Integrated Test Module Design for Microfluidic Large-Scale Integration

Mengchu Li, *Student Member, IEEE*, Yushen Zhang *Student Member, IEEE*, Ju Young Lee, Hudson Gasvoda, Ismail Emre Araci, Tsun-Ming Tseng, *Member, IEEE*, Ulf Schlichtmann, *Senior Member, IEEE*

***Abstract*—Microfluidic large-scale integration (mLSI) is a promising lab-on-a-chip platform for high-throughput bio-applications. Due to the high integration scale and the small feature size, control channels on mLSI chips are prone to blockage and leakage defects, which may lead to faulty behavior of valves and erroneous experimental results. Thus, mLSI chips need to be tested before usage. Current mLSI-tests are mostly performed in a straightforward way by testing each valve individually, which is very time-consuming and error-prone. As the integration scale of mLSI chips keeps increasing, there is a pressing demand for more efficient test approaches. This work proposes the first built-in-self-test (BIST) method for mLSI with an integrated test module design. Instead of testing individual valves, the proposed method directly tests the control channels and thus greatly improves the test efficiency. Only $\frac{n}{2}$ and $\lceil \log_2(n+1) \rceil$ test operations are required to test the blockage and leakage defects, respectively, of n control channels. The proposed test module consumes moderate area overhead and the test method is easy to operate. Neither specialized software nor external pressure sensors are required for carrying out the tests. Experiments show that our test approach is sensitive enough to detect defects that have a feature size as small as $10\mu m$ and that are several centimeters away from the test module.**

***Index Terms*—Testing, Microfluidic large-Scale integration, Continuous-flow microfluidics, mLSI, BIST, Design for testability.**

I. Introduction

Microfluidic large-scale integration (mLSI) is an emerging platform for high-throughput chemical and biological applications such as protein analysis [1], gene sequencing [2] and synthetic biology [3]. Compared to other lab-on-a-chip platforms, mLSI stands out by integrating a large number of elastomeric valves into a miniaturized chip, which allows various fluid manipulation operations to be performed with different reagents at nanoliter scale in parallel in a sensitive and precise manner [4].

An mLSI chip consists of two layers: a flow layer for fluid transportation and a control layer for pressure transportation. A valve is formed with channel segments from both layers and a membrane in-between. Pressurizing and depressurizing the control channel segment of a valve will lead to a shape change of the membrane, which will thus seal and unseal the corresponding flow channel segment, respectively.

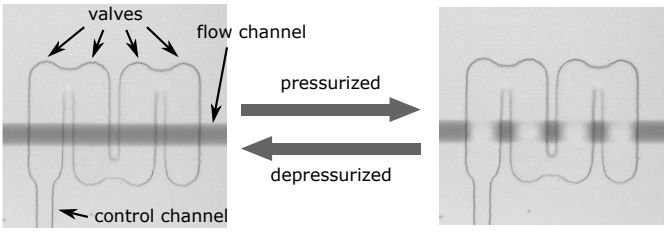


Figure 1: Pressurizing valves blocks the fluid movement.

With channels and valves, complex microfluidic components such as chambers, mixers and fluid switching elements can be constructed [5]. For example, Figure 1 shows a peristaltic pump consisting of four valves which are sequentially connected by a single control channel[1]. When the control channel is pressurized, pressure will propagate to each of the four valves one by one, pushing the fluid in the corresponding flow channel from left to right. Such pumps are important components of rotary mixers [6]. By rapidly pressurizing and depressurizing the control channel, a circulation flow can be generated inside the mixer which enables efficient mixing operations.

Compared to flow channels which have a typical width of $100\mu m$, the width of a control channel (excluding valve segments) is usually less than $30\mu m$ to avoid forming valves at unwanted locations [7]. Due to the high integration scale and the small feature size, control channels are prone to defects. Commonly-seen control channel defects can be modeled as *blockage* and *leakage* [8]:

- Blockage refers to the situation that a control channel is broken or blocked, and thus pressure cannot be transported to related valves properly, i.e. the valves cannot be pressurized.
- Leakage refers to the situation that two control channels connect improperly such that pressure in one control channel leaks to the other channel, i.e. valves in the other channel will be pressurized unintentionally.

To avoid erroneous experimental results and waste of precious samples, biologists/chemists need to test their mLSI chips before executing the assays. Currently, mLSI tests are

This work was supported by German Research Foundation (DFG) under Grant 456006534.

Mengchu Li, Yushen Zhang, Tsun-Ming Tseng and Ulf Schlichtmann are with the Chair of Electronic Design Automation, Technical University of Munich (TUM), Arcisstr. 21, Munich 80333, Germany (e-mail: mengchu.li@tum.de; yushen.zhang@tum.de; tsun-ming.tseng@tum.de; ulf.schlichtmann@tum.de).

Ju Young Lee, Hudson Gasvoda, and Ismail Emre Araci are with the Department of bioengineering, Santa Clara University 500 El Camino Real, Santa Clara, CA 95053, USA (e-mail: jlee16@scu.edu; hgasvoda@scu.edu; iaraci@scu.edu).

[1]Alternatively, valves in a peristaltic pump can be connected to multiple control channels so that each valve is controlled individually. In this case, the corresponding control channels need to be sequentially (de)pressurized.

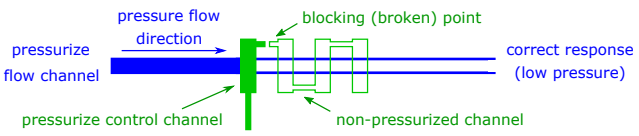


Figure 2: A blockage defect that cannot be detected with the functional test method. The blockage occurs between the first and the second valves of a peristaltic pump such that only the first valve can properly be pressurized, making the pump unable to work. Despite that, the pressure status at the right end of the flow channel still meets the expectation, i.e. the faulty behavior of the three valves on the right is masked by the correct behavior of the leftmost valve.

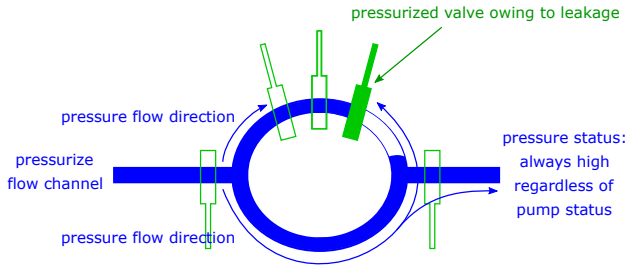


Figure 3: A leakage defect that cannot be detected with the functional test method. The leakage occurs in one of the control channels connected to a peristaltic pump, leading the rightmost valve in the pump to be pressurized unexpectedly. Thus, when the pump is actuated, fluid flow will stop at the rightmost valve and cannot form the required circulation for efficient mixing. However, pressure can still propagate through the bottom ring to the right end of the flow channel and result in correct pressure status, i.e. the faulty behavior of the valve is masked by the flow-layer structure.

usually performed by manually inspecting the valve actuation under the microscope. In order to obtain enough resolution to judge the valve behavior, only a small chip area can be inspected each time, which makes it very difficult to detect the leakage defects. As the integration scale of mLSI chips keeps increasing, the time-consuming and error-prone manual tests can hardly satisfy the demand. Thus, it is necessary to develop advanced mLSI test approaches.

So far, few mLSI test approaches have been proposed. Specifically, the functional test approaches proposed by Hu et al. [9], [10] represent the state of the art. These approaches connect the fluid inlets and outlets of a to-be-tested chip to external pressure sources and pressure sensors, respectively. A commercial automatic test pattern generation (ATPG) tool is then applied to generate test patterns for pressurizing the flow channels. For each test pattern, the ATPG tool calculates a set of expected pressure states at the outlets. If the results from the sensors do not match the calculation, one can conclude that there are some channel defects.

There are three major drawbacks of this functional test approach:

- First, the functional test approach measures control channel defects indirectly via the flow channels, which may lead to some defects being masked, such as the blockage and the leakage defects shown in Figure 2 and Figure 3, respectively.
- Second, the test results can only be gathered from fluid in-/outlets, the number of which is usually limited on current mLSI chips, which further leads to a limited number of available test patterns and thus limited fault-coverage.
- Third, the functional test method requires extra equipment (pressure sources and sensors) and a commercial ATPG tool to generate the test patterns, which implies additional costs and puts high demands on the test skills of the operator. Thus, it is hard for the end-users of mLSI chips, who do not necessarily have expertise in engineering, to carry out the test operations.

