Fault Localization in Programmable Microfluidic Devices

Alessandro Bernardini, Chunfeng Liu, Bing Li, Ulf Schlichtmann
Chair of Electronic Design Automation, Technical University of Munich, Germany

alessandro.bernardini@tum.de, chunfeng.liu@tum.de, b.li@tum.de, ulf.schlichtmann@tum.de

***Abstract*—Programmable Microfluidic Devices (PMDs) have revolutionized the traditional biochemical experiment flow. Test algorithms for PMDs have recently been proposed. Test patterns can be generated algorithmically. But an algorithm for fault localization once some faults have been identified is not yet available. When testing a PMD, once a test pattern fails it is unknown where the stuck valve is located. The stuck valve can be any one valve out of many valves forming the test pattern. In this paper, we propose an effective algorithm for the localization of stuck-at-0 faults and stuck-at-1 faults in a PMD. The stuck valve is localized either exactly or within a very small set of candidate valves. Once the locations of faulty valves are known, it becomes possible to continue to use the PMD by resynthesizing the application.**

I. Introduction and Related Work

Microfluidic biochips are miniaturized devices that allow the automated execution of biochemical experiments with the use of only very small quantities of reagents and with high execution efficiency [1].

Micro channels and valves form the basic components of a flow-based microfluidic biochip [1], [2], [3].

When PMDs are manufactured, defects may result in chips not functioning correctly [4] and thus test methods are needed [5], [4], [6], [7]. Dedicated methods for testing faults in PMDs have recently been proposed in [8], [2].

A stuck-at-0 valve is a valve that cannot open and a stuck at-1 valve is a valve that cannot close properly. In [8], [2] test patterns formed by simple paths for stuck-at-0 valve testing and cut-sets for stuck-at-1 valve testing are generated. If a test fails, then we know that some valve is stuck, but it is *not* possible to know which valve is stuck.

In this paper, we propose algorithms for the localization of stuck valves.

II. Motivation and Problem Formulation

Any defect that produces a valve that cannot open properly is called a *stuck-at-0 fault*. Any defect for which a valve cannot close properly is a *stuck-at-1 fault*. To test stuck-at-0 faults and stuck-at-1 faults, test patterns identified by *flow paths* and *cut-sets* on a biochip are generated respectively [8], [2]. For an explanation of the graph representation in Figure 1, see [2]. Testing is done by applying pressure from a pressure source and measuring pressure at a pressure meter as described in detail in [8], [2].

We construct a graph $G = (V, E)$ that describes the PMD. The set of nodes V contains all the channels and cells in the PMD and the set of edges E identifies the valves in the PMD. More precisely, for channels/cells $n_1, n_2 \in V$ we have that

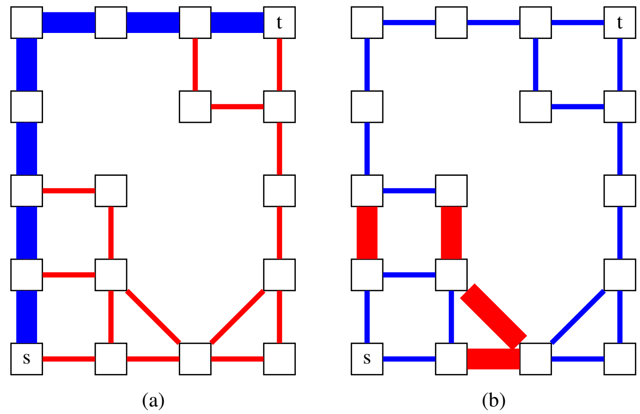


Fig. 1. A simple s-t path (a) (thick edges) and an s-t cut-set (b) (thick edges). Paths and cut-sets form test patterns for stuck-at-0 and stuck-at-1 faults respectively [2]. The color indicates the state of the valves for testing (blue means open and red closed).

$\{n_1, n_2\} \in E$ if and only if there is a valve that controls the *direct* flow between n_1 and n_2. The pressure source is represented by a node $s \in V$ and the sink with the pressure meter by a node $t \in V$. We will use henceforth the following *color coding*, which is important for the remainder of this paper: an edge in red represents a closed valve and an edge in blue represents an open valve. An orange edge represents a closed valve that *can* open and a cyan edge represents an open valve that *can* close. Thus orange and cyan represent valves that are *not* stuck. Stuck edges are labeled in the graph for more clarity. Figure 1 shows an example.

