Microfluidic Very Large-Scale Integration for Biochips: Technology, Testing and Fault-Tolerant Design

Ismail Emre Araci
Department of Bioengineering,
Stanford University, USA
earaci@stanford.edu

Paul Pop
DTU Compute Dept.,
Technical University of Denmark
paupo@dtu.dk

Krishnendu Chakrabarty
Electrical and Computer Engineering Dept.,
Duke University, USA
krish@duke.edu

***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. By combining these microvalves, more complex units such as mixers, switches, multiplexers can be built, hence the name of the technology, "microfluidic Very Large-Scale Integration" (mVLSI). A roadblock in the deployment of microfluidic biochips is their low reliability and lack of test techniques to screen defective devices before they are used for biochemical analysis. Defective chips lead to repetition of experiments, which is undesirable due to high reagent cost and limited availability of samples. This paper presents the state-of-the-art in the mVLSI platforms and emerging research challenges in the area of continuous-flow microfluidics, focusing on testing techniques and fault-tolerant design.**

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

The next three subsections present the technology, components and application areas for mVLSI biochips. Section II discusses testing strategies and Section III discusses an approach to fault-tolerant design.

A. Technology and Fabrication

The key component of continuous-flow biochips is an on-chip micromechanical valve (Fig. 1), which is analogous to a transistor in microelectronics [1]. The biochip has two logical layers: *flow layer* and the *control layer*. The liquid in the flow layer is manipulated using the control layer. A valve is formed at the cross section of channels in corresponding layers. 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 [28] 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 III 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 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 10x10 μm^2 is called microfluidic Very

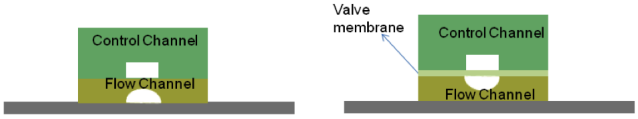


Fig. 1. Two layer valve cross-section (left) and three layer valve cross-section (right) for mVLSI [4].

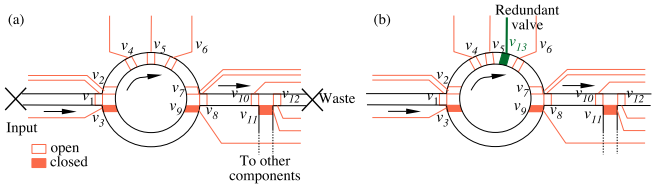


Fig. 2. Rotary mixer: (a) regular design and (b) fault-tolerant design

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 [5]. Standard multilayer soft lithography technique is adapted for monolithic fabrication of the mVLSI chips [4]. The main difference of mVLSI process compared to the standard fabrication process is the addition of a third layer as the valve membrane as shown in Fig. 1b. This thin valve membrane is obtained by spin coating PDMS on a blank silicon wafer at very high speeds and extended spin durations. This results in highly uniform films with a thickness as small as 0.3 μm. It is also observed that as the PDMS cross-linker mixing ratio reduced from 1:10 to 1:30, the resulting film thickness is reduced from 1 μm to 0.3 μm for a spin speed of 12,000 rpm, and spin duration of 15 min. Due to its low viscosity and low Youngs modulus (E), PDMS with low cross-linker ratio (1:30) is typically used for the fabrication of valve membranes for mVLSI.

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 [6], [28]. For a survey of recent component developments, see [28]. A mixer is a key requirement for mVLSI biochips where fluid flow is laminar thus mixing only occurs by diffusion. This becomes especially problematic for large molecules such as DNA because of the longer diffusion times (1 kbp DNA segment will diffuse 100 m distance in 15 minutes). Although there are alternative mixing strategies reported in the literature, a rotary mixer (Fig. 2a) is an elegant solution to this problem [7]. 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. 2a. 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 maximum cycling frequency is around 30 Hz and this gives complete mixing of even the largest objects in about 30 seconds. We discuss a possible fault-tolerant mixer design (Fig. 2b) in Section III.

C. Application Areas

Microfluidic platforms are used in many application areas [1], [8]–[17], 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 [18].