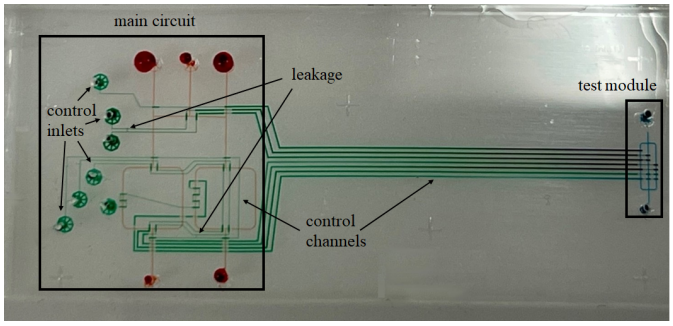


Figure 4: The photo of an mLSI chip integrated with a test module. Control channels, flow channels in the main circuit and flow channels in the test module are filled with green, red and blue food dye, respectively. In particular, the width and spacing of control channels between the main circuit and the test module are enlarged to make these channels more robust against defects.

Targeting the present deficiencies, this work proposes an integrated test module design for mLSI, which allows to directly test the control channels. Since the number of control channels is usually much smaller than the number of valves, the number of test operations is greatly reduced compared to inspecting valves. Besides, the area overhead of the test module is moderate. Compared to the functional test approach, the proposed method does not depend on the flow layer structure, and thus prevents the blockage and leakage defects from being masked. The test protocols are simple, and the test patterns can be derived intuitively. Neither specialized software nor external pressure sensors are required for carrying out the tests.

II. Prototype Design

In this work, we propose to integrate test modules onto mLSI chips to perform mLSI tests. We refer to mLSI designs integrated with test modules as *testable designs*.

A testable design consists of two parts: a *main circuit* for assay execution, and a *test module* with an independent flow circuit for testing. Figure 4 shows the photo of a testable mLSI chip, the main circuit of which contains two rotary

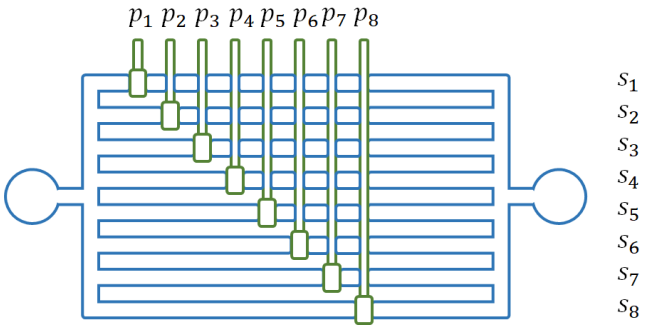


Figure 5: A prototype design of the test module.

mixers for efficient mixing of fluids[2]. In particular, there are two leakage defects between the control channels in the main circuit. The key idea of our test method is to let pressure propagate from control channels in the main circuit to valves in the test module, so that control channel defects in the main circuit can be reflected by faulty behavior of valves in the test module.

Figure 5 shows a prototype design of the test module for testing 8 control channels. As shown in the figure, this module consists of 8 flow channels aligned in parallel, each of which is integrated with a valve connected to a to-be-tested control channel. A test operation consists of three steps:

1) Pressurize the control channels following a test pattern. We denote a test pattern p as a sequence of binary digits, each of which indicates the pressure status in a to-be-tested control channel. Specifically, 0 represents *not pressurized* and 1 represents *pressurized*. We refer to the i$^{\text{th}}$ bit in p as p_i. For example, $p=00000001$ is a test pattern that pressurizes the 8$^{\text{th}}$ control channel, i.e. $p_1=0$, $\cdots$, $p_7=0$, $p_8=1$.
2) Send fluids such as food dye from a fluid inlet through the test module.
3) Inspect the fluid status in the test module under microscope. The inspection result is recorded as a sequence s of binary numbers, each of which indicates the fluid status in a flow channel of the test module. Specifically, 0 represents that fluids are blocked by the valve in the corresponding channel and 1 represents that fluids pass through the valve in the corresponding channel. We refer to the i$^{\text{th}}$ element in s as s_i. Using this prototype, the faultless inspection result for a test pattern p is always $\bar{p}$. For example, for $p=00000001$, we expect s to be $\bar{0}\bar{0}\bar{0}\bar{0}\bar{0}\bar{0}\bar{0}\bar{1}=11111110$, i.e. fluids pass through valves in all flow channels except for the bottom one.

Test for blockage

Blockage in a control channel i can be detected if the channel is pressurized but fails to block the fluids, i.e. $p_i=1$ but $s_i=1\neq\bar{1}$. Blockage in all control channels can be tested using this prototype with only one operation.

[2]To note is that the control channels between the main circuit and the test module can be shortened. In this design, we intentionally prolonged the control channels to demonstrate that the testability is not influenced by the channel length.

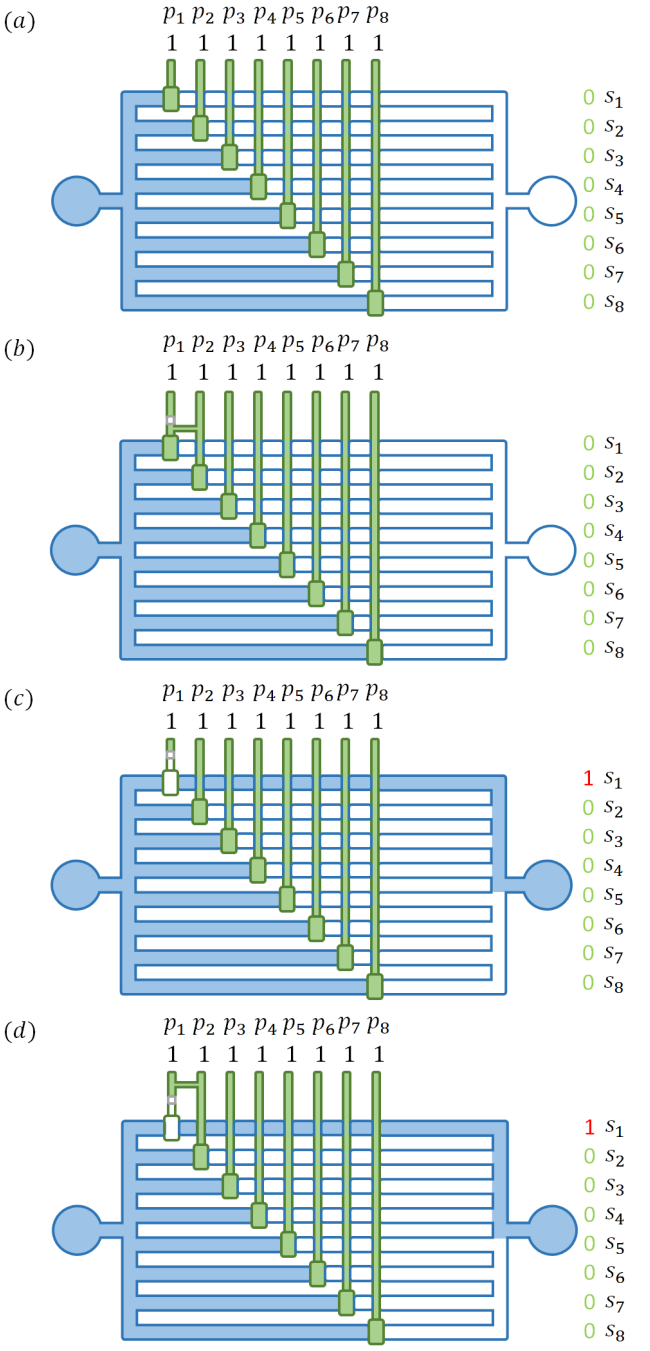


Figure 6: (a) The faultless result of the test for blockage. (b) If a blockage happens prior to a leakage, the leakage dominates. Thus, the defects will be detected in the leakage tests. (c) A blockage alone will result in faulty inspection results. (d) A blockage that happens posterior to a leakage will result in faulty inspection results.

- Test pattern: $p=11111111$.
- If the result is faultless, either there is no blockage, as shown in Figure 6(a), or the blockage exists prior to a leakage in the same channel, as shown in Figure 6(b).
- If $s_i=1\neq\bar{p}_i$ for any $1\leq i\leq 8$, there is blockage in the i$^{\text{th}}$ control channel, as shown in Figure 6(c) and (d).

Test for leakage

Leakage between control channels i and j can be detected if only one of the two channels is pressurized but both channels block the fluids, i.e. $p_i \neq p_j$ but $s_i = s_j = 0$. In other words, to detect leakage between any two control channels i and j, there must be a test pattern p so that $p_i \neq p_j$.

