

Continuous-Flow Biochips: Technology, Physical Design Methods and Testing

Paul Pop, *Member, IEEE*, Ismail Emre Araci and Krishnendu Chakrabarty, *Fellow, IEEE*

Abstract—Microfluidic biochips are replacing the conventional biochemical analyzers by integrating all the necessary functions for biochemical analysis using microfluidics. Biochips are used in many application areas, such as, in vitro diagnostics, drug discovery, biotech and ecology. The focus of this paper is on continuous-flow biochips, where the basic building block is a microvalve and, by combining these microvalves, more complex units such as mixers, switches, multiplexers can be built. Although the complexity of biochips is increasing, they are still designed manually, using software such as AutoCAD. Another roadblock in the deployment of microfluidic biochips is the lack of test techniques: defective chips lead to repetition of experiments, which is undesirable due to increased labor and high reagent cost. This paper presents the state-of-the-art in flow-based biochips technology and emerging research challenges in the areas of physical design and testing techniques.

Index Terms—Emerging technologies, Microfluidic biochips, Physical design, Testing

I. INTRODUCTION

Microfluidics-based biochips have become an actively researched area in recent years. Sometimes also referred to as lab-on-a-chip, biochips integrate different biochemical analysis functionalities (e.g., dispensers, filters, mixers, separators, detectors) on-chip, miniaturizing the macroscopic chemical and biological processes to a sub-millimetre scale [1]. These microsystems offer several advantages over the conventional biochemical analyzers, e.g., reduced sample and reagent volumes, speeded up biochemical reactions, ultra-sensitive detection and higher system throughput, with several assays being integrated on the same chip [2].

There are several types of microfluidic biochip platforms, each having its own advantages and limitations [3]. In this paper, we focus on the flow-based biochips in which the microfluidic channel circuitry on the chip is equipped with chip-integrated microvalves that are used to manipulate the on-chip fluid flow [1]. By combining several microvalves, more complex units like mixers, micropumps, multiplexers etc. can be built up, with hundreds of units being accommodated on

P. Pop is with the Department of Applied Mathematics and Computer Science, Technical University of Denmark. Address: Technical University of Denmark, Richard Petersens Plads, Building 322, office 126, DK2800 Kongens Lyngby, Denmark. Phone: +45 45 25 37 32. Fax: +45 45 88 13 99. Email: paupo@dtu.dk

I. E. Araci is with the Department of Bioengineering, Stanford University, USA. Address: Stanford University, James H. Clark Center E300, Stanford, CA 94305-5432. Phone: +1 (650) 721-2195. Fax: +1 (650) 724-5473. Email: earaci@stanford.edu

K. Chakrabarty is with the Electrical and Computer Engineering Department, Duke University, USA. Address: Electrical and Computer Engineering, Duke University, Box 90291, 130 Hudson Hall, Durham, NC 27708. Phone: +1 (919) 660-5244. Fax: +1 (919) 660-5293. Email: krish@duke.edu

one single chip. The technology is therefore referred to as “microfluidic Very Large-Scale Integration” (mVLSI) [4].

A. Technology and Fabrication

The key component of continuous-flow biochips is an on-chip micromechanical valve (Fig. 1a), which is analogous to a transistor in microelectronics [1]. The biochip has two logical layers: *flow layer* and the *control layer*. The fluid in the flow layer is manipulated using the control layer. A valve is formed at the cross section of channels in corresponding layers (point *a* in Fig. 1a). Typically, micromechanical valves are made of silicone rubber (polydimethylsiloxane, PDMS) and actuated by applying fluidic pressure to the elastomeric membrane. The external pneumatic air pressure that is applied to the membrane is controlled using a solenoid valve. Other valve technologies have been proposed, see [5] for a survey.

The fabrication of continuous-flow biochip devices is realized based on a simple yet effective microfabrication process called multilayer soft lithography (MSL). The standard MSL process starts with drawing the layers of the design in a computer aided design software such as AutoCAD. Researchers have started to propose top-down design flows, with the aim of replacing the manual drawing in AutoCAD with an automated synthesis process; see Section II for a discussion. Then, a photomask based on this design is used to produce molds by photolithography. The type of the resist that is used in mold making step determines the cross-section shape and height of the fluidic channel. Then two part silicone rubber (i.e., PDMS) is mixed and cast on to the corresponding molds for control and flow layer production. Depending on the type and requirements of the device, the casting of PDMS can be realized by spin coating (for thinner layers) or by simply pouring (for thicker layers) the liquid PDMS on the mold. Heat treatment of the liquid PDMS at 80 °C for at least 20 minutes partially cures and solidifies the PDMS which allows the layers to be cut and punched (for I/O access holes). Finally these layers are aligned and bonded on a glass substrate.

The technology of fabricating micromechanical valves at dimensions smaller than $10 \times 10 \mu\text{m}^2$ is called microfluidic Very

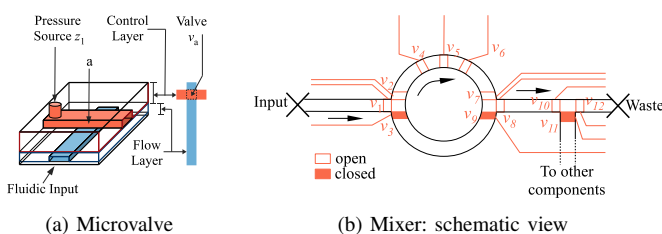


Fig. 1. Microvalve and microfluidic mixer

Large-Scale Integration (mVLSI) [4]. mVLSI technology is especially attractive for digital biology where single biological entities (e.g. proteins, enzymes, cells) are manipulated and/or quantified with high-throughput [6]. The standard MSL technique has been adapted for monolithic fabrication of the mVLSI chips [4].