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 [20], see Fig. 3, 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 [19]. Typically multiple displacement amplification, which is an isothermal amplifica-

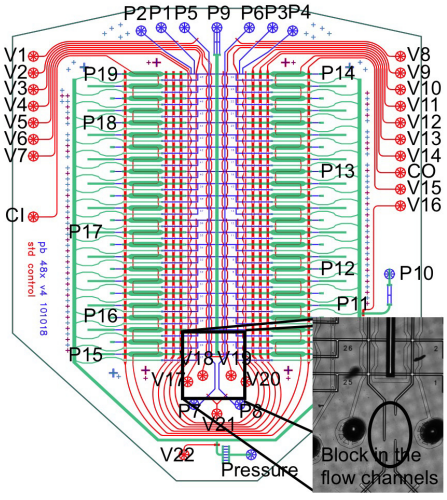


Fig. 3. Layout of the WGA chip [20]. The detail shows a fault.

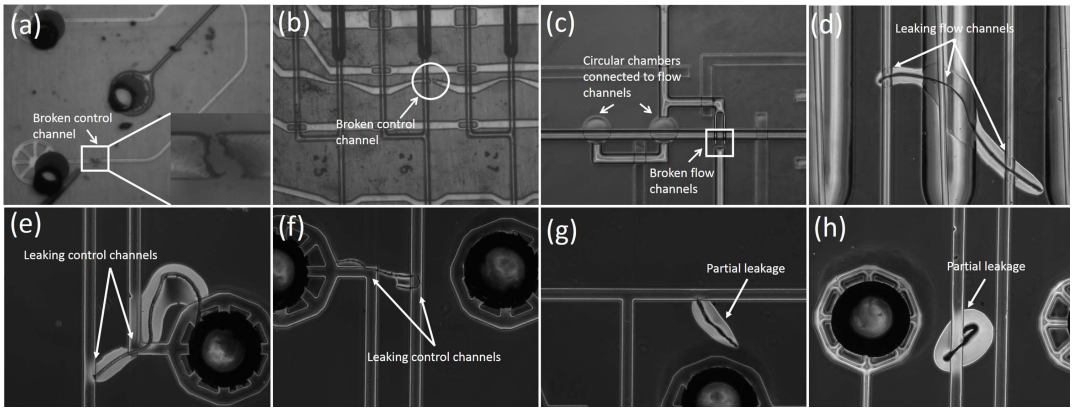


Fig. 4. Images of some typical visible defects in a fabricated flow-based microfluidic biochip.

tion scheme that uses random primers and that is based on the strand-displacement ability of φ29 DNA polymerase is used in whole genome amplification studies [19].

D. Motivation for Testing and Fault-Tolerant Design

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 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 is discussed in Section II-A.

Therefore, for more reliable chips, better quality control methods have to be developed and alternative components/paths have to be added to the chip design to perform critical functions in a chip. 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 [21].

Therefore, an automated functional test is important for mass adoption of mVLSI because, improved reliability through detecting all of the defects that can interfere with an experiment before starting to work with the chip, will stimulate the working environment. We discuss testing strategies in Section II-B.

Besides automated tests, designing alternative paths and components to perform critical functionalities can be another key strategy in improving the reliability of the mVLSI chips. Due to the small dimensions of the channels, valves can be placed redundantly at some critical locations without requiring a large chip area. When a single redundant valve is defective, the function of this valve will be performed by the other valves at this location. Such an example is the fault-tolerant mixer in Fig. 2b, which uses an extra valve in the pump component (v_{13}). Section III has more details, and proposes an approach to fault-tolerant design.

II. Testing

A. Defects and Fault Modeling

Let us now present the typical defects and how they can be modeled. For a more detailed discussion, see [21].

- *Block:* Microchannels may be disconnected, blocked, or in some cases, even missing. Fig. 4(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 [22] 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. 4(d)-(f) shows some examples of leak defects caused by fiber pollutant in fabricated microfluidic devices. Moreover, some partial leak defects are shown in Fig. 4(g)-(h). These

TABLE I
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.

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 I).