To this end, we build a matrix where each column is the binary representation of a control channel, and each row is a test pattern. Thus, the matrix for testing n control channels is of size $\lceil \log_2(n) \rceil \times n$. For example, to test leakage of 8 control channels using the prototype, we build a 3×8 matrix:

- Test patterns:

	1	2	3	4	5	6	7	8
p^1:	0	0	0	0	1	1	1	1
p^2:	0	0	1	1	0	0	1	1
p^3:	0	1	0	1	0	1	0	1

,

where the first channel is represented as 000, as shown in the first column, and the second channel is represented as 001, etc. Correspondingly, the first test pattern is 00001111, which can be read from the first row, and the second test pattern is 00110011, etc. We denote the pressure status of the i$^{\text{th}}$ channel in the k$^{\text{th}}$ test pattern as p_i^k, which represents the entry in the i$^{\text{th}}$ column and in the k$^{\text{th}}$ row. Since each column represents a unique binary number, we can ensure that any two channels $i \neq j$ differ in at least one bit. Thus, there is at least one test pattern p^k where the i$^{\text{th}}$ and the j$^{\text{th}}$ bits have different binary values, i.e. $p_i^k \neq p_j^k$.

Proof by contradiction: assume that $\forall k$: $p_i^k = p_j^k \Rightarrow p_i^1 p_i^2 p_i^3 \equiv p_j^1 p_j^2 p_j^3 \Rightarrow$ *channel* i *and channel* j *have the same binary representation* $\Rightarrow i = j \Rightarrow$ *Contradiction.*

Thus, leakage between any two of n control channels can be tested with $\lceil \log_2(n) \rceil$ test operations.

- If the results of all test operations are faultless, either there is no leakage, as shown in Figure 7(a), or the leakage exists prior to a blockage in the same channel, as shown in Figure 7(b).
- If for test pattern p^k, there exists $s_i^k = 0 \neq \bar{0}$ for any $p_i^k = 0$, there is a leakage between the i$^{\text{th}}$ control channel and other channels, as shown in Figure 7(c) and (d).

Analysis of the prototype design

- Fault coverage: full coverage of the blockage and leakage defects in the main circuit. In case that there are multiple defects in the same control channel, the defects will lead to faulty inspection results in either the blockage or the leakage tests.
- Efficiency for testing n control channels:

	for blockage	for leakage
Number of operations	1	$\lceil \log_2(n) \rceil$

- Resource overhead for testing n control channels:

	inlets	flow channels
Number of components	2	n

III. Enhanced Design to Reduce Resource Overhead

The prototype design of the test module allows for extremely efficient mLSI tests. However, since n flow channels are needed for testing n control channels, the prototype design can occupy much chip area. To reduce resource overhead, we enhance the design so that only $\lceil \log_2(n+1) \rceil$ flow channels are needed for testing n control channels.

To encode the test results with a reduced number of flow channels, we connect different control channels to valves in different flow channels in the test module, so that for any two

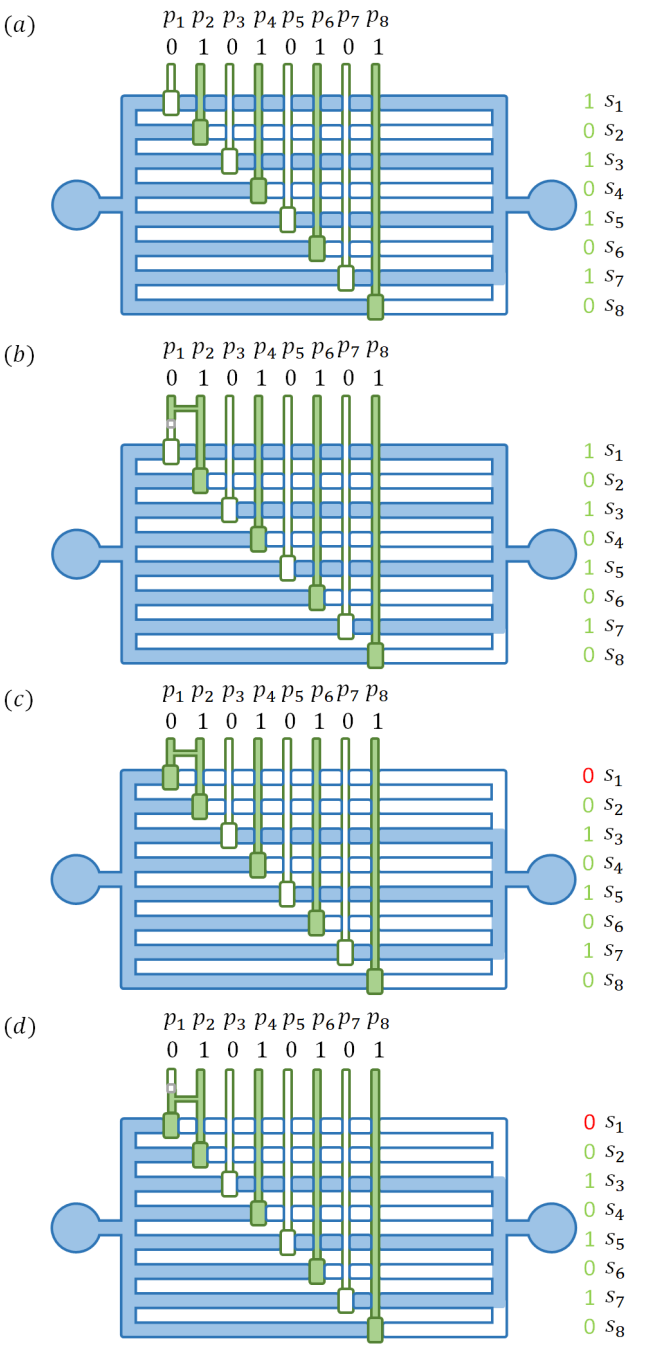


Figure 7: (a) The third test pattern and the faultless result for testing leakage defects. (b) If a leakage happens prior to a blockage, the blockage dominates. Thus, the defects will be detected in the blockage test. (c) A leakage alone will result in faulty inspection result. (d) A leakage that happens posterior to a blockage will result in faulty inspection results.

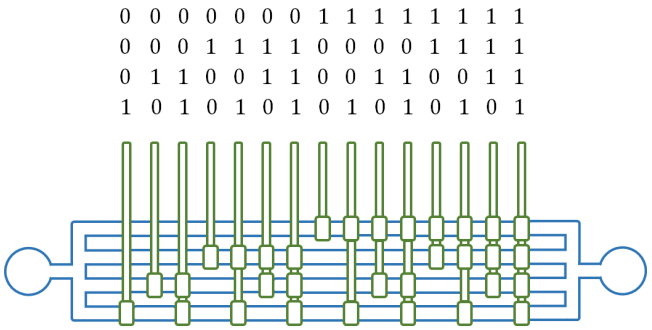


Figure 8: The enhanced design of the test module.

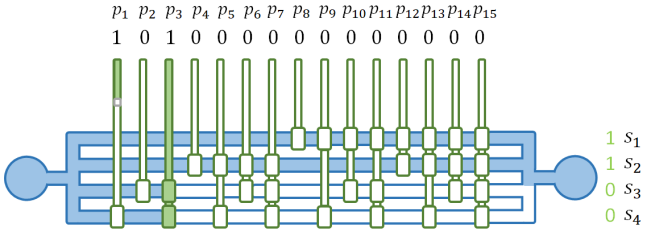


Figure 9: The expected faultless result of pressurizing the first (i.e. 0001) and the third (i.e. 0011) control channels is $s=1100$, which can be achieved as long as there is no blockage in channel 0010. Blockage in channel 0001 is thus masked.

different control channels i and j, pressurizing i will block a different set of flow channels than pressurizing j.

Figure 8 shows the enhanced design of the test module consisting of 4 flow channels, which can be used to test at most 15 (i.e. 2^4-1) control channels. Specifically, we represent the index of a to-be-tested control channel as a 4-bit binary expression in the range of 0001 to 1111. For example, the 5$^{\text{th}}$ control channel is indexed by 0101. For each control channel, if the k$^{\text{th}}$ bit of its index is 1, we connect it to a valve on the k$^{\text{th}}$ flow channel in the test module. For example, control channel 0101 is connected to valves on the second and the fourth flow channels.