If some test [8], [2] fails, there is the need to further localize the stuck edges.

III. Definitions

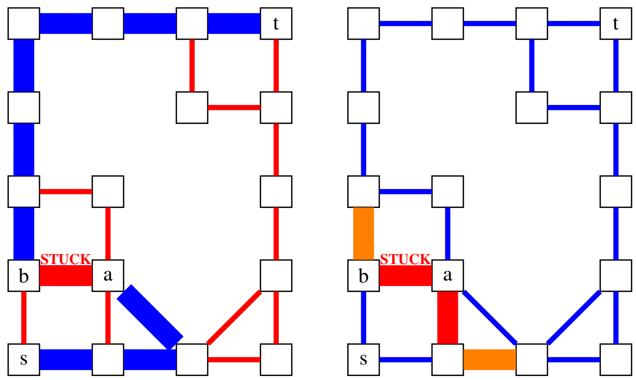
A simple path in the graph G is a path (and thus also a subgraph) with no cycles. We will identify a simple path from the pressure source s to the pressure sink t (a simple s-t path) with the set of *edges* p forming the path itself.

An s-t cut-set is formally a set of edges, such that when removing the edges from the graph (assumed connected), we obtain a partition into two disjoint blocks, such that the source s is in one block and the sink t is in the other block of the partition.

A simple s-t path forms a test pattern for testing stuck-at-0 faults in a PMD and an s-t cut-set forms a test pattern for testing stuck-at-1 faults in a PMD [8], [2].

A cut-set may not test all edges in the cut-set itself [2].

978-3-9819263-2-3/DATE19/



(a) (b)

Fig. 2. Localizing stuck-at-0 faults when a test fails for the physical PMD. (a) A simple s-t path p_f (thick edges) forming a test pattern for stuck-at-0 faults that fails. Some edge $\{a, b\}$ in the path is stuck-at-0. (b) An s-t cut-set $c_{\{a,b\}}$ (thick edges) that contains $\{a, b\}$.

Given two sets x and y, we will indicate the set difference (i.e. the relative complement) with $x - y$.

IV. Localization of Faults

A. *Localization of stuck-at-0 faults*

In the present section we describe Algorithm 1 for the localization of stuck-at-0 faults. At the end of the section an example will illustrate the algorithm.

Let us assume that we have a set $\mathcal{P}$ of simple s-t paths forming a complete set of test patterns for stuck-at-0 faults in the PMD and let us assume that we have a set $\mathcal{C}$ of s-t cut-sets forming a complete set of test patterns for stuck-at-1 faults in the PMD.

Let us assume that the test for the path $p_f \in \mathcal{P}$ fails. Some edge $\{a, b\}$ in the path p_f must be stuck-at-0 as shown in Figure 2(a). When the test fails it is unknown which edge along the path is actually stuck-at-0 and thus the edge $\{a, b\}$ needs to be localized.

For actually finding $\{a, b\}$ exactly or possibly within a small set of valves we proceed as described in Algorithm 1. Let $\mathcal{E}$ be the set of edges in the failing path that are known not to be stuck-at-0. We initialize $\mathcal{E}$ with the empty set.

We pick (line 12) an edge e_u in p_f for which it is not yet known if the edge is stuck-at-0 or not. Then (line 14) we consider an s-t cut-set $c_{e_u} \in \mathcal{C}$ that tests e_u for stuck-at-1 faults. This cut-set certainly exists because the set $\mathcal{C}$ is complete. The cut-set c_{e_u} will certainly contain e_u. For each edge e in c_{e_u} that is also in p_f (line 16), we close the valves in the cut set c_{e_u} and open the other valves checking that no pressure reaches the meter and then we open the edge e measuring the pressure at the sink t. If we detect pressure at the sink, we can affirm that it was certainly possible to successfully open the edge e. We thus proceed knowing that e is not stuck-at-0 and insert e in the set $\mathcal{E}$ in line 26. If, when opening the edge, a pressure is *not* detected at the sink we have localized a *candidate* for a stuck-at-0 edge. The edge is thus inserted in the set $\mathcal{S}_0$ containing candidate edges for stuck-at-0 faults (line 29).