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

Testing strategies have been proposed for droplet-based biochips, which manipulate the fluids as droplets [23], but they are not applicable to mVLSI biochips, which manipulate the fluid as a continuous flow. Researchers have recently started to propose testing approaches for mVLSI biochips [21]. Here, we report on a possible test strategy, presented in [21].

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

Logic	Valve state	Valve condition	Pressure response
1	open	deactivated	high
0	closed	activated	low

TABLE II
LOGIC REPRESENTATION OF VALVE STATES AND PRESSURE RESPONSE.

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. Table II connects the logic representation of valve states to the corresponding pressure response. 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 III illustrates the test strategy to target the faults in Table I for the design in Fig. 5a. The test effectiveness depends on the quality of test patterns. As expected, the more complicated the microfluidic biochip structure is, the harder 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.

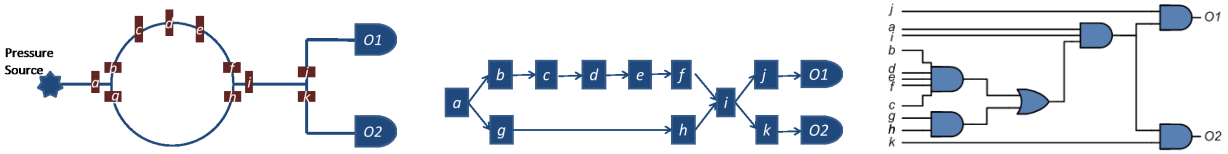
Defects in both flow channels and control channels can be modeled as the faulty behavior of a valve. Furthermore, a binary logic framework can be defined whereby an activated valve and a deactivated valve can be defined as logic "0" and "1", respectively. Hence, Table IV defines behavioral-level fault models for a flow-based microfluidic biochip.

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. 5b illustrates an example for the design of Fig. 5a. 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*.

TABLE III
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.



(a) Example layout; one mixer; *a-k* are valves (b) Valve network for (a) (c) Logic circuit model for (a)

Fig. 5. Example models used for testing

TABLE IV
BEHAVIORAL-LEVEL FAULT MODEL FOR FLOW-BASED BIOCHIPS.

	Flow Layer	Control Layer
Block	stuck-at-0	stuck-at-1
Leak	OR bridge (1-dominant)	AND bridge (0-dominant)

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. 5c, 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. 5b.

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.

Therefore, based on Fig. 5c and Table IV, 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 according to Table IV, and after that this fault is associated with the primary input *g* in the logic circuit model (Fig. 5c). 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. 5c. 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 [20] for validating the testing approach. The chip is first modeled as a logic circuit using the method discussed in Section II-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. 3. Control channels are shown in red. The blue and green flow channels have different dimensions. Therefore, their connections can be tested be 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. 3 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.

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

III. Fault-Tolerant Design

As we have discussed in the earlier sections, the consequences of application failure can be costly, so testing methods are needed to identify defective biochips. To increase the yield, and to potentially also prevent the failure *during* the operation of the biochip, we advocate the use of fault-tolerant biochip design. When the consequences of failure are drastic, researchers have already considered introducing redundancy to provide fault-tolerance. Such an example is the "Mars Organic Analyzer" biomarker detector chip, see [24]. Because failure on Mars is extremely costly (no experiments will be possible on Mars if the chip fails), the biochip has been designed to be able to tolerate faults, i.e., it uses uses extra valves to ensure that a redundant route can be formed if the valves pumping into the sample reservoir fail.

The vision is to provide application fault-tolerance at run-time (online), detecting the faults as they appear, and reconfiguring the application. However, in this paper our assumption is that the faults are detected during testing, and that the operation of the biochip is reconfigured offline (at design time) to avoid the faults. We are interested to introduce redundancy such that the applications can still successfully run on a defective biochip. Redundancy is the addition of extra *resources*, normally not needed for correct operation, to be used for fault-tolerance.