Test for blockage

In blockage tests, we need to carefully select test patterns to prevent blockage defects from being masked. For example, as shown in Figure 9, if we pressurize the first (i.e. 0001) and the third (i.e. 0011) control channels together in one test operation, as long as there is no blockage in the third channel, the test results will always be correct, regardless of whether there is any blockage in the first control channel or not. In other words, blockage in the first control channel will be masked by the correct behavior of the third control channel.

Specifically, blockage in a control channel i may be masked by the correct behavior of another control channel j, if every bit in the binary expression of i is smaller than every bit in the binary expression of j.

- Test pattern: to ensure full fault coverage, we pressurize two control channels with sequential indices i and $i+1$ in each test operation, where i is an odd number. Since i is odd, the least significant bit of i must be 1 and the least significant bit of $i+1$ must be 0. Since $1>0$, defects in control channel i will not be masked by the correct behavior of control channel $i+1$. On the other hand, since $i+1>i$, there must be a bit in the binary expression of $i+1$ that is larger than the bit in the binary expression of i. Thus, defects in control channel $i+1$ will not be masked by the correct behavior of control channel i.
- For a test operation that tests control channels i and j, the faultless inspection result is the bitwise NOR of i and j. E.g., the faultless result for testing control channels 1101 and 1110 is $(\overline{1\vee1})(\overline{1\vee1})(\overline{0\vee1})(\overline{1\vee0})$, i.e. 0000, as shown in Figure 10(a). If the result is faultless, either there is no blockage or the blockage is dominated by a leakage. Otherwise, if the k$^{\text{th}}$ bit of the test result is faulty, we can derive that the blockage occurs in the control channel with the k$^{\text{th}}$ bit of the index equal to 1, as shown in Figure 10(b) and (c).

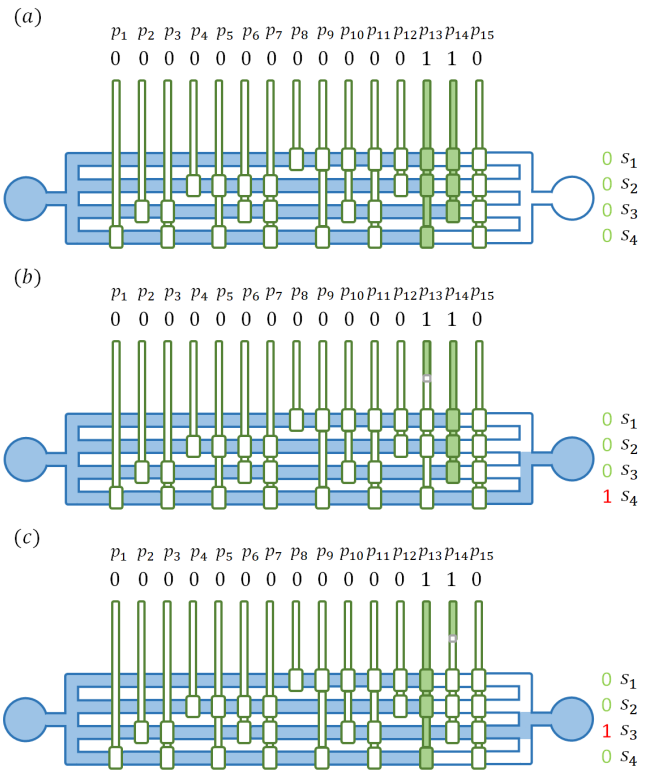


Figure 10: (a) Test pattern and the faultless result for testing blockage defects in the 13$^{\text{th}}$ (1101) and the 14$^{\text{th}}$ (1110) control channels. (b) Blockage in control channel 1101 results in faulty inspection results in the 4$^{\text{th}}$ flow channel. (c) Blockage in control channel 1110 results in faulty inspection results in the 3$^{\text{rd}}$ flow channel.

Test for leakage

Test for leakage with the enhanced test module design is similar to test for leakage with the prototype design.

- Test pattern: we need to ensure that for any two control channels i and j, there exists a test pattern p where i and j have different pressure status, i.e. $p_i \neq p_j$. To this end,

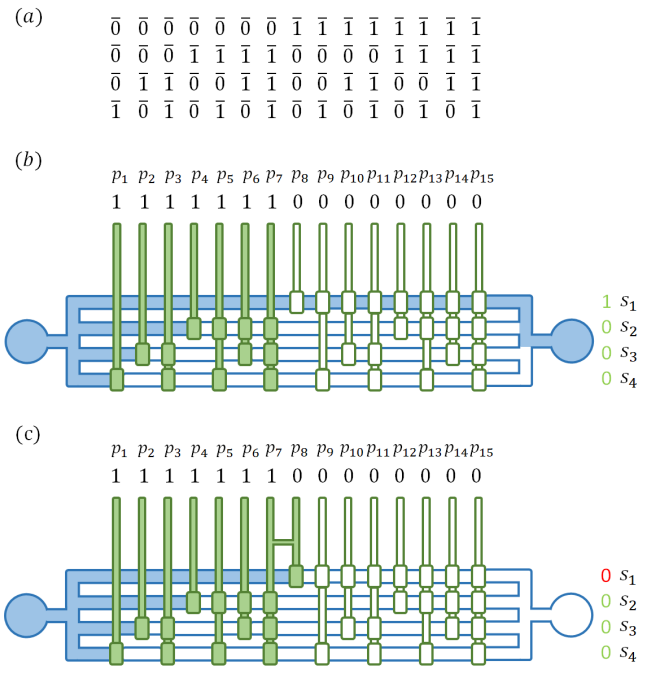


Figure 11: (a) The matrix that encodes the test pattern. (b) The first test pattern and the faultless results. (c) Leakage between any two control channels in different groups results in faulty inspection results in the first flow channel.

we build a matrix where each column is the complement of the binary representation of a control channel index, and each row is a test pattern. Thus, the matrix for testing n control channels is of size $\lceil \log_2(n+1) \rceil \times n$.

For example, to test leakage between 15 control channels using the design shown in Figure 8, we build a 4×15 matrix, as shown in Figure 11(a), where the first column is $\bar{0}\bar{0}\bar{0}\bar{1}$, i.e., the complement of channel index 0001; the second column is $\bar{0}\bar{0}\bar{1}\bar{0}$, etc. Similarly, the first test pattern is $\bar{0}\bar{0}\bar{0}\bar{0}\bar{0}\bar{0}\bar{1}\bar{1}\bar{1}\bar{1}\bar{1}\bar{1}\bar{1}\bar{1}$, as can be read from the first row, the second test pattern is $\bar{0}\bar{0}\bar{0}\bar{1}\bar{1}\bar{1}\bar{1}\bar{0}\bar{0}\bar{0}\bar{0}\bar{1}\bar{1}\bar{1}\bar{1}$, etc.

As each column of the matrix is a unique binary representation, any two columns i and j differ in at least one bit, i.e. there is at least one row (test pattern) where the i[th] and the j[th] columns (control channels) have different binary entries (pressure status). By complementing the binary representations, we ensure that in the k[th] test pattern, control channels that **are not** connected to valves in the k[th] flow channel are pressurized; and control channels that **are** connected to valves in the k[th] flow channel are not pressurized. Thus, fluids can pass through the k[th] flow channel by default, but if a leakage exists between the two groups of control channels that **are** and that **are not** connected to valves in the k[th] flow channel, the k[th] flow channel will be blocked.

For example, the first test pattern tests the leakage between the first seven channels (group of channels that **are not** connected to valves in the first flow channel) and the last eight channels (group of channels that **are** connected to valves in the first flow channel). If there is no leakage between channels from different groups, fluids can pass through the first flow channel, i.e. $s_1 = 1$, as shown in Figure 11(b). But if there is a leakage between any two channels from different groups, the first flow channel will be blocked, i.e. $s_1 = 0$, as shown in Figure 11(c).

- The faultless result for the k[th] test pattern is that the k[th] flow channel is not blocked, i.e. $s_k = 1$. If the inspection result is faultless, either there is no leakage between the two tested groups of control channels, or the leakage is dominated by a blockage. Otherwise, there is a leakage.
- If we do not only want to detect the leakage but also want to locate the pair of leaky control channels, we need at most another $\lfloor \frac{n}{2} \rfloor$ test operations. Specifically, if the result of the k[th] test pattern is faulty, we can derive that the indices of the two leaky control channels differ in the k[th] bit. Otherwise, the k[th] bits of the indices of the two leaky control channels are the same. Thus, with the results of the leakage tests, as long as we know the index of one leaky control channel, we can derive the index of the other leaky control channel. Therefore, to locate the leakage between two groups of control channels, we can arbitrarily select a group and pressurize the control channels in that group one by one.