B. Components and Architecture

Based on the basic micromechanical valve operation principle, many components have been developed, such as, pump, rotary mixer, multiplexer, sieve valves, filter [5], [7]. For a survey of recent component developments, see [5]. A mixer is a key requirement for laminar fluid flows where mixing only occurs by diffusion, e.g., for channel sizes larger than 10 μm . This becomes especially problematic for large molecules such as DNA because of the longer diffusion times (1 kbp DNA segment will diffuse 100 μm distance in 15 minutes). Although there are alternative mixing strategies reported in the literature, a rotary mixer (Fig. 1b) is an elegant solution to this problem [8]. Typically a channel loop with a few millimeter diameter and with dimensions of 100 μm wide by 10 μm high is used to build the rotary mixer shown in Fig. 1b. The valves here are marked as v_i , and v_4-v_6 forms a mixing pump. The series of on/off actuation sequences, such as 001, 011, 010, 110, 100, 101 are applied to operate this on-chip mixing pump. The components are interconnected and form a biochip architecture; Fig. 4b shows an architecture schematic, and Fig.2 shows an example physical layout.

C. Application Areas

Microfluidic platforms are used in many application areas [1], [9]–[13], such as, in vitro diagnostics (point-of-care, self-testing), drug discovery (high-throughput screening, hit characterization), biotech (process monitoring, process development), ecology (agriculture, environment, homeland security). They also offer exciting application opportunities in the realm of massively parallel DNA analysis, enzymatic and proteomic analysis, cancer and stem cell research, and automated drug discovery. Utilizing these biochips to perform

food control testing, environmental (e.g., air and water samples) monitoring and biological weapons detection are also interesting possibilities.

In high noise and variability systems (e.g. biological components and networks) high-throughput measurements are required to perform more accurate statistical analysis. The high level of automation and parallelism capability that is offered by high-throughput integration of the active components is especially well suited for single cell studies. As a result, there is an increase in the number of research studies that have been published in this field. This trend has also become apparent in the commercial domain with the marketing of single cell genomic analysis chip, C1, as the most recent product offered by the largest mVLSI company, Fluidigm [14].

Single cell genomic studies are especially important for cells that cannot be cultured with traditional methods such as microbes. For these cells, single-cell genomic approaches can be the only way to understand the connection between an organisms identity and the functional capabilities provided by its genome. The Whole Genome Amplification (WGA) chip [15], see Fig. 2, designed for this purpose can perform the critical functionalities required for single cell genomic analysis of microbes such as selection/transfer of a single cell to a lysis chamber, providing the stringent lysis conditions, and matching these conditions to different microbe types and finally amplification of the genomic content in chambers where amplification reagents and contents of the lysed cells are mixed together. Besides the automated control of these complex protocols, small reaction chambers (nanoliter volume) of the WGA chip have the advantage of improving the performance of biochemical amplifiers [16]. Typically multiple displacement amplification, which is an isothermal amplification scheme that uses random primers and that is based on the strand-displacement ability of $\phi 29$ DNA polymerase is used in whole genome amplification studies [16].

D. Motivation for Physical Design and Testing Techniques

Although biochips are becoming more complex everyday, Computer-Aided Design (CAD) tools for these chips are still in their infancy. Initial CAD research has been focussed on device-level physical modeling of components [17], [18]. Designers are using full-custom and bottom-up methodologies involving many manual steps to implement these chips. The challenges facing biochips are similar to those faced by microelectronics some decades ago. As in the microelectronics area, CAD tools will reduce the development costs, increase the design productivity and yield, and are the key to the further growth and market penetration of biochips. Researchers have proposed top-down synthesis methodologies for droplet-based biochips [19]. However, the architecture of the droplet-based chips differs significantly from the flow-based chips.

An important consideration for mVLSI is the reliability of the chip and the predictable behavior of the valves. It is found that some of the PDMS physical properties, hence fabrication yield, are dependent on the humidity, therefore the fabrication parameters have to be strictly controlled to maintain the high fabrication yield for mVLSI. The main point of failure is the

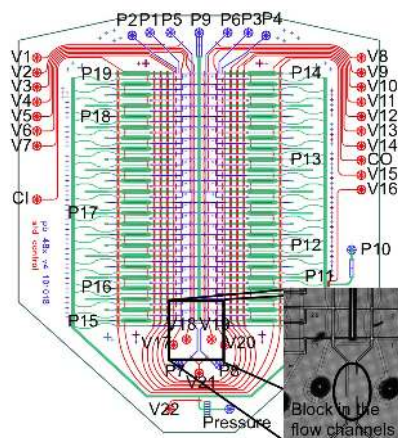


Fig. 2. Layout of the WGA chip [15]. The detail shows a fault.