We propose a fault-tolerant design strategy, which is part of an overall mVSLI physical design flow. 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 [25], [26]. Designers are using full-custom and bottom-up methodologies involving many manual steps to implement these chips. Researchers have proposed top-down synthesis methodologies for droplet-based biochips

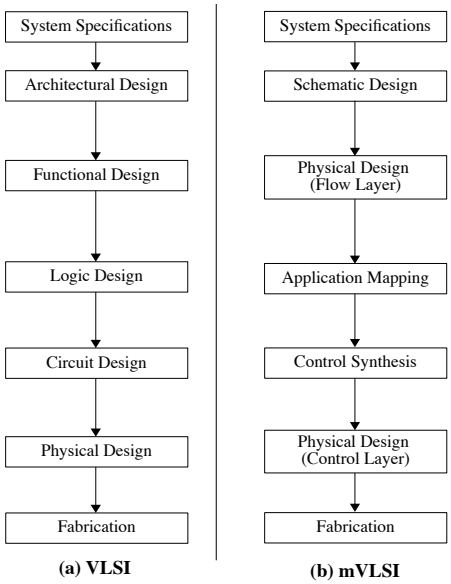


Fig. 6. VLSI vs mVLSI Design Flow

[27]. However, the architecture of the droplet-based chips differs significantly from the flow-based chips.

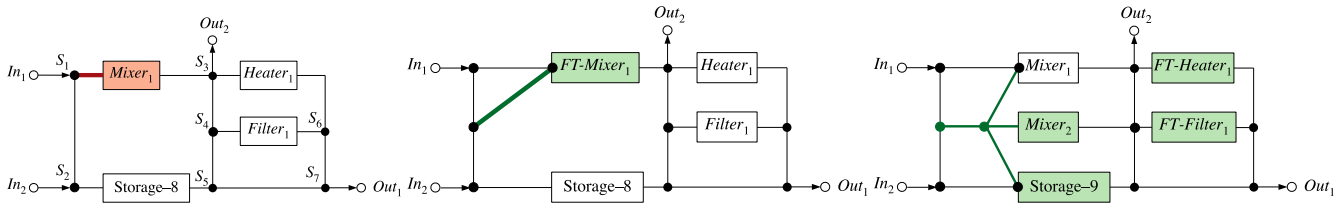
Fig. 6a shows a simplified design flow for *microelectronics* VLSI. Motivated by the similarity between VLSI and mVLSI, researchers have proposed [29] the mVLSI design flow shown in Fig. 6b. Recent research on mVLSI design methods has started to address design tasks in this design flow. We will refer to these results when describing the design tasks. An overview of the recent developments in mVLSI is presented in [28]. 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 [29]–[31] for the flow layer, routing approaches for the flow layer [29], [32], as well as integrated approaches for the placement and routing [29].

After the flow channels have been routed, the channel lengths and therefore the routing latencies for the fluids that traverse these channels can now be calculated. Next, the given biochemical application is mapped onto this biochip architecture and the optimized schedule for its execution is generated (the "Application Mapping" box). Researchers have started to propose approaches to the application mapping and scheduling [31], [33], [34]. 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. Recent research has addressed both the control-pin minimization [35] and the control channel routing.

Fault-tolerant design strategies have been proposed for droplet-based biochips; these biochips have a regular array structure, composed of electrodes which manipulate the droplets. In this context, fault-tolerance means introducing redundant electrodes, in case the electrodes in the original architecture become faulty [36]. These approaches are not suitable for mVLSI biochips.

Our fault-tolerant design strategy is part of the flow layer physical design step. Our algorithm takes as input (i) a *netlist* of components, i.e., the components in the architecture and their interconnections, (ii) an application model consisting of a sequencing graph, where each node is an operation and edges capture fluid dependencies, (iii) a fault model, and (iv) a set of constraints imposed by the designer, and produces as output a fault-tolerant netlist. We are interested in that fault-tolerant netlist, which fulfills the constraints imposed by the designer (e.g., in terms of maximum biochip area to be used for fault-tolerance) and corresponds to an architecture that is able to successfully run the biochemical application even in case of the occurrence of faults in the given fault model.