  For example, if the results of all except the first test pattern are faultless, we can derive that there is a leakage between the first seven and the last eight control channels and the indices of the two leaky control channels differ only in the first bit. Thus, we can locate the leakage by pressurizing the first seven channels one by one. If pressurizing channel 0001 leads to a faulty result, we can identify that the other leaky channel is 1001 (i.e. $\bar{0}001$).

Analysis of the enhanced design

- Fault coverage: full coverage of the blockage and leakage defects in the main circuit. In case that there are multiple defects in the same control channel, the defects will lead to faulty inspection results either in the blockage tests or in the leakage tests.
- Efficiency for testing n control channels:

	for blockage	for leakage
Number of operations	$\lceil \frac{n}{2} \rceil$	$\lceil \log_2(n+1) \rceil$

- Resource overhead for testing n control channels:

	inlets	flow channels
Number of components	2	$\lceil \log_2(n+1) \rceil$

- Comparison: compared to the prototype design, the enhanced design significantly reduces the resource overhead, i.e. the number of flow channels needed for testing n control channels is reduced from n to $\lceil \log_2(n+1) \rceil$. Besides, the reduced flow channel requirement is important from the imaging point of view. When the number of to-be-tested control channels is not extremely large, it is possible to inspect all flow channels in the test module in one field of view. As a trade-off, the number of test operations for blockage increases from 1 to $\lceil \frac{n}{2} \rceil$. Test for leakage can still be performed very efficiently with $\lceil \log_2(n+1) \rceil$ operations.

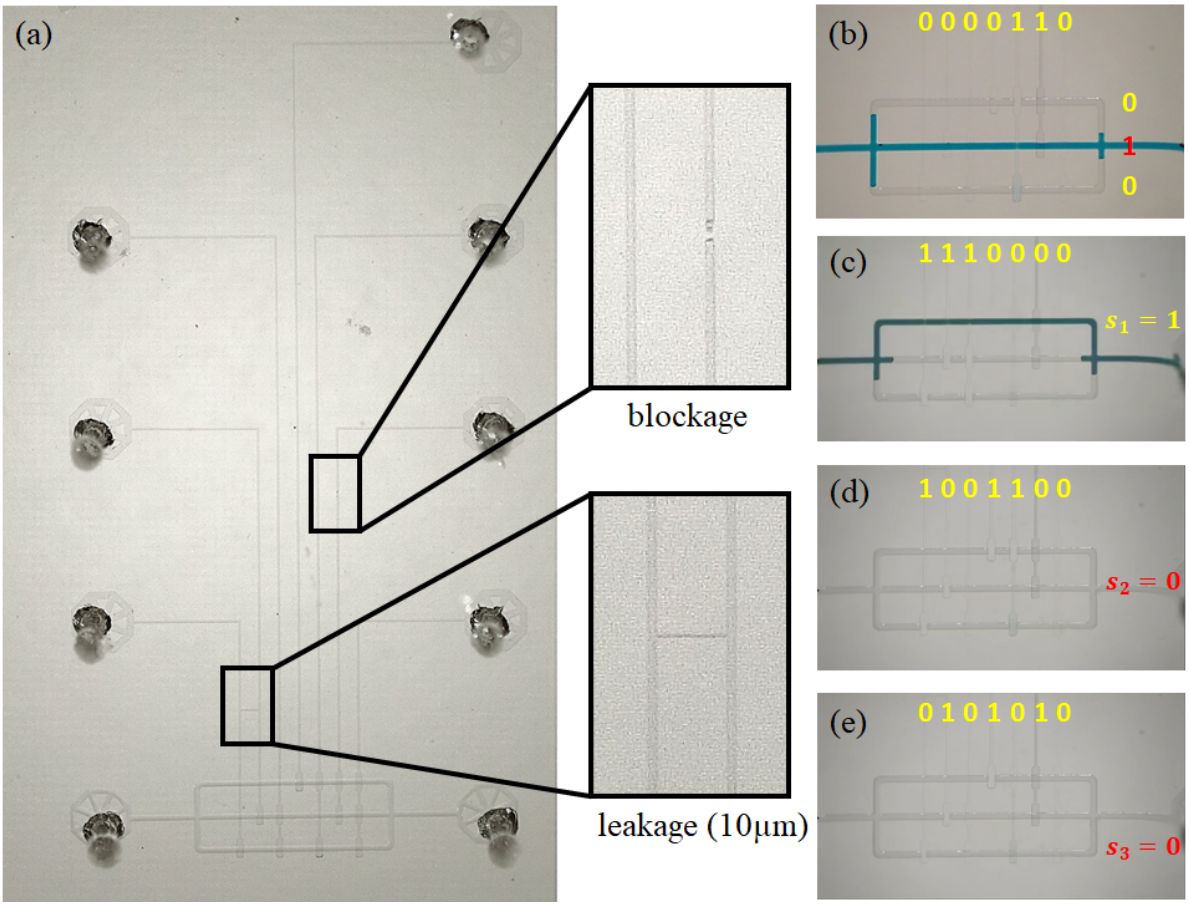


Figure 12: (a) An mLSI chip with a blockage defect and a $10\mu m$ leakage defect. (b) The result (010) of the 3rd test pattern (0000110) in blockage tests is faulty (000 is the faultless result). Based on the faulty result, we know that either the 5^{th} or the 6^{th} control channel is blocked and the 2^{nd} bit of the binary index of the blocked channel is 1, i.e., the blockage is in the 6^{th} control channel. (c) The result ($s_1 = 1$) of the 1st test pattern (1110000) in leakage tests is faultless. (d) The result ($s_2 = 0$) of the 2nd test pattern (1001100) in leakage tests is faulty ($s_2 = 1$ is the faultless result). (e) The result ($s_3 = 0$) of the 3rd test pattern (0101010) in leakage tests is faulty ($s_3 = 1$ is the faultless result). Based on the faulty results, we know that the binary indices of the leaky control channels differ in the 2nd and the 3rd bits. Thus, the possible pair of leaky channels is: (001,010), (100,111) or (101,110), i.e., (1,2), (4,7) or (5,6).

IV. Result

Experimental method

We fabricated mLSI chips with various defects. The widths of flow channels and control channels are $100\mu m$ and $30\mu m$, respectively. The dimensions of valves are $100\mu m \times 300\mu m$ and the applied pressure is 160 kPa. In particular, we produced leakage defects by connecting a pair of control channels with small control channel segments of $10\mu m$, $20\mu m$ and $30\mu m$ widths. We tested the fabricated chips with the proposed test method. For each test pattern, we carried out the corresponding test operation in 4 steps:

1) Depressurize all control channels connected to the test module;
2) Rinse the test module by sending water from a fluid inlet through the test module;
3) Pressurize selected control channels according to the test pattern;
4) Send food dye from the same fluid inlet as in step 2 through the test module and inspect the fluid status in the test module.

Specifically, the depressurization and pressurization of the control channels (steps 1 and 3) could be done almost immediately; sending a type of fluids through the test module (steps 2 and 4) took about 10 to 20 seconds, depending on the number of blocked flow channels in the test module[3]. Considering the additional time to switch between different fluid inputs (food dye and water) and to inspect the test results, it took about a minute to carry out one test operation.

Case study 1: Demonstration of the proposed test method

Figure 12 shows the working mechanism of the test module. Figure 12(a) shows a chip containing 7 control channels. There is a blockage defect in the 6^{th} control channel and a $10\mu m$

[3]A flow channel can be blocked by a pressurized valve on that channel. When a flow channel is blocked, the hydraulic resistance of the test module will change, which affects the speed of the fluid movement [11].