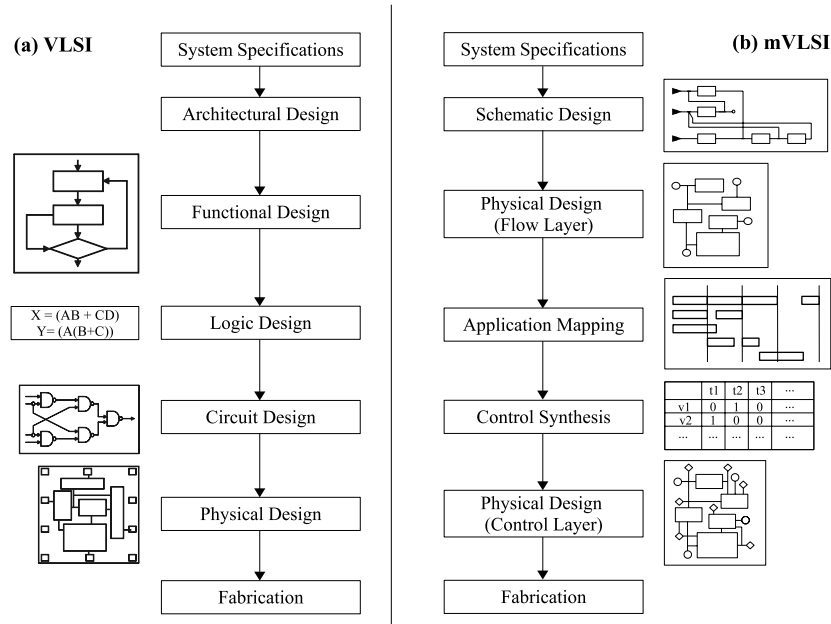


Fig. 3. VLSI vs mVLSI Design Flow

collapsing of the valve membrane and its irreversible bonding to either the flow or control channel. Recent experiments reveal that these failures are correlated with the large fluctuations in the relative humidity. As the chip density increases, fabrication constraints become tighter because a single faulty valve in a critical location can make an entire chip defective. The typical defects and their modeling are discussed in Section III-A.

Typically, for quality control, researchers examine the chips under the microscope before starting an experiment. This method has a very low throughput and it is labor-intensive, but most importantly the fault coverage (percentage of detectable faults) obtained using visual inspection is inadequate: defects can easily escape detection and some defects are invisible under the microscope even at high magnification. For example, valves which are not completely closed or leaky, or poorly bonded layers which could result in a short-circuit under pressure, are undetectable defects through visual inspection. Moreover, visual inspection can lead to an unnecessary yield loss. For example, when there is a slight misalignment between the layers, the chip could still be fully functional but can be considered as defective upon visual inspection. Also, debris trapped in between different layers may not affect functionality but a chip with debris on different layers may be classified as a defective chip by visual inspection [20]. Therefore, automated functional tests are important for the mass adoption of biochips. We discuss testing strategies in Section III-B.

II. PHYSICAL DESIGN

Fig. 3a shows a simplified design flow for *microelectronics* VLSI [21]. Motivated by the similarity between VLSI and mVLSI, researchers have proposed [22] the mVLSI design flow shown in Fig. 3b. The flow presented in the figure is an example, and the tasks may happen in a different order, and may be integrated with each other; also, there are backward feedback arrows to earlier steps, which are not depicted.

Recent research on mVLSI design methods has started to address design tasks in this design flow. An overview of the recent developments in mVLSI is presented in [5]. The design flow is supported by models: researchers have proposed a graph model for biochemical applications, where each node is an operation, and edges capture the fluid transport (see Fig. 4a) and a topology graph-based system-level model of a biochip architecture, which is independent of the underlying biochip implementation technology (see Fig. 4b) [22].

Given the system specifications (e.g., application requirements, chip area), the mVLSI design flow starts with the schematic design (*netlist*) of the required biochip. This is followed by the physical synthesis of the flow layer, i.e., placement of components and routing of flow channels while following the design rules. Researchers have proposed placement algorithms [22]–[24] for the flow layer, routing approaches for the flow layer [22], [25], as well as integrated approaches for the placement and routing [22].

Next, the given biochemical application is mapped onto this biochip architecture (the “Application Mapping” box). This step is described separately in Section II-C, and consists of operation binding, fluid routing, and scheduling. To perform detailed scheduling, we need the information on routing latencies for the fluids that traverse these channels, which can be determined only after the flow channels have been routed, and their lengths are known.

Researchers have started to propose approaches to the application mapping and scheduling [24], [26], [27]. Based on the schedule, the control information (which valves to open and close at what time and for how long) can now be extracted. Optimization schemes can be used to minimize the chip pin-count in the control layer, reducing the macro-assembly around the chip. This is followed by the control layer routing and then the chip design is ready to be sent for fabrication.

The following subsections explain the design tasks involved

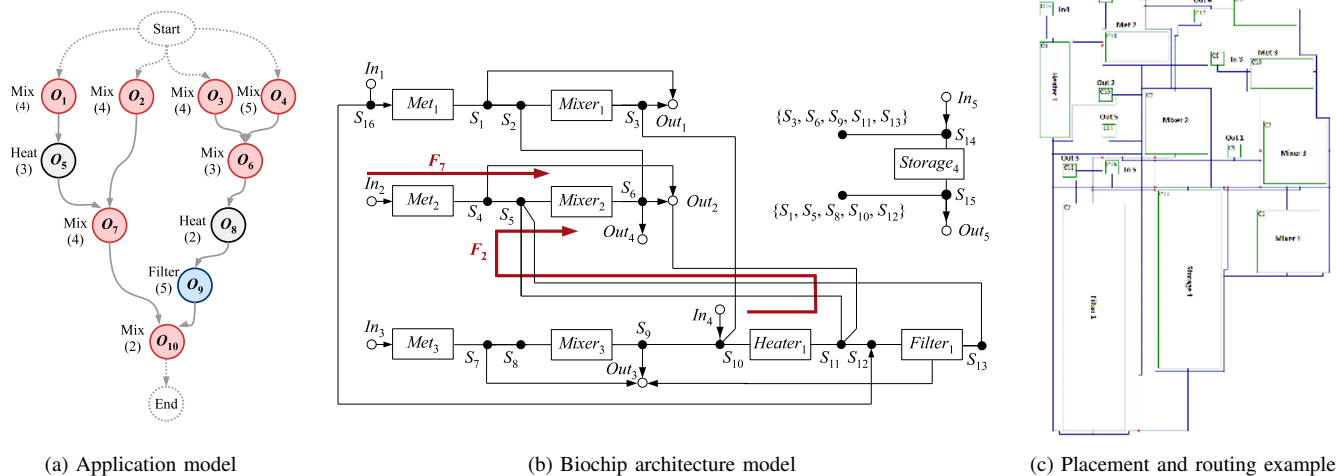


Fig. 4. Biochip application and architecture example

in the biochip synthesis using Fig. 4 as an illustrative example.

