Multiple Fault Diagnosis in Digital Microfluidic Biochips

DANIEL DAVIDS, SIDDHARTHA DATTA, ARINDAM MUKHERJEE, BHARAT JOSHI, and ARUN RAVINDRAN
University of North Carolina, Charlotte

Microfluidics-based biochips consist of microfluidic arrays on rigid substrates through which, movement of fluids is tightly controlled to facilitate biological reactions. Biochips are soon expected to revolutionize biosensing, clinical diagnostics, and drug discovery. Critical to the deployment of biochips in such diverse areas is the dependability of these systems. Thus, robust testing techniques are required to ensure an adequate level of system dependability. Due to the underlying mixed technology and energy domains, such biochips exhibit unique failure mechanisms and defects. In this article we present a highly effective fault diagnosis strategy that uses a single source and sink to detect and locate multiple faults in a microfluidic array, without flooding the array, a problem that has hampered realistic implementations of all existing strategies. The strategy renders itself well for a built-in self-test that could drastically reduce the operating cost of microfluidic biochips. It can be used during both the manufacturing phase of the biochip, as well as field operation. Furthermore, the algorithm can pinpoint the actual fault, as opposed to merely the faulty regions that are typically identified by strategies proposed in the literature. Also, analytical results suggest that it is an effective strategy that can be used to design highly dependable biochip systems.

Categories and Subject Descriptors: B.8.1 [**Performance and Reliability**]: Reliability, Testing, and Fault-Tolerance; I.6.5 [**Simulation and Modeling**]: Model Development; I.6.6 [**Simulation and Modeling**]: Simulation Output Analysis

General Terms: Algorithms, Reliability, Performance

Additional Key Words and Phrases: Microfluidic biochip, multiple fault, testing, faults tolerance, droplet flooding

1. INTRODUCTION

Over the past decade, research in integrated circuit testing has broadened from digital testing to include the testing of analog and mixed-signal devices. More recently, new test techniques for mixed-technology microelectromechanical

Author's address: D. Davids, S. Datta, A. Mukherjee (contact author), B. Joshi, A. Ravindran, Department of ECE, University of North Carolina at Charlotte, 9201 University City Blvd., Charlotte, NC 28223-0001; email: amukherj@uncc.edu.

systems (MEMS) have also been receiving attention [Deb and Blanton 2000, 2004; Dhayni et al. 2004; Kolpekwar and Blanton 1997; Mir et al. 2000]. As MEMS rapidly evolves from single components to highly integrated systems for safety-critical applications, dependability is emerging as an important performance parameter. Fabrication techniques such as silicon micromachining lead to new types of manufacturing defects in MEMS [Deb and Blanton 2000]. Moreover, due to their underlying mixed technology and multiple energy domains (e.g., electric, mechanical, and fluidic), such composite microsystems exhibit failure mechanisms that are significantly different from those in electronic circuits. In fact, the 2003 *International Technology Roadmap for Semiconductors* (ITRS) recognizes the need for new test methods for the disruptive device technologies that underlie composite microsystems, and highlights them as one of the five difficult test challenges beyond 2009 [International Technology Roadmap 2006].

Microfluidics-based biochips constitute an emerging category of mixed-technology microsystems [Su et al. 2005b]. Recent advances in microfluidics technology have led to the design and implementation of miniaturized devices for various biochemical applications. These microsystems, referred to interchangeably in the literature as microfluidics-based biochips, lab-on-a-chip, and bioMEMS [Pollack et al. 2002; Su and Chakrabarty 2004] promise to revolutionize biosensing, clinical diagnostics, and drug discovery. Such applications can benefit from the small size of biochips, as well as their use of microliter/nanoliter sample volumes, lower cost