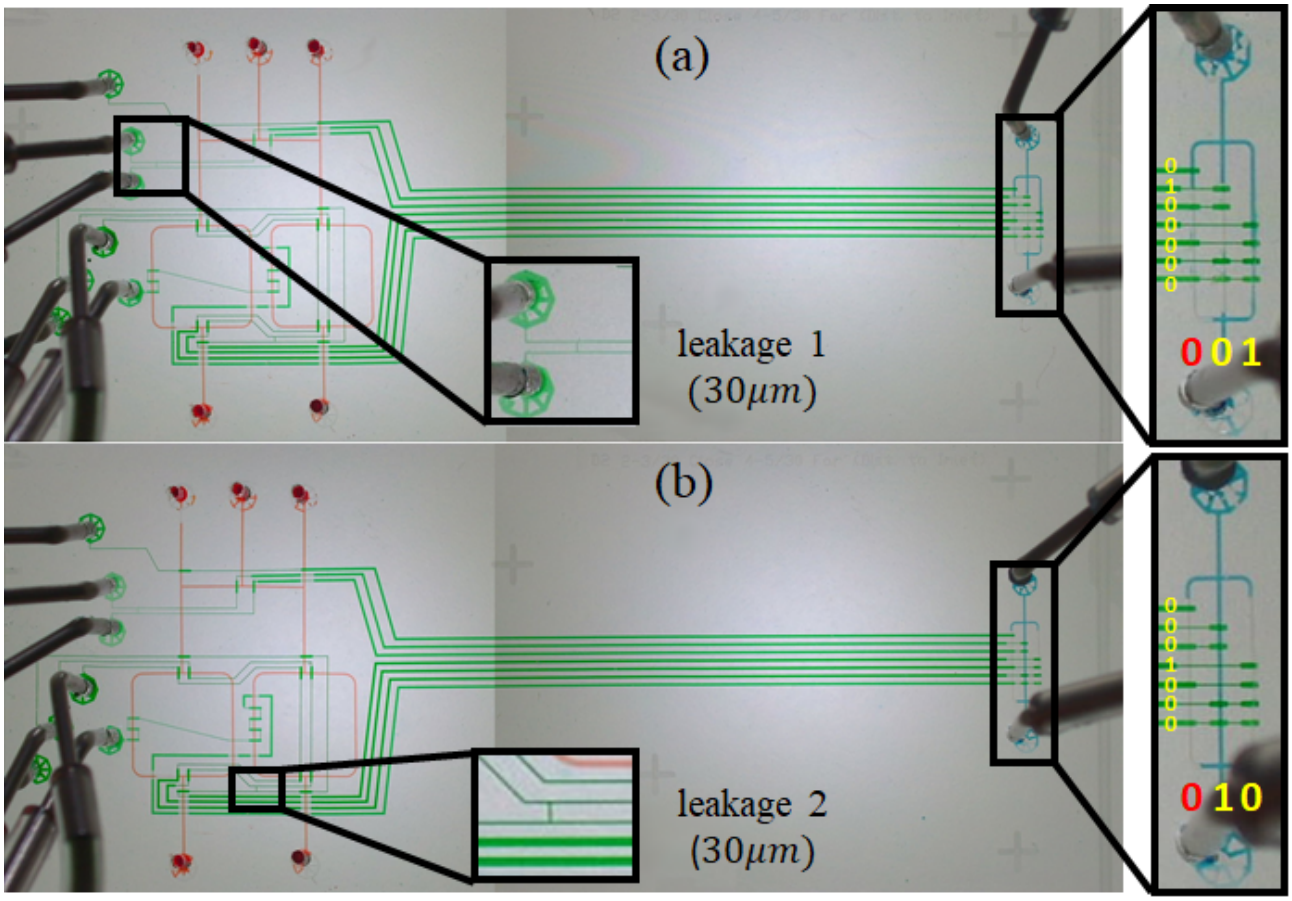


Figure 13: (a) Due to leakage 1, when we pressurize the 2^{nd} control channel, not only the middle but also the left flow channel in the test module is blocked, i.e. the result (001) of the test pattern 0100000 is faulty (101 is the faultless result). (b) Due to leakage 2, when we pressurize the 4^{th} control channel, not only the right but also the left flow channel in the test module is blocked, i.e. the result (010) of the test pattern 0001000 is faulty (110 is the faultless result).

TABLE I: Case study 1 — Test patterns and test results

Type of Defects	Test Pattern	Test Result		
		Expected	Inspected	Faultless?
Blockage	1100000	000	000	T
	0011000	000	000	T
	0000110	000	010	F
	0000001	000	000	T
Leakage	1110000	$s_1=1$	$s_1=1$	T
	1001100	$s_2=1$	$s_2=0$	F
	0101010	$s_3=1$	$s_3=0$	F

leakage defect between the 1^{st} and the 2^{nd} control channel. We performed 4 (i.e. $\lceil \frac{7}{2} \rceil$) and 3 (i.e. $\lceil \log_2(7+1) \rceil$) test operations to test for the blockage and the leakage defects, respectively. Table I shows the corresponding test patterns and test results.

- In the tests for blockage, we got faultless results for all except the 3^{rd} test pattern, as shown in Table I and in Figure 12(b). Specifically, when we pressurized the 5^{th} and the 6^{th} control channels, they failed to block the second flow channel in the test module. Thus, a blockage defect was detected.
- In the tests for leakage, we got faultless result for the 1^{st} test pattern and faulty results for the 2^{nd} and the 3^{rd} test patterns, as shown in Table I and in Figure 12(c), (d) and (e). Specifically, for the n^{th} test pattern, fluids were expected to pass through the n^{th} flow channel in the test module. Since only the 1^{st} pattern worked as expected, a leakage defect was detected.

Case study 2: Defects at different positions

Figure 13 shows an mLSI chip with two $30\mu m$ leakage defects between control channels $(2,3)$ and $(4,5)$, respectively. In particular, leakage 1 is close to the pressure inlet, and leakage 2 is far from the pressure inlet. Besides, we prolonged the control channels between the main circuit and the test module so that both defects are far from the test module.

To show that channel defects in the main circuit will lead to faulty behavior of valves in the test module regardless of the positions of the defects, we individually pressurized the 2^{nd} and the 4^{th} control channel.

- Specifically, since the 2^{nd} control channel is only connected to the valve in the middle flow channel of the test module, pressurizing it should only lead to blockage of the middle flow channel. However, due to leakage 1,