A. Allocation and Schematic Design

In this step, the microfluidic components required for implementing a given biochemical application are allocated from a component library, while taking into account the imposed resource constraints. Next, based on the given application, a chip schematic is designed and the netlist is generated. For example, to implement the biochemical application from Fig. 4a under the constraints given in Table I columns 1 and 2, we could use an allocation such as the one captured by the last two columns in Table I. The schematic design corresponding to such an application and allocation is presented in Fig. 4b. Note that the storage units are needed in order to save the output of a component so that it can be used at a later stage. The flow path set is also generated in this step. A *flow path* is the path starting from the point of fluid sample origin and ending at the fluid sample destination point, e.g., *Heater₁* to *Mixer₂* in Fig. 4b. Source-sink paths associated with each flow path are also defined, e.g., for the flow path *Heater₁* to *Mixer₂* in Fig. 4b, the source-sink path is (*In₄*, *S₁₀*, *Heater₁*, *S₁₁*, *S₅*, *Mixer₂*, *S₆*, *Out₂*). Routing constraints are also extracted at this stage. Two flow paths, whose corresponding source-sink paths have a common vertex are mutually exclusive and need to be listed as routing constraints, e.g., *F₇* and *F₂* in Fig. 4b are mutually exclusive since they share common vertices (e.g., *S₅*) in their source-sink paths.

TABLE I
ALLOCATED COMPONENTS

Function	Constraints	Allocated Units	Notations
Input port	5	5	<i>In₁ ... In₅</i>
Output port	5	5	<i>Out₁ ... Out₅</i>
Mixer	3	3	<i>Mixer₁ ... Mixer₃</i>
Heater	2	1	<i>Heater₁</i>
Filter	1	1	<i>Filter₁</i>
Metering Units	3	3	<i>Met₁ ... Met₃</i>
Storage Units	4	4	<i>Storage_x</i>

B. Physical Synthesis

In this step, the allocated components are placed on a chip layout area and the interconnections between components are routed as channels on the chip such that the application completion time is minimized. The placement and routing phases are governed by design rules imposed by the fabrication process. During placement, the components are treated as fixed size blocks, represented by rectangles, each having a fixed length and width. The placement is done in such a way that all design rules are satisfied and no two components overlap on the chip. For mVLSI-based biochips, the placement and routing phases can be divided into two stages, one for each logical layer in the chip: the flow layer and the control layer.

1) *Flow Layer*: This stage involves determining the placement of microfluidic components and the fluidic inlet/ outlet ports on the chip layout area, and then routing the interconnecting nets as microfluidic flow channels. In VLSI chips, the intersection of nets is considered a short-circuit and is thus not permitted. However, net intersection is possible in the biochip flow layer. A switch, composed of four valves controlling the channel intersection, is placed at the location of the intersection so that both channels can be used, at different points in time, without unintended fluid mixing. Considering that only one layer is available for routing all flow channel nets, the possibility of net intersection helps in achieving 100% routability. However, net intersections cause routing constraints, resulting in longer application completion times. Fig. 4c shows a placement and routing scheme for the flow layer of the biochip architecture shown in Fig. 4b.

2) *Control Layer*: In this stage, the placement of the control valves and the control ports is decided, and then the valves are connected to the control ports through control channel routing. The positions of the valves that are used inside a microfluidic component can be obtained directly from the component library. The positions of the valves that need to be placed on the flow channels are inferred from the flow routing information (e.g., valves need to be placed at all flow channel intersections). Contrary to the flow channels, control channels are not allowed to intersect.

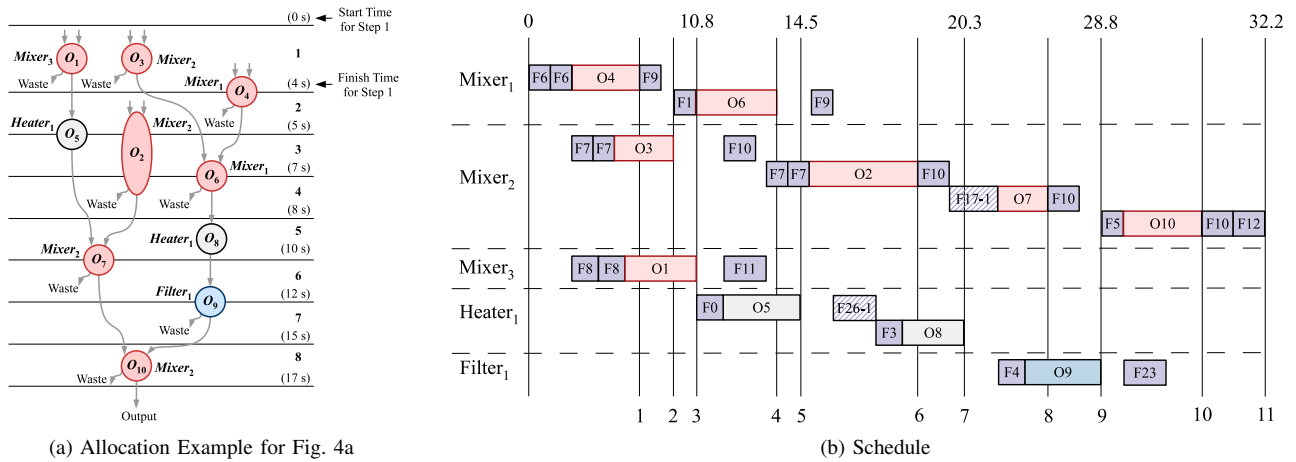


Fig. 5. Illustrative Example

After the placement is complete, the next step is to connect the valves to the ports using control channels. The control channels can be routed over/under any flow channel/component without forming a valve. The crossing of the control channel over a flow channel forms a valve only if the control channel has a larger width. The flow path channel lengths (used to calculate the routing latencies) and any additional routing constraints (imposed because of net intersections in the flow layer) can now be extracted from the layout and captured in the biochip architecture model.