Fig. 7a presents an example input netlist, where we have two inputs, one output, one mixer, a storage components (consisting of 8 channels which can store fluids), one heater and



(a) Example architecture (netlist) (b) Fault-tolerance to the fault pattern in Fig. 7a (c) Fault-tolerant architecture

Fig. 7. Fault-tolerant architecture examples

one filter. The flow channel intersections are called "switches" and are denoted with S_i. The fault models used as input to our algorithm can be specified in several ways. If the designer has used the biochip extensively and has noticed a repeating fault pattern, such a fault pattern can be provided as input. For example, in Fig. 7a the designer has specified that she is interested to tolerate a stuck channel between switch S_1 and $Mixer_1$ (the channel is depicted with a thick red line) and a malfunctioning valve in the pump component of $Mixer_1$ (such pumping valves are used more extensively compared to the other valves, and hence are more likely to fail). Such a precisely given fault pattern represents a simple case for our algorithm, where the optimization focuses on introducing redundancy only for the specified faults. For example, Fig. 7b shows a possible architecture that would tolerate the faults from Fig. 7a. Thus, we have introduced a redundant channel (the thick green line), which can be used as an alternative if the channel S_1 and $Mixer_1$ fails, and we have used a fault-tolerant mixer, i.e., $FT\text{-}Mixer_1$. Such a fault tolerant mixer, see Fig. 2b for an example, uses a fourth valve (v_{13}) in the pump component of the mixer, which is normally composed on three valves. Thus, if one of the valves fails, the fault-tolerant mixer still has three functioning valves to perform the needed pumping action.

However, often, we do not know the exact fault pattern that has to be tolerated. Instead, the designer would specify a more general fault model. For example, for the architecture in Fig. 7a we assume that we do not know the actual fault pattern, and we are interested to tolerate any single channel blockage and any single valve malfunction, wherever they would happen. Note that this is an example; more than a single fault in channels or valves can be specified as input to our algorithm. The difficulty in determining a fault-tolerant architecture in this case, is that we do not know a-priori where the fault will actually occur. We know the faults only after we have tested the biochip, and not during the design phase, which is discussed here. Our fault-tolerant architecture has to be able to tolerate any single fault occurrence in a channel or a valve. A possible such fault-tolerant architecture is presented in Fig. 7c, where we have used two mixers (one is redundant), and we have used fault-tolerant versions for the storage, heater and filter components. A fault-tolerant storage simply contains redundant channels, in our case 9 channels instead of 8, needed to tolerate a channel failure. A fault-tolerant heater will contain an additional meandering channel sitting on top of the heated area (an off-chip metal plate placed under the chip). Similarly, a fault-tolerant filter contains an additional filtration channel. A redundant channel structure is used in-between the inputs and the rest of the components. Note, we assume that fluid routing can be done through components such as mixers and storage, but not through the heater and filter.

As mentioned, our algorithm takes as input also a graph of operations, which models the biochemical application; see [34] for details. Biochemical applications may have timing requirements, so we assume that the application has a deadline by which it must complete. As discussed in the mVLSI design flow, the application is compiled on a given biochip architecture in the "Application Mapping" box in Fig. 6b, such that the imposed deadline is satisfied. During the testing phase, we determine the faults, and these are given as input to the compilation task, which will have to ignore the faulty components in the architecture. In our fault-tolerant design strategy we are interested to derive that fault-tolerant architecture, which will allow our application to be successfully compiled on a faulty architecture in the presence of any faults. Thus, we propose a compilation-based evaluation approach for the evaluation of each fault-tolerant netlist visited during the design space exploration, which can determine if, given any possible fault pattern, we will be able to successfully run the application. The evaluation approach relies on two checks (1) a "k-connectivity test" [37], which checks if the netlist becomes disconnected if k channels are faulty (a disconnected architecture cannot run the application, since it cannot route the fluids) and (2) a "worst-case execution test", which checks if, considering the worst-case fault-occurrence scenario, the application meets its deadline. Note that these testes depend on the application model. In our example, let us assume that the application does not perform any mixing after the filtering and heating steps, so no redundancy is needed in the output channels of the filtering and heating components (as we have in-between the inputs and the mixing components).