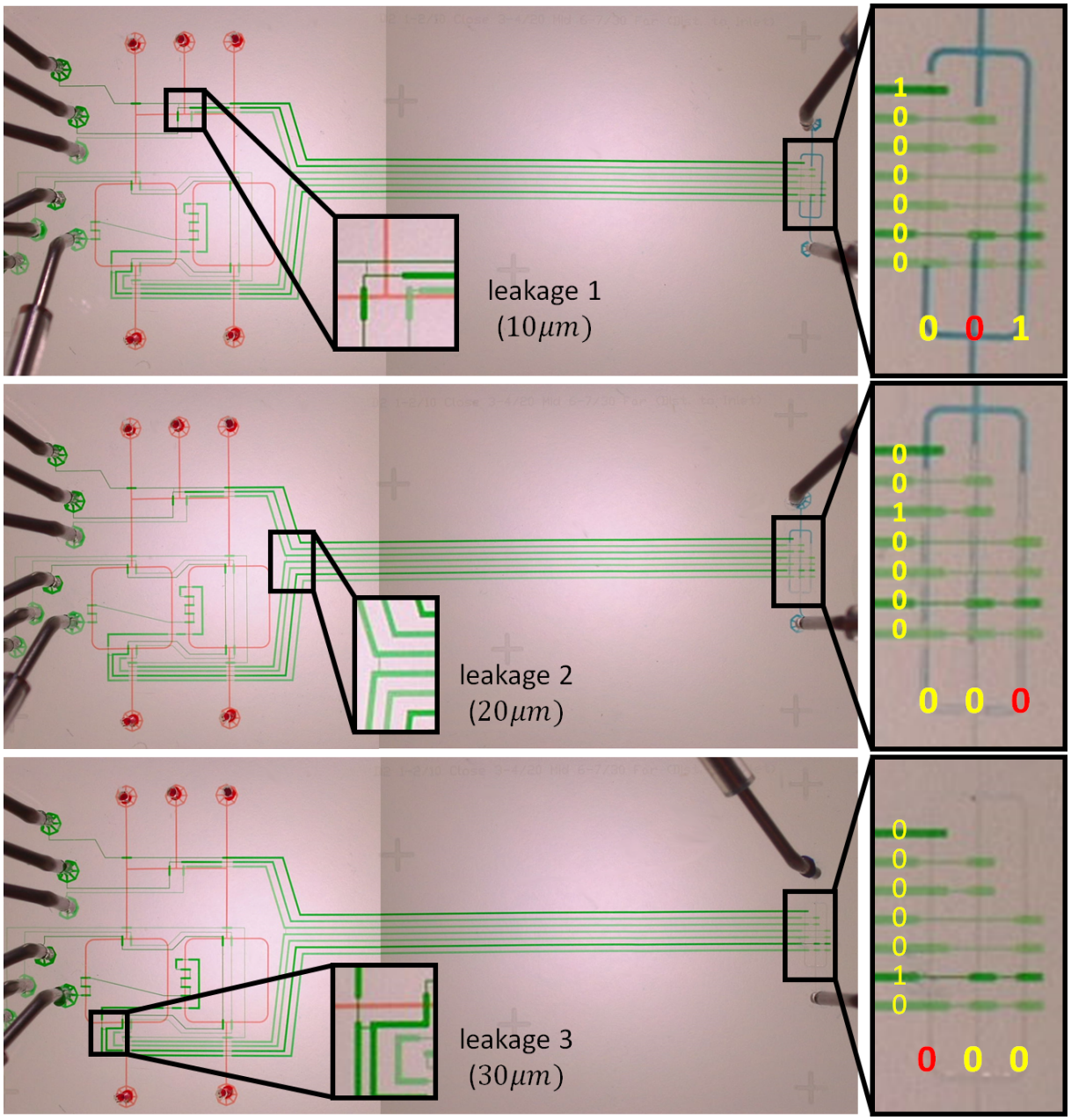


Figure 14: (a) Due to leakage 1, when we pressurize the 1^{st} control channel, not only the left but also the middle flow channel in the test module is blocked, i.e. the result (001) of the test pattern 1000000 is faulty (011 is the faultless result). (b) Due to leakage 2, when we pressurize the 3^{rd} control channel, not only the left and the middle, but also the right flow channel in the test module is blocked, i.e. the result (000) of the test pattern 0010000 is faulty (001 is the faultless result). (c) Due to leakage 3, when we pressurize the 6^{th} control channel, not only the middle and the right but also the left flow channel in the test module is blocked, i.e. the result (000) of the test pattern 0000010 is faulty (100 is the faultless result).

i.e. the leakage close to the pressure inlet, as shown in Figure 13(a), we observed that not only the middle, but also the left flow channel of the test module was blocked. The faulty result indicates that there is a leakage defect.

- Similarly, since the 4^{th} control channel is only connected to the valve in the right flow channel of the test module, pressurizing it should only lead to blockage of the right flow channel. However, due to leakage 2, i.e. the leakage far from the pressure inlet, as shown in Figure 13(b), we observed that not only the right, but also the left flow channel of the test module was blocked. The faulty result indicates that there is a leakage defect.

Case study 3: Defects of different feature sizes

Figure 14 shows a similar mLSI chip but with three leakage defects of $10\mu m$, $20\mu m$ and $30\mu m$ widths between control channels $(1,2)$, $(3,4)$ and $(6,7)$, respectively.

To show that channel defects in the main circuit will lead to faulty behavior of valves in the test module even when the

defects have small feature sizes, we individually pressurized the 1^{st}, the 3^{rd} and the 6^{th} control channel.

- Since the 1^{st} control channel is only connected to the valve in the left flow channel of the test module, pressurizing it should only lead to blockage of the left flow channel. However, due to leakage 1, i.e. the $10\mu m$ leakage, as shown in Figure 14(a), we observed that not only the left, but also the middle flow channel of the test module was blocked. The faulty result indicates the defect.
- Since the 3^{rd} control channel is connected to valves in the left and the middle flow channels of the test module, pressurizing it should lead to blockage of the left and the middle flow channels. However, due to leakage 2, i.e. the $20\mu m$ leakage, as shown in Figure 14(b), we observed that all flow channels were blocked. The faulty result indicates the defect.
- Since the 6^{th} control channel is connected to valves in the middle and the right flow channels of the test module, pressurizing it should lead to blockage of the middle and the right flow channels. However, due to leakage 3, i.e. the $30\mu m$ leakage, as shown in Figure 14(c), we observed that all flow channels were blocked. The faulty result indicates the defect.

V. Discussion

Comparison among different test methods

We compare three test methods: the functional test method [10] and the two test methods proposed by us based on the prototype and the enhanced test modules, respectively. Table II shows the efficiency and resource overhead of these test methods on three test cases from [10].

TABLE II: Efficiency and overhead of different test methods

Case	# n	Method	# tp_{block}	# tp_{leak}	# tp	# f
ChIP	28	Func. Test	NA	NA	30	\
		Prototype	1	5	6	28
		Enhanced	14	5	19	5
WGA	23	Func. Test	NA	NA	12	\
		Prototype	1	5	6	23
		Enhanced	12	5	17	5
Cell Culture	48	Func. Test	NA	NA	92	\
		Prototype	1	6	7	48
		Enhanced	24	6	30	6

#n: the number of control channels in the to-be-tested chip;
#tp_{block}: the number of test patterns for testing blockage defects;
#tp_{leak}: the number of test patterns for testing leakage defects;
#tp: the number of test patterns in total, i.e. # tp = # tp_{block} + # tp_{leak};
#f: the number of flow channels as resource overhead.

- Efficiency: both test methods based on integrated prototype/enhanced test modules are very efficient for testing leakage, while the prototype test module is extremely efficient for testing blockage. Considering that it takes about a minute to carry out a test pattern with our test approach, the leakage tests for each of the three test cases can be carried out in about 5 to 6 minutes, while the blockage tests take about one minute and $14-24$ minutes with the prototype test module and the enhanced test module, respectively. For testing a given number of control channels, the number of test operations applying our methods is deterministic, while the number of test operations applying the functional test method is not deterministic. In general, our methods are significantly more efficient than the functional test method in terms of the number of test operations.
- Resource overhead: our methods require integrating extra flow channels onto the to-be-tested chip. Compared to the prototype design, the enhanced test module design significantly reduces the number of the required flow channels, and thus reduces the area overhead. The functional test method does not require additional on-chip resources, but it requires external equipment such as pressure sources, pressure sensors, and commercial ATPG tools.
- Complexity of the test protocols: it is very simple and intuitive to generate the test patterns and to analyze the results for the two test-module based methods. No specialized equipment or knowledge is required for carrying out the tests. On the other hand, the test pattern generation and result analysis for the functional test method requires professional software and computational efforts. As reported in [9], the cost of their testing system is less than \$1000. Besides, knowledge and skills are required for carrying out the functional tests.

TABLE III: Area of the main circuit and the test modules

Case	# n	$Area_m$	Type	$Area_t$	$Area_{t/m}$
Kinase	15	17.2×15	Prototype	6.9×3	8%
			Enhanced	6.9×0.8	2.1%
Nucleic Acid	17	16.3×24.55	Prototype	7.3×3.4	6.2%
			Enhanced	7.3×1	1.8%
ChIP (4IP)	26	14.2×43.5	Prototype	9.1×5.2	7.6%
			Enhanced	9.1×1	1.4%
mRNA	23	29.1×21.25	Prototype	8.5×4.6	6.3%
			Enhanced	8.5×1	1.3%
ChIP (10IP)	31	47.9×36.45	Prototype	10.1×6.2	3.6%
			Enhanced	10.1×1	0.6%
Cell-free bio. net.	24	20×37	Prototype	8.7×4.8	5.6%
			Enhanced	8.7×1	1.2%

The area is calculated as length $\times$ width.
#n: the number of control channels in the to-be-tested chip;
$Area_m(mm^2)$: the area of the main circuit, i.e. the original chip design.
$Area_t(mm^2)$: the area of the test module.
$Area_{t/m}$: *the proportion of the area of the test module to the area of the main circuit.*

Implementation overhead of the test methods

To investigate the implementation overhead of the proposed test methods, we calculate the area consumptions of the test modules for testing six mLSI designs proposed in [12], and compare them with the area of the original designs. The calculation follows the same design rules as applied in [12], where the width of each flow channel and each valve is $100\mu m$, the diameter of each fluid inlet plus a safety margin

is $1600\mu m$, and the minimum spacing distance between two channels is $100\mu m$. Besides, we also reserve some additional length ($300\mu m$) to each flow channel of the test module for the convenience of inspection. Table III shows the results of the comparison.

As we can see from the table, the area of the test module is dependent on the number of the to-be-tested control channels. Compared with the prototype design, the enhanced design has much fewer flow channels and thus occupies much less area. In general, the prototype test module causes about $3.6-8\%$ area overhead, while the enhanced test module causes only about $0.6-2.1\%$ area overhead, which is quite modest.

Besides, in order to integrate the test module onto the chip, additional control channel routing efforts will be required, which also results in additional area overhead. But for mLSI designs where all control channel ends are connected to the same chip boundary, e.g. the designs synthesized by Columba S [13], a state-of-the-art mLSI physical design tool, we can expect the routing efforts to be fairly small.