C. Application Mapping

The next step is mapping the biochemical application such that the application completion time is minimized and the dependency, resource and routing constraints are satisfied. The binding for the operations can be the same as determined when generating the schematic, or can be modified in this step. The binding of the edges and the scheduling for both the operations and the edges is generated now. Fig. 5b shows the schedule for the case when the application in Fig. 4a is scheduled on the architecture in Fig. 4b. The schedule is represented as a Gantt chart, where, we represent the operations and fluid routing phases as rectangles, with their lengths corresponding to their execution duration.

D. Implementation and Experimental Evaluation

The biochip synthesis problem presented is NP-complete. Our synthesis strategy in this paper is to solve each design task separately, by adapting well-known heuristic algorithms from VLSI domain. We used resource-constrained List-Scheduling [22] for allocation, see Fig. 5a. We used Simulated

Annealing (one of the most used methods for cell placement in VLSI [21]) for performing component placement on the chip. Various algorithms have been proposed for routing over the years. We extended Hadlock's Algorithm [21] for the flow layer routing, and we modified the Pathfinder algorithm [28] for control channel routing. A variant of List Scheduling was also used for the application mapping [27].

We evaluated our proposed approach by synthesizing biochip architectures for three real life assays, see Table II: (1) PCR (polymerase chain reaction) mixing stage; 7 mixing operations; (2) Multiplexed IVD (in-vitro diagnostics); 12 operations; (3) CPA (Colorimetric Protein Assay); 55 operations. Column 1 presents the application and column 2 shows the list of allocated components, in the following format: (Input ports, Output ports, Mixers, Heaters, Filters, Detectors). Columns 3–5 present total length of the flow channels, total number of net intersections and the total number of valves on the chip, respectively. The imposed chip area is 250×250 . Chip area and total channel lengths are scaled, with a unit length being equal to $150 \mu\text{m}$. The last column presents the completion time δ_G of the application, in seconds, on the synthesized architecture.

III. TESTING

A. Defects and Fault Modeling

Let us now present some typical defects and describe how they can be modeled. For a more detailed discussion, see [20].

- **Block:** Microchannels may be disconnected, blocked, or in some cases, even missing. Fig. 6(a)-(c) shows some examples of block defects in fabricated microfluidic devices. The potential causes are environmental particles or imperfect silicon wafer mold.
- **Leak:** Some defective spots on the wall can connect independent micro-channels. The flows in either of them infiltrate into the other channel and the resulting cross-contamination can be catastrophic. It has been reported in [29] that the probability of a leaked channel pair increases as the length of the channels increases. It is higher if the distance between parallel channels decreases, and is less for channels that do not run in parallel. Fig. 6(d)-(f) shows some examples of leak defects caused by fiber

TABLE II
REAL-LIFE APPLICATIONS

Appl.	Allocated Units	Net Length	Total Inters.	Total Valves	δ_G
PCR	(3, 3, 3, 0, 0, 0)	198	4	67	19.7 s
IVD	(5, 5, 3, 0, 0, 3)	393	10	101	20 s
CPA	(5, 5, 5, 0, 0, 3)	1360	51	295	72.7 s

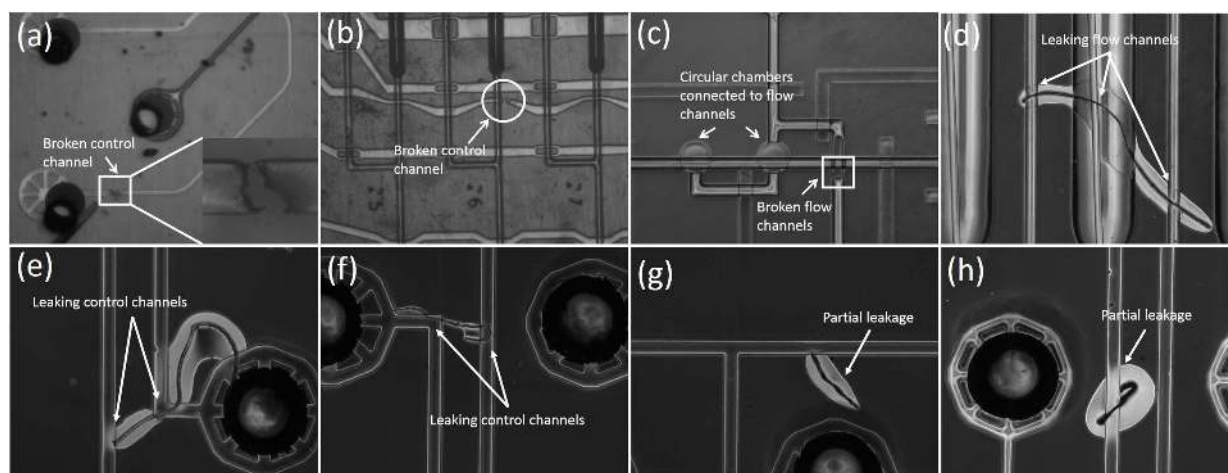


Fig. 6. Images of some typical visible defects in a fabricated flow-based microfluidic biochip.

TABLE III
FAULTY BEHAVIOR DUE TO DEFECTS IN THE TWO LAYERS.

	Flow Layer	Control Layer
Block	Fluid flow cannot go through the obstacle inside channel so transport is blocked.	Pressure cannot reach the flexible membrane, which prevents the corresponding valve from closing.
Leak	Fluid flow permeates the adjacent microchannels.	Control channels of two independent valves are unintentionally connected. Pressure on either valve activates both.

pollutant in fabricated microfluidic devices. Moreover, some partial leak defects are shown in Fig. 6(g)-(h). These defective spots might become fully leakage when high pressure is injected into the channels.