Algorithm 1: Algorithm for localizing stuck-at-0 valves

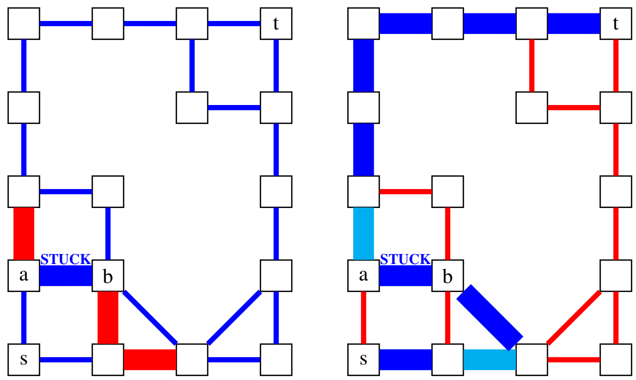
Input : physical PMD, described with graph $G = (V, E)$ with source s and sink t, complete set of test s-t paths $\mathcal{P}$ and complete set of test s-t cut-sets $\mathcal{C}$ with test for path $p_f \in \mathcal{P}$ failing.

Output: set $\mathcal{S}_0$ of candidate stuck-at-0 edges such that $\mathcal{S}_0 \supseteq$ set of edges in p_f that are stuck-at-0.

1 $\mathcal{S}_0 := \emptyset$//set of candidate stuck-at-0 edges
2 $\mathcal{E} := \emptyset$//set of edges that are known not to be stuck-at-0
3 **if** *consider_all_paths* **then**
4   **for** $p \in \mathcal{P}$ **do**
5   **if** *test for* p *is successful* **then**
6   //all edges in path p considered not stuck-at-0
7       $\mathcal{E} \leftarrow \mathcal{E} \cup (p \cap p_f)$
8   **end**
9   **end**
10 **end**
11 **while** $p_f - (\mathcal{E} \cup \mathcal{S}_0) \neq \emptyset$ **do**
12   $e_u \leftarrow$ pick edge in $p_f - (\mathcal{E} \cup \mathcal{S}_0)$
13   //it is not known if e_u is stuck-at-0
14   $c_{e_u} \leftarrow$ cut-set in $\mathcal{C}$ that tests e_u for stuck-at-1
15   //observe that $e_u \in p_f \cap c_{e_u}$
16   **for** $e \in p_f \cap c_{e_u}$ **do**
17   **if** $e \in \mathcal{E} \cup \mathcal{S}_0$ **then**
18   continue
19   **end**
20   open all valves in E
21   close all valves in c_{e_u}
22   check that pressure does not reach t from s (abort otherwise)
23   open e
24   **if** *pressure does reach* t **then**
25   //valve e necessarily opens properly.
26       $\mathcal{E} \leftarrow \mathcal{E} \cup \{e\}$
27   **else**
28   //valve e is stuck-at-0 or pressure otherwise blocked (false positive).
29       $\mathcal{S}_0 \leftarrow \mathcal{S}_0 \cup \{e\}$
30   **end**
31   **end**
32 **end**
33 **return** $\mathcal{S}_0$

As an example, Figure 2(b) shows the s-t cut-set $c_{\{a,b\}}$ that contains 3 edges in the path p_f that fails, path shown in Figure 2(a). All those 3 edges must be opened and closed and each time the pressure must be measured at the sink t. Opening a functioning valve will result in the detection of pressure at the sink. When opening $\{a, b\}$ no pressure will be detected at the sink t because the edge $\{a, b\}$ is stuck-at-0 and the edge $\{a, b\}$ is thus inserted in $\mathcal{S}_0$.

As an optional initialization step (lines 3-10) , we can iterate over all the test paths in the set $\mathcal{P}$. If for a path $p \in \mathcal{P}$ the test does *not* fail, we can assume that all edges in p are not stuck-at-0. Only in the presence of multiple faults of different types fault masking may occur in rare cases [8] with the possibility of false negatives, i.e. stuck-at-0 edges classified as not stuck-at-0 (we omit further details for compactness). All edges in $p \cap p_f$ are thus considered not stuck-at-0 and inserted into the



(a) (b)

Fig. 3. Localizing stuck-at-1 faults when a test fails for the physical PMD. (a) A simple s-t cut-set c_f (thick edges) forming a test pattern for stuck-at-1 faults that fails. Some edge $\{a, b\}$ in the cut-set is stuck-at-1. (b) An s-t simple path $p_{\{a,b\}}$ (thick edges) that contains $\{a, b\}$. Opening and closing valves of c_f in the shown path and measuring the pressure at t will lead to the localization of the stuck-at-1 edge $\{a, b\}$ because no absence of pressure is detected when closing the faulty valve.

set $\mathcal{E}$ in line 7. Omitting this optional initialization step will lead to a higher number of necessary pressure measurements but also the possibility of false negatives is avoided.