Once such a fault-tolerant netlist is determined, it is given as an input to the physical synthesis tasks, so we can check if the resulted physical design satisfies the imposed input constraints. For example, it may happen that the total biochip area used in the flow layer is too large, or that the number of control pins needed to drive the fault-tolerant biochip goes over a specified threshold (biochips are often limited in the number of input

pressure sources that they can use). We then go back to the previous phases in the mVLSI design flow, and we iterate until a satisfying solution is obtained.

IV. Conclusion and Future Work

In this paper we have addressed continuous-flow mVLSI biochips, based on the manipulation of fluids through fabricated micro-channels, where the basic building block is a microvalve. Although they are a key enabling technology for several application areas, a potential roadblock in the deployment of microfluidic biochips is the lack of test techniques to screen defective devices before they are used for biochemical analysis. Prior work on fault detection and fault tolerance in biochips has been limited to digital ("droplet") microfluidics. Recent work has addressed the automated testing of mVLSI biochips, and this paper has reported on such a technique. We hope that more work will be done in this area in the future, to bring the same level of automation to the testing of mVLSI biochips, as the one taken now for granted in microelectronics. Future work is also needed for the fault-tolerant design of mVLSI biochips. During the physical design of the biochip layout, redundancy can be introduced for valves, channels and microfluidic units, increasing thus the yield. A long-term vision is that, during the operation of the biochips, recovery techniques can also be employed, e.g., re-executing erroneous operations, based on runtime error detection.

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. Haeberle, 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] 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.

[6] 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.

[7] H.-P. Chou, M. A. Unger, and S. R. Quake, "A Microfabricated Rotary Pump," Biomedical Microdevices, vol. 3, pp. 323-330, Dec 2001.

[8] C. L. Hansen, M. O. A. Sommer, and S. R. Quake, "Systematic investigation of protein phase behavior with a microfluidic formulator," *Proceedings of the National Academy of Sciences USA*, vol. 101(40), pp. 14 431–14 436, 2004.

[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] C. C. Lee, A. Elizarov, C. J. Shu, Y. S. Shin, and A. N. Dooley, "Multistep synthesis of a radiolabeled imaging probe using integrated microfluidics," *Science*, vol. 310, pp. 1793–1796, 2005.

[12] J. S. Marcus, W. F. Anderson, and S. R. Quake, "Microfluidic single-cell mrna isolation and analysis," *Analytical Chemistry*, vol. 78(9), pp. 3084–3089, 2006.

[13] 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.

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

[15] 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. 16 266–16 271, 2008.

[16] "Verinata Health," http://www.verinata.com/.

[17] C. Fang et al., "Integrated microfluidic and imaging platform for a kinase activity radioassay to analyze minute patient cancer samples," *Cancer Research*, vol. 70, no. 21, pp. 8299–8308, November 2010.

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

[19] P. C. Blainey and S. R. Quake, "Dissecting genomic diversity, one cell at a time," Nature Methods, vol. 11, pp. 349-349, Mar 2014.

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

[21] 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.

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

[23] T. Xu and K. Chakrabarty, "Fault Modeling and Functional Test Methods for Digital Microfluidic Biochips," *IEEE Trans. Biomedical Circuits and Systems*, vol. 3, no. 4, pp. 241–253, Aug. 2009.

[24] W. H. Grover, R. A. Mathies, "Monolithic Membrane Valves and Pumps", chapter in Lab-on-a-Chip Technology (Editors: Keith E. Herold and Avraham Rasooly), Caister Academic Press, 2009.

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

[26] 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.

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

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

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] W. H. Minhass, P. Pop, J. Madsen, and T. Ho, "Control Synthesis for the Flow-Based Microfluidic Large-Scale Integration Biochips," in *Proc. of the Asia and South Pacific Design Automation Conference*, pp. 205–212, 2013.

[36] M. Alistar, P. Pop, J. Madsen, "Application-specific fault-tolerant architecture synthesis for digital microfluidic biochips", in *Proc. of the Asia and South Pacific Design Automation Conference*, pp. 794-800, 2013.

[37] S. Even, "An algorithm for determining whether the connectivity of a graph is at least k," Computer Science Technical Reports, Cornell University, 1973.