- *Misalignment*: Control layer and flow layer are misaligned. As a result, membrane valves either cannot be closed or are not even formed. The corresponding faulty behavior is similar to that of a block in the control channels.
- *Faulty pumps*: Pumps with defects fail to generate pressure when actuated. The faulty behavior here is similar to that for block; it interrupts the transmission of pressure.
- *Degradation of valves*: The membranes of valves might lose their flexibilities or even be perforated after a large number of operations. A consequence of this defect is that the valves cannot seal flow channels.
- *Dimensional errors*: The fabricated microchannels might be too narrow in comparison to the designed dimensions. The mismatch of height-to-width ratio may lead to a valve that cannot be closed; as a result, the flow cannot be stopped in flow channels underneath the valve.

Despite the complexity of flow-based microfluidic biochips, the consequence of the above defects can be described as either a block or a leak. While these two generic fault types (block and leak) can be observed in both layers, their respective faulty behaviors are different (Table III).

We next make the observation that the errors due to defects can be modeled in terms of faulty behaviors of valves. For example, a block in a flow channel can be modeled as a valve that cannot be opened (deactivated), while a block in a control channel can be represented by valves that cannot be closed (activated). Similar behavioral models can be defined for leaks.

B. Testing Strategy

Researchers have recently started to propose testing approaches for mVLSI biochips [20]. Here, we report on a possible test strategy, presented in [20]. For testing, feedback signals are needed to identify chip conditions. However, for flow-based microfluidic biochips, only inlets and outlets are available to communicate with the outside environment. Therefore, we use a test set-up where feedback is generated when pressure sensors are connected to the outlets and pumps are connected to the inlets. If there is a path between pump sources (inlets) and pressure sensors (outlets), pressure sensors at the outlets detect a high pressure generated by the pumps. The measured high pressure is defined as output “1”. If all routes between inlets and outlets are blocked, pressure sensors cannot sense the high pressure injected by the pumps. The absence of high pressure is defined as output “0”. In flow-based biochips, all ports are physically identical, regardless of the functional classification of inlets and outlets. During testing, only one of the ports in the flow layer is connected to a pressure source, while the rest are connected to pressure sensors. Similarly, a set of definitions for valve conditions is formulated. A “1” at a valve means that the valve is deactivated, i.e., low pressure in the control channel, while “0” indicates that the valve is activated, i.e., high pressure in the control channel. A binary pattern, also known as a test vector, is applied to all valves to set their open/close states. The actual responses of pressure sensors are compared to the expected responses. The microfluidic biochip is considered good if the two sets of responses match.

Table IV illustrates the test strategy to target the faults in Table III for the design in Fig. 7a. The test effectiveness depends on the quality of test patterns. As expected, the more complicated the microfluidic biochip structure is, the harder

TABLE IV
TESTING STRATEGY FOR DIFFERENT KINDS OF FAULTS.

	Flow Channel	Control Channel
Block	Position: $g-h$. Both valves g and h are deactivated to form a route $inlet-a-g-h-i-k-O2$. If the output at $O2$ is “0”, the defect is detected.	Position: valve h . The block in control layer prevents valve from closing. Deactivate valve a, g, i, k and $O2$ but activate the rest, including valve h . If $O2$ is “1”, the defect is detected.
Leak	Position: between $b-c$ & $g-h$. Deactivate valve a, b, h, i and k . If high pressure is sensed at $O2$, the leaking defect is detected.	Position: valve f & h . Turn on valve a, g, h, i, k but activate f . If there is a leakage, high pressure in control channel f will activate valve h and therefore block route.

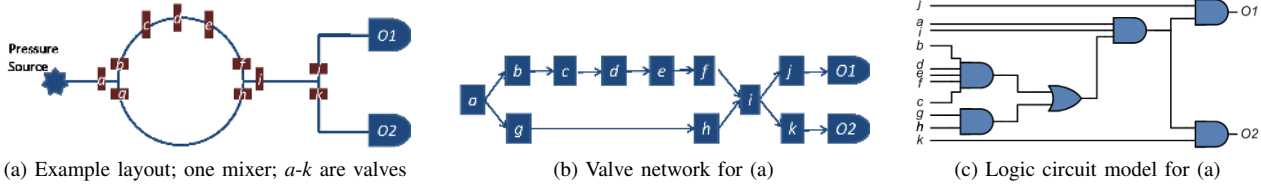


Fig. 7. Example models used for testing

it is to determine a test pattern set that covers every fault type for each valve and channel. Therefore, it is necessary to further abstract defects and microfluidic structures to facilitate automatic test-vector generation.

Defects in both flow channels and control channels can be modeled as the faulty behavior of a valve. Furthermore, a binary logic framework can be defined whereby an activated valve and a deactivated valve can be defined as logic “0” and “1”, respectively.

According to valve-based fault analysis, all types of defects occurring in both control channels and flow channels can be mapped to a specific behavioral-level fault at a valve. Such a classification simplifies the test problem for a 3D structure to that for a 2D design. It also simplifies test generation for chips with complicated networks of channels and valves.

For ease of description and analysis of biochip channel networks, we develop a discretized schematic of a valve network in place of a continuous fluid-flow topology. Fig. 7b illustrates an example for the design of Fig. 7a. Logic relationships that define flow-based biochips can be inferred from this schematic, e.g., valve b is serially connected to valve c, d, e and f . Therefore, either of these valves can potentially block the route, i.e., there is an “AND” logic relationship among them. On the other hand, routes $b-f$ and $g-h$ are in parallel, hence the activation of either of the two routes can lead to output “1”, i.e., high pressure sensed by the corresponding pressure sensor. There is an “OR” logic relationship between them. We can thereby further abstract flow-based biochips from the intermediate schematic representative of valve networks to valve-based logic gate circuit diagrams, as shown in Fig. 7c, whose logic expression is $\{O1, O2\} = \{j, k\} \cdot a \cdot i \cdot (b \cdot c \cdot d \cdot e \cdot f + g \cdot h)$. The primary inputs are nodes in the schematic of Fig. 7b.