A final remark on line 22 where we check absence of pressure in t. Should pressure be detected in the check in line 22, then some valve in the cut-set c_{e_u} must be stuck-at-1 and we have the presence of multiple faults of different type in the PMD.

B. Localization of stuck-at-1 faults

An exact dual reasoning can be used for the localization of stuck-at-1 faults, as described by Algorithm 2.

Figure 3(b) shows a path $p_{\{a,b\}}$ from s to t containing the faulty edge $\{a, b\}$. Closing a functioning valve will result in the absence of pressure at the sink t. When closing $\{a, b\}$ pressure will still be detected at the sink t because the edge $\{a, b\}$ is stuck-at-1 and the edge $\{a, b\}$ is thus inserted in S_1.

C. False positives

We quickly discuss why false positive results are a possibility. Let us assume that the test for stuck-at-0 valves corresponding to the path in Figure 4(a) fails. For $e_u = \{a, c\}$ let c_{e_u} be the cut-set shown in Figure 4(b) (thick edges). Because the edge $\{a, b\}$ is stuck-at-0 (i.e. is closed, in red), then the edge $\{a, c\}$ in Figure 4(b) is not validly tested for stuck-at-1 faults. When opening $e = e_u = \{a, c\}$ for the situation depicted in Figure 4(b) a pressure would *not* reach the sink t because of the closed valve $\{a, b\}$. But then in line 29 of Algorithm 1 we would add the edge $\{a, c\}$ to the set S_0 of candidate stuck-at-0 edges.

Observing Figure 5(a) and Figure 5(b), we see that the edge $\{a, c\}$ will be classified as a candidate stuck-at-1 edge.

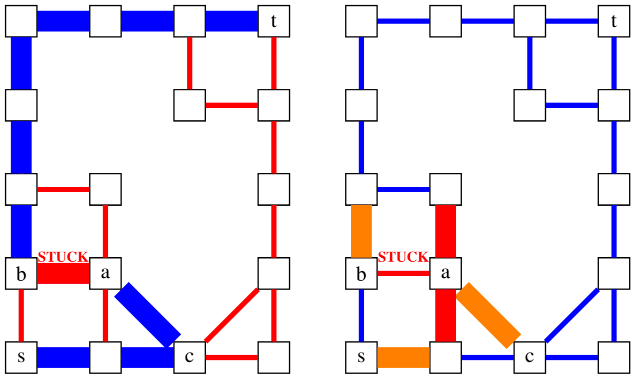
V. Simulation Results

We have written a simulator in C++ for simulating Algorithm 1 and Algorithm 2 running on a Intel(R) Xeon(R) Gold 6126 CPU @ 2.60GHz microprocessor. We use the PMDs as in [8]. Table I contains the results of the simulations for stuck-at-0 valves and Table II the results for stuck-at-1 valves without performing the optional initialization step. For each PMD we iterate over *all* edges simulating each time the case that the current edge is stuck-at-0 and separately the case that the current edge is stuck-at-1 and we simulate Algorithm 1 and Algorithm 2. We obtain a *vector* of sets $\mathcal{S}_0$ (for the stuck-at-0 fault type) and a *vector* of sets $\mathcal{S}_1$ (for the stuck-at-1 fault type). In addition we record for each fault type and each simulation the number of necessary pressure measurements to perform. Table I and Table II show the obtained mean values (μ), standard deviations (σ) and maximum values (max) for the cardinality of $\mathcal{S}_0$ and the cardinality of $\mathcal{S}_1$ respectively, together with the obtained mean values (μ) and standard deviations (σ) of the number of pressure measurements to perform. Finally Table III and Table IV show results whit the optional initialization, but false negatives in the presence of multiple faults of different type are here possible.

