochip for protein crystallization", *Proceeding of IEEE/ACM International Conference on Computer-Aided Design*, 2008.
- [70] T. Xu, V. K. Pamula and K. Chakrabarty, "Automated, accurate and inexpensive solution-preparation on a digital microfluidic biochip", *Proceeding of IEEE International Conference on Biomedical Circuits and Systems*, 2008.
- [71] A. Moreno, E. Saridakis and N. E. Chayen, "Combination of oils and gels for enhancing the growth of protein crystals", *Journal of Applied Crystallography*, vol. 35, pp. 140-142, 2002.
- [72] <http://www.pcbdesign.org/pcb-layout/understanding-pcb-layers>.
- [73] E. Bach and J. Shallit, *Algorithmic Number Theory: Efficient Algorithms*, Cambridge, MA: MIT Press, 1996.
- [74] F. J. Scheid, *Schaum's Outline of Theory and Problems of Numerical Analysis*, New York: McGraw-Hill, 1968.

Biography

Full name: Tao Xu

Place of birth: Tsingdao, China

Date of birth: Nov. 19, 1982

Schools attended:

- Zhejiang University, Hangzhou, China, BSE, June 2005
- Zhejiang University, Hangzhou, China, BS, June 2005
- Duke University, Durham, NC, USA, MS, Jan 2007

Publications:

- T. Xu, W. Hwang, F. Su and K. Chakrabarty, “Automated design of pin-constrained digital microfluidic biochips under droplet-interference constraints”, *ACM Journal on Emerging Technologies in Computing Systems*, vol. 3, article 14, November 2007
- T. Xu and K. Chakrabarty, “Parallel scan-like test and multiple-defect diagnosis for digital microfluidic biochips”, *IEEE Transactions on Biomedical Circuits and Systems*, vol. 1, pp. 148-158, June 2007.
- T. Xu and K. Chakrabarty, “Integrated droplet routing and defect tolerance in the synthesis of digital microfluidic biochips”, *ACM Journal on Emerging Technologies in Computing Systems*, vol. 4., no. 3, article 11, August 2008.
- T. Xu and K. Chakrabarty, “Defect-aware high-level synthesis and module placement for microfluidic biochips”, *IEEE Transactions on Biomedical Circuits and Systems*, vol. 2, pp. 50-62, March 2008.
- T. Xu and K. Chakrabarty, “A droplet-manipulation method for achieving high-throughput in cross-referencing-based digital microfluidic biochips”, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, pp. 1905-1917, November 2008.
- T. Xu and K. Chakrabarty, “Droplet-trace-based array partitioning and a pin assignment algorithm for the automated design of digital microfluidic biochips”, *Proceeding of IEEE/ACM International Conference on Hardware/Software Codesign and System Synthesis*, pp. 112-117, 2006.

- T. Xu, K. Chakrabarty and F. Su, “Defect-aware synthesis of droplet-based microfluidic biochips”, *Proceeding of IEEE International Conference on VLSI Design*, pp. 647-652, 2007.
- T. Xu and K. Chakrabarty, “A cross-referencing-based droplet manipulation method for high-throughput and pin-constrained digital microfluidic arrays”, *Proceeding of Design, Automation and Test in Europe (DATE) Conference*, pp. 552-557, 2007.
- T. Xu and K. Chakrabarty, “Parallel scan-like testing and fault diagnosis techniques for digital microfluidic biochips”, *Proceeding of IEEE European Test Symposium*, pp. 63-68, 2007.
- T. Xu and K. Chakrabarty, “Integrated droplet routing in the synthesis of microfluidic biochips”, *Proceeding of IEEE/ACM Design Automation Conference*, pp. 948-953, 2007.
- T. Xu and K. Chakrabarty, “Functional testing of digital microfluidic biochips”, *Proceeding of IEEE International Test Conference*, 2007.
- T. Xu and K. Chakrabarty, “Automated design of digital microfluidic lab-on-chip under pin-count constraints”, *Proceeding of ACM International Symposium on Physical Design*, pp. 90-98, 2008 (Invited Paper).
- T. Xu and K. Chakrabarty, “Broadcast electrode-addressing for pin-constrained multi-functional digital microfluidic biochips”, *Proceeding of IEEE/ACM Design Automation Conference*, pp. 173-178, 2008.
- T. Xu, K. Chakrabarty and V. K. Pamula, “Design and optimization of a digital microfluidic biochip for protein crystallization”, *Proceeding of IEEE/ACM International Conference on Computer-Aided Design*, 2008.
- T. Xu, V. K. Pamula and K. Chakrabarty, “Automated, accurate and inexpensive solution-preparation on a digital microfluidic biochip”, *Proceeding of IEEE International Conference on Biomedical Circuits and Systems*, 2008.
- T. Xu, P. Thwar, V. Srinivasan, V. K. Pamula and K. Chakrabarty, “Digital microfluidic biochip for protein crystallization”, *IEEE-NIH Life Science Systems and Applications Workshop*, Bethesda, MD, 2007.
- Y. Zhang, K. Chakrabarty and T. Xu, “Digital microfluidic logic gates”, *Proceeding of 3rd International Conference on Nano-Networks*, 2008.

- Y. Zhao, T. Xu and K. Chakrabarty, “Built-in self-test and fault diagnosis for lab-on-chip using digital microfluidic logic gates”, *Proceeding of IEEE International Test Conference*, 2008.