We list two important attributes of the logic circuit model: (1) Only primary inputs (valves) and outputs (pressure sensors) have physical meaning. All other circuit connections are used to represent logical relationships. As a result, we only need to target faults at the primary inputs of this circuit. (2) A series connection of valves in a flow route is mapped to an AND gate. On the other hand, a parallel connection of valves is mapped to an OR gate.

We note that a physical defect in a flow-based biochip can be mapped to a fault at a primary input of a logic circuit. For example, to target a block defect in flow channel $g-h$, we can first map this defect to a stuck-at-0 fault, and after that this fault is associated with the primary input g in the logic circuit model (Fig. 7c). Similarly, a leak defect between valve f and h can be represented by an AND bridge fault between primary inputs f and h of Fig. 7c. Based on the logic circuit model, we can readily determine the actual (with faults) and expected (fault-free) responses of pressure sensors and therefore accelerate the search for test stimuli. If the actual outputs are different from the expected ones, we can not only conclude that the chip is faulty, but also infer the positions and types of defects. The logic circuit model therefore provides a concise representation and we can use Automatic Test Pattern Generation (ATPG) algorithms and tools for test-stimuli generation.

C. Applications to Fabricated Biochip

We used the WGA chip [15] for validating the testing approach. The chip is first modeled as a logic circuit using the method discussed in Section III-B, and after that test patterns are generated by TetraMAX, an ATPG tool from Synopsys. The chip contains 235 valves, 9 ports in the flow layers, and 23 ports in the control channels. The chip layout is shown in Fig. 2. Control channels are shown in red. The blue and green flow channels have different dimensions. Therefore, their connections can be tested by assign a pressure source at either of them and a pressure sensor at the other. The rest of chip can be tested by 12 test vectors, which are shown in Table V. The port “Pressure” is connected to a pressure source.

A fault-free chip and a defective chip with block defects shown in Fig. 2 are tested. As expected, all sensor feedback data match the expected responses for the fault-free chip. In the case of the defective chip, pressure sensors report errors at Test Pattern 10 and 11 due to the block defects.

IV. CONCLUSION AND FUTURE WORK

In this paper we have addressed continuous-flow mVLSI biochips, based on the manipulation of fluids through fab-

TABLE V
TEST PATTERNS FOR WGA CHIP AND THEIR EXPECTED FAULT-FREE RESPONSES.

	Test Pattern	Expected Response
1	11111 11111 11111 11111 10	00000 00000 00000 0000
2	01011 01100 10111 10011 01	00000 00001 00000 0000
3	10110 01111 11110 01111 11	00000 00000 00000 0000
4	10111 11011 11101 01111 01	11000 00010 00000 0000
5	01011 11111 01110 00011 01	00001 10000 00000 0000
6	11011 01011 11011 01111 11	00000 00000 00000 0000
7	01011 00111 11111 01111 11	00000 00000 00000 0000
8	11001 01011 01111 01010 11	00000 00000 00000 0000
9	01011 01111 10111 01100 11	00000 00000 00000 0000
10	10110 11010 11111 00101 11	00000 01110 11110 0000
11	11111 11111 11111 01111 11	11111 11110 11111 1111
12	11111 01111 11111 01111 11	00000 00000 00000 0000

ricated micro-channels, where the basic building block is a microvalve. Although they are a key enabling technology for several application areas, potential roadblocks in the deployment of microfluidic biochips are the lack of physical design tools and test techniques. Prior work on design tools and testing in biochips has been limited to digital (“droplet”) microfluidics. Recent work has addressed top-down physical design and automated testing of mVLSI biochips, and this paper has reported on such techniques. We hope that more work will be done in this area in the future, to bring the same level of automation to the design and testing of mVLSI biochips, such as the one taken for granted in microelectronics.

REFERENCES

[1] T. Thorsen, S. J. Maerkl, and S. R. Quake, “Microfluidic large-scale integration,” *Science*, vol. 298, no. 5593, pp. 580–584, October 2002.

[2] G. M. Whitesides, “The origins and the future of microfluidics,” *Nature*, vol. 442, pp. 368–373, July 2006.

[3] D. Mark, S. Haerberle, G. Roth, F. Stetten, and R. Zengerle, “Microfluidic lab-on-a-chip platforms: requirements, characteristics and applications,” *Chem. Soc. Rev.*, vol. 39, pp. 1153–1182, 2010.

[4] I. E. Araci and S. R. Quake, “Microfluidic very large scale integration (mVLSI) with integrated micromechanical valves,” *Lab Chip*, vol. 12, pp. 2830–2806, 2012.

[5] I. E. Araci and P. Brisk, “Recent developments in microfluidic large scale integration,” *Current opinion in biotechnology*, vol. 25, pp. 60–68, 2014.

[6] D. Witters, B. Sun, S. Begolo, J. Rodriguez-Manzano, W. Robles, and R. F. Ismagilov, “Digital biology and chemistry,” *Lab on a Chip*, vol. 14, pp. 3225–3232, 2014.

[7] J. Melin and S. R. Quake, “Microfluidic large-scale integration: The evolution of design rules for biological automation,” *Annual Review of Biophysics and Biomolecular Structure*, vol. 36, pp. 213–231, 2007.

[8] H.-P. Chou, M. A. Unger, and S. R. Quake, “A Microfabricated Rotary Pump,” *Biomedical Microdevices*, vol. 3, pp. 323–330, Dec 2001.