For the testing of n control channels, the prototype test module needs to be integrated with n valves, and the enhanced test module needs to be integrated with up to $\lceil \frac{n}{2} \rceil \lceil \log_2(n+1) \rceil$ valves. To note is that prolonging control channels and integrating valves into the test module will not notably increase the required pressure to drive the valves. Experiments conducted in [14] demonstrated that it is reliable to pressurize/depressurize thousands of valves simultaneously at around 180 and 280 kPa.

VI. Future Directions

Automation

Considering that it is still common practice among mLSI users to test the chips by visual inspection, the proposed methods do not demand the use of sensors, which may call for additional equipment costs. But in case that sensors are available, they can be integrated into the proposed test modules to automate the test process and thus save the time and efforts for visual inspection. Compared to naïve mLSI test methods that directly inspect the valve behavior, our methods do not require high-resolution images since we can use colored fluids such as food dye to reflect the channel defects in the test modules without contaminating the main circuit. Besides, our methods only require to inspect a small chip area, which makes it possible to obtain all necessary images without moving the camera.

False alarm

The proposed test methods focus on the main circuit. If there is no defect in the main circuit, but there are some defects in the test module, a false alarm will be caused. Thus, for control channels outside the main circuit, we can enlarge their width to reduce the chance of blockage defects, and enlarge their spacing distance to reduce the chance of leakage defects, as shown in Figure 4, 13 and 14. To accurately evaluate the false positive rate, further experiments and quantitative analysis are required as the next step.

VII. Conclusion

This work proposes the first integrated test module designs for mLSI. Compared to the state-of-the-art functional test methods, the built-in-self-test methods based on integrated test modules significantly increase the fault coverage and test efficiency. Besides, the test protocols are greatly simplified, so that mLSI users who are not testing experts can carry out the tests by themselves without costly equipment.

VIII. Acknowledgment

We thank Dr.-Ing. Jan Otterstedt from Infineon Technologies and Technical University of Munich for his inspiring ideas, which initiate the design of the test module.

References

[1] A. R. Wu, T. L. Kawahara, N. A. Rapicavoli, J. van Riggelen, E. H. Shroff, L. Xu, D. W. Felsher, H. Y. Chang, and S. R. Quake, "High throughput automated chromatin immunoprecipitation as a platform for drug screening and antibody validation," *Lab on a Chip*, vol. 12, pp. 2190–2198, 2012.

[2] R. A. W. III, P. C. Blainey, H. C. Fan, and S. R. Quake, "Digital PCR provides sensitive and absolute calibration for high throughput sequencing," *BMC Genomics*, 2009.

[3] H. Huang and D. Densmore, "Integration of microfluidics into the synthetic biology design flow," *Lab on a Chip*, vol. 14, no. 18, pp. 3459–3474, 2014.

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

[5] M. Li, T.-M. Tseng, B. Li, T.-Y. Ho, and U. Schlichtmann, "Component-oriented high-level synthesis for continuous-flow microfluidics considering hybrid-scheduling," in *Proceedings of the 54th Annual Design Automation Conference 2017*, 2017, pp. 51:1–51:6.

[6] J. Liu, M. Enzelberger, and S. Quake, "A nanoliter rotary device for polymerase chain reaction," *Electrophoresis*, vol. 23, pp. 1531–1536, 2002.

[7] Stanford Foundry, *Basic Design Rules*. http://web.stanford.edu/group/foundry, 2022.

[8] K. Hu, T.-Y. Ho, and K. Chakrabarty, "Testing of flow-based microfluidic biochips," in *2013 IEEE 31st VLSI Test Symposium (VTS)*. IEEE, 2013, pp. 1–6.

[9] 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, no. 10, pp. 1463–1475, 2014.

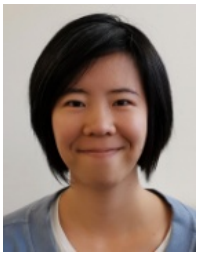
[10] K. Hu, B. B. Bhattacharya, and K. Chakrabarty, "Fault diagnosis for leakage and blockage defects in flow-based microfluidic biochips," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 7, pp. 1179–1191, 2015.

[11] M. Li, T.-M. Tseng, Y. Ma, T.-Y. Ho, and U. Schlichtmann, "VOM: Flow-path validation and control-sequence optimization for multilayered continuous-flow microfluidic biochips," 2019.

[12] T.-M. Tseng, M. Li, D. N. Freitas, T. McAuley, B. Li, T.-Y. Ho, I. E. Araci, and U. Schlichtmann, "Columba 2.0: A co-layout synthesis tool for continuous-flow microfluidic biochips," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 8, pp. 1588–1601, 2018.

[13] T.-M. Tseng, M. Li, D. N. Freitas, A. Mongersun, I. E. Araci, T.-Y. Ho, and U. Schlichtmann, "Columba s: A scalable co-layout design automation tool for microfluidic large-scale integration," in *Proceedings of the 55th Annual Design Automation Conference*, 2018, pp. 163:1–163:6.

[14] I. E. Araci and S. R. Quake, "Microfluidic very large scale integration (mVLSI) with integrated micromechanical valves," *Lab on a Chip*, vol. 12, pp. 2803–2806, 2012.



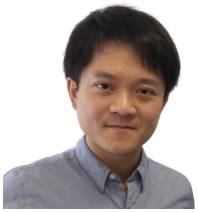
Mengchu Li (S'16) received the B.A degree in German Studies from Tongji University in China. She then came to Germany and received the B.Sc and M.Sc degrees in Computer Science from Ludwig-Maximilians-Universität München (LMU). She is currently a doctoral researcher in the Chair of Electronic Design Automation at Technical University of Munich (TUM). Her research interests include design automation for continuous-flow microfluidics and optical networks-on-chips.

Yushen Zhang (S'22) received the B.Sc. and M.Sc. degrees in Computer Science (Informatics) from the Technical University of Munich (TUM), Germany, in 2017 and 2020, respectively. He is currently a research associate and doctoral candidate at the Chair of Electronic Design Automation, School of Computation, Information and Technology, TUM. He is working on various research topics, including computer-aided design, microfluidic biochips, and 3D-printed microfluidics.

Ju Young Lee received his B.S. and M.S. degrees from the bioengineering department at Santa Clara University. During his time there, he joined Prof. I. Emre Araci's group to research microfluidic biosensors. His focus has been on skin mountable microfluidic sensors for monitoring human body activity.

Hudson Gasvoda received his combined B.S and M.S in Mechanical Engineering from Worcester Polytechnic Institute in 2020. He is currently a research assistant at Santa Clara University under Dr. Araci. His research interests include microfluidic sensors, digital image correlation, and biowearables.

I. Emre Araci has received his B.S. and M.S. degrees from Electrical Engineering Department, Ege University, Izmir, Turkey. He received his Ph.D. from the College of Optical Sciences, the University of Arizona in 2010. He then joined Prof. Stephen Quake's group in the Bioengineering Department at Stanford University as a postdoctoral associate and as the director of the Stanford Microfluidics Foundry (SMF). He joined the Department of Bioengineering at Santa Clara University in 2015. He has been an Associated Professor in the same department, since 2021. His primary research goals are directed toward the development and application of wearable microfluidic and optofluidic technologies for continuous health monitoring.



Tsun-Ming Tseng (S'10–M'15) received the bachelor degree in electronics engineering from National Chiao Tung University (NCTU), Hsinchu, Taiwan, in 2010, the M.Sc. and the Dr.-Ing. degrees from Technical University of Munich (TUM), Munich, Germany, in 2013 and 2017, respectively. He leads a research group in the Chair of Electronic Design Automation, TUM. His research interests focus on design automation for emerging technologies, such as microfluidic biochips and optical networks-on-chips.

Ulf Schlichtmann (S'88–M'90–SM'18) received the Dipl.Ing. and Dr.-Ing. degrees in electrical engineering and information technology from Technical University of Munich (TUM), Munich, Germany, in 1990 and 1995, respectively. He was with Siemens AG, Munich, and Infineon Technologies AG, Munich, from 1994 to 2003, where he held various technical and management positions in design automation, design libraries, IP reuse, and product development. He has been a Professor and the Head of the Chair of Electronic Design Automation, TUM, since 2003, where he also served as Dean of the Department of Electrical and Computer Engineering, from 2008 to 2011, and as Associate Dean of Studies of International Studies since 2013. Since 2016, he is an elected member of TUM's Academic Senate and Board of Trustees. He is a member of the German National Academy of Science and Engineering and also serves on a number of advisory boards. Ulf's current research interests include computer-aided design of electronic circuits and systems, with an emphasis on designing reliable and robust systems. In recent years, he has increasingly worked on emerging technologies, such as microfluidic biochips and optical interconnect.