Algorithm 2: Algorithm for localizing stuck-at-1 valves

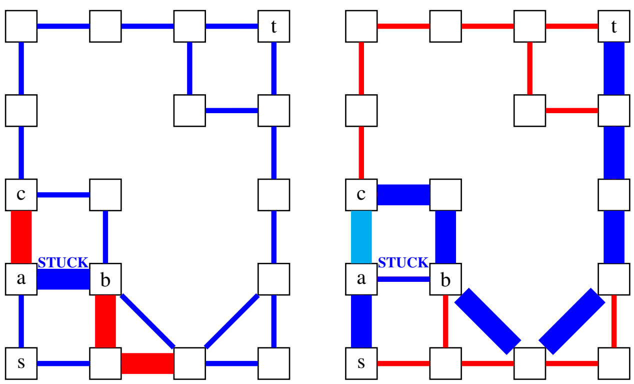
Input : physical PMD, described with graph $G = (V, E)$ with source s and sink t, complete set of test s-t paths $\mathcal{P}$ and complete set of test s-t cut-sets $\mathcal{C}$ with test for cut-set $c_f \in \mathcal{C}$ failing.
Output: set $\mathcal{S}_1$ of candidate stuck-at-1 edges such that $\mathcal{S}_1 \supseteq$ set of edges in c_f that are stuck-at-1.

1 $\mathcal{S}_1 := \emptyset$//set of candidate stuck-at-1 edges
2 $\mathcal{E} := \emptyset$//set of edges that are known not to be stuck-at-1
3 **if** *consider_all_cut-sets* **then**
4   **for** $c \in \mathcal{C}$ **do**
5   **if** *test for c is successful* **then**
6   //all edges in cut-set c considered not stuck-at-1
7             $\mathcal{E} \leftarrow \mathcal{E} \cup (c \cap c_f)$
8   **end**
9   **end**
10 **end**
11 **while** $c_f - (\mathcal{E} \cup \mathcal{S}_1) \neq \emptyset$ **do**
12     $e_u \leftarrow$ pick edge in $c_f - (\mathcal{E} \cup \mathcal{S}_1)$
13   //it is not known if e_u is stuck-at-1
14     $p_{e_u} \leftarrow$ path in $\mathcal{P}$ that tests e_u for stuck-at-0
15   //observe that $e_u \in c_f \cap p_{e_u}$
16   **for** $e \in c_f \cap p_{e_u}$ **do**
17   **if** $e \in \mathcal{E} \cup \mathcal{S}_1$ **then**
18   continue
19   **end**
20   close all valves in E
21   open all valves in p_{e_u}
22   check that pressure reaches t from s (abort otherwise)
23   close e
24   **if** *pressure does not reach t* **then**
25   //valve e necessarily closes properly.
26             $\mathcal{E} \leftarrow \mathcal{E} \cup \{e\}$
27   **else**
28   //valve e is stuck-at-1 or pressure otherwise reaching sink (false positive).
29             $\mathcal{S}_1 \leftarrow \mathcal{S}_1 \cup \{e\}$
30   **end**
31   **end**
32 **end**
33 **return** $\mathcal{S}_1$



(a) (b)

Fig. 4. (a) A failing path (thick edges). The edge $\{a, b\}$ is stuck-at-0. (b) For the edge $e_u = \{a, c\}$ we consider the shown cut-set for c_{e_u} (thick edges). When opening the edge $e = e_u = \{a, c\}$ the pressure cannot reach t because $\{a, b\}$ is closed and Algorithm 1 will classify the edge $\{a, c\}$ as a candidate stuck-at-0 edge in line 29. We obtain a false positive.



(a) (b)

Fig. 5. (a) A failing cut (thick edges). The edge $\{a, b\}$ is stuck-at-1. (b) For the edge $e_u = \{a, c\}$ we consider the shown path for p_{e_u} (thick edges). When closing the edge $e = e_u = \{a, c\}$ the pressure will still reach t because $\{a, b\}$ is open and Algorithm 2 will classify the edge $\{a, c\}$ as a candidate stuck-at-1 edge in line 29. We obtain a false positive.