[9] J. W. Hong, Y. Chen, W. F. Anderson, and S. R. Quake, “Molecular biology on a microfluidic chip,” *Journal of Physics: Condensed Matter*, vol. 18(18), pp. 691–701, 2006.

[10] J. W. Hong, V. Studer, G. Hang, W. F. Anderson, and S. R. Quake, “A nanoliter-scale nucleic acid processor with parallel architecture,” *Nature Biotechnology*, vol. 22(4), pp. 435–439, 2004.

[11] S. E. et. al., “Discovery of a hepatitis C target and its pharmacological inhibitors by microfluidic affinity analysis,” *Nature Biotechnology*, vol. 12, pp. 1019–1027, 2008.

[12] C. D. C. et. al., “Microfluidics-based diagnostics of infectious diseases in the developing world,” *Nature Medicine*, vol. 17, pp. 1015–1019, 2011.

[13] H. C. Fan, Y. J. Blumenfeld, U. Chitkara, L. Hudgins, and S. R. Quake, “Noninvasive diagnosis of fetal aneuploidy by shotgun sequencing DNA from maternal blood,” *Proceedings of the National Academy of Sciences USA*, vol. 105(42), pp. 16266–16271, 2008.

[14] www.fluidigm.com/products/c1-system.

[15] P. Blainey, S. Quake, “Digital MDA for enumeration of total nucleic acid contamination,” *Nucleic Acids Research Advance Access*, Nov. 2010.

[16] P. C. Blainey and S. R. Quake, “Dissecting genomic diversity, one cell at a time,” *Nature Methods*, vol. 11, pp. 349–349, Mar 2014.

[17] J. Siegrist et al., “Numerical modeling and experimental validation of uniform microchamber filling in centrifugal microfluidics,” *Lab Chip*, vol. 10, pp. 876–886, 2010.

[18] I. Klammer et al., “Numerical analysis and characterization of bionic valves for microfluidic pdms-based systems,” *Journal of Micromechanics and Microengineering*, vol. 17, no. 7, pp. S122–S127, 2007.

[19] K. Chakrabarty and T. Xu, *Digital Microfluidic Biochips: Design Automation and Optimization*. Boca Raton, FL: CRC Press, 2010.

[20] K. Hu, F. Yu, T.-Y. Ho, and K. Chakrabarty, “Testing of Flow-Based Microfluidic Biochips: Fault Modeling, Test Generation, and Experimental Demonstration,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 33, pp. 1463–1475, Oct 2014.

[21] S. M. Sait and H. Youssef. *VLSI physical design automation: theory and practice*. World Scientific Publishing Co. Pte. Ltd., 1999.

[22] W. H. Minhass, P. Pop, J. Madsen, and F. S. Blaga, “Architectural synthesis of flow-based microfluidic large-scale integration biochips,” in *Proceedings of the International Conference on Compilers, Architectures and Synthesis of Embedded Systems*, pp. 181–190, 2012.

[23] J. McDaniel, B. Parker, and P. Brisk, “Simulated Annealing-based Placement for Microfluidic Large Scale Integration (mLSI) Chips,” in *Proceedings of the International Conference on Very Large Scale Integration*, pp. 213–218, 2014.

[24] K.-H. Tseng, S.-C. You, J.-Y. Liou, and T.-Y. Ho, “A top-down synthesis methodology for flow-based microfluidic biochips considering valve-switching minimization,” in *Proceedings of the International Symposium on Physical Design*, pp. 123–129, 2013.

[25] C.-X. Lin, C.-H. Liu, I.-C. Chen, D. Lee, and T.-Y. Ho, “An efficient bi-criteria flow channel routing algorithm for flow-based microfluidic biochips,” in *Proceedings of the Design Automation Conference*, pp. 1–6, 2014.

[26] T. A. Dinh, S. Yamashita, T.-Y. Ho, and Y. Hara-Azumi, “A clique-based approach to find binding and scheduling result in flow-based microfluidic biochips,” in *Proc. of the Asia and South Pacific Design Automation Conference*, pp. 199–204, 2013.

[27] W. H. Minhass, P. Pop, and J. Madsen, “System-level modeling and synthesis of flow-based microfluidic biochips,” in *Proceedings of the International Conference on Compilers, Architectures and Synthesis of Embedded Systems*, pp. 225–234, 2011.

[28] L. McMurchie, C. Ebeling “PathFinder A Negotiation-based Performance-driven Router for FPGAs” In *Proceedings of the 1995 ACM third international symposium on Field-programmable gate arrays*, pp. 111–117. ACM, 1995.

[29] J. Wang et al., “Optimal Protocol for Molding PDMS with A PDMS Master”, *Chips & Tips (Lab on a Chip)*, 06 July 2010.

Paul Pop has received his Ph.D. degree from University of Linköping in 2003. He is an associate professor at the Department of Applied Mathematics and Computer Science, Technical University of Denmark. His research interests are related to the design and programming of microfluidic biochips, and safety-critical systems. He a Member of IEEE and is the Chairman of the IEEE Danish Chapter on Embedded Systems.

I. Emre Araci has received his Ph.D. degree from University of Arizona in 2010. He then joined Prof. Stephen Quakes group in the Bioengineering Department at Stanford University as a postdoctoral associate and as the director of the Stanford Microfluidics Foundry. His is interested in the development and application of implantable and miniaturized micro/optofluidic technologies for biology and medicine. His recent work was recently published in *Nature Medicine*.

Krishnendu Chakrabarty received the Ph.D. degree from the University of Michigan in 1995. He is the William H. Younger Distinguished Professor in the Department of Electrical and Computer Engineering and at Duke University. Prof. Chakrabarty's current research projects include: design-for-testability and resilience in integrated circuits; microfluidic biochips and cyberphysical systems; optimization of enterprise systems. He is a Fellow of ACM and a Fellow of IEEE.