PMD	$\|\mathcal{S}_0\|$			number of pressure measurements		
dimension	μ	σ	max	μ	σ	max
5×5	1.36	0.57	3	18.94	3.16	22
10×10	1.48	1.88	18	72.60	22.74	94
15×15	1.44	1.83	19	140.59	41.66	189
20×20	1.23	1.09	18	237.81	91.72	359
30×30	1.09	0.45	12	757.83	128.31	853

TABLE I
SIMULATION RESULTS STUCK-AT-0 (WITHOUT OPTIONAL STEP)

VI. Conclusion and Future Work

In this paper, we have provided algorithms for the localization of faults in programmable microfluidic devices (PMDs). To the best of our knowledge, this is the first solution to the localization problem. The provided algorithms can be used to obtain a set of valves that contains the stuck valve that causes the test to fail. Thus the stuck valve can be localized as one of the valves in the obtained set. On typical PMDs a single faulty valve can on average be localized within one or two valves. After the localization, it becomes possible to continue to use the PMD. Simulations confirm correctness.

PMD	$\|\mathcal{S}_1\|$			number of pressure measurements		
dimension	μ	σ	max	μ	σ	max
5×5	1.78	0.86	4	6.44	2.15	8
10×10	1.63	0.50	3	12.02	4.76	18
15×15	2.06	1.23	8	21.10	8.73	38
20×20	2.10	1.29	9	26.67	9.49	38
30×30	2.13	1.08	11	39.63	13.84	78

TABLE II
SIMULATION RESULTS STUCK-AT-1 (WITHOUT OPTIONAL STEP)

PMD	$\|\mathcal{S}_0\|$			number of pressure measurements		
dimension	μ	σ	max	μ	σ	max
5×5	1.13	0.33	2	7.81	4.18	22
10×10	1.23	1.02	10	31.26	13.25	51
15×15	1.17	0.76	6	35.49	23.54	140
20×20	1.42	0.73	5	73.24	45.61	336
30×30	1.02	0.37	11	174.62	92.26	547

TABLE III
SIMULATION RESULTS STUCK-AT-0 (WITH OPTIONAL STEP)

PMD	$\|\mathcal{S}_1\|$			number of pressure measurements		
dimension	μ	σ	max	μ	σ	max
5×5	1.55	0.63	3	4.50	1.98	7
10×10	1.59	0.49	2	12.02	4.76	18
15×15	1.79	0.98	7	15.03	6.66	24
20×20	1.92	1.16	9	21.44	9.71	36
30×30	2.01	0.99	8	33.96	13.22	58

TABLE IV
SIMULATION RESULTS STUCK-AT-1 (WITH OPTIONAL STEP)

References

[1] J. Melin and S. Quake. Microfluidic large-scale integration: the evolution of design rules for biological automation. *Annu. Rev. Biophys. Biomol. Struct.*, 36:213–231, 2007.

[2] Alessandro Bernardini, Chunfeng Liu, Bing Li, and Ulf Schlichtmann. Efficient spanning-tree-based test pattern generation for programmable microfluidic devices. *Microelectronics Journal*, 79:38–45, 2018.

[3] L. M. Fidalgo and S. J. Maerkl. A software-programmable microfluidic device for automated biology. *Lab Chip*, 11:1612–1619, 2011.

[4] Kai Hu, Feiqiao Yu, Tsung-Yi Ho, and Krishnendu Chakrabarty. Testing of flow-based microfluidic biochips: Fault modeling, test generation, and experimental demonstration. *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, 33(10):1463–1475, 2014.

[5] Kai Hu, Tsung-Yi Ho, and Krishnendu Chakrabarty. Testing of flow-based microfluidic biochips. In *Proc. VLSI Test Symp. (VTS)*, pages 1–6, 2013.

[6] Kai Hu, Tsung-Yi Ho, and Krishnendu Chakrabarty. Test generation and design-for-testability for flow-based mVLSI microfluidic biochips. In *Proc. VLSI Test Symp. (VTS)*, pages 97–102, 2014.

[7] Kai Hu, Bhargab B. Bhattacharya, and Krishnendu Chakrabarty. Fault diagnosis for flow-based microfluidic biochips. In *Proc. VLSI Test Symp. (VTS)*, pages 1–6, 2015.

[8] Chunfeng Liu, Bing Li, Bhargab B Bhattacharya, Krishnendu Chakrabarty, Tsung-Yi Ho, and Ulf Schlichtmann. Testing microfluidic fully programmable valve arrays (fpvas). In *2017 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 91–96. IEEE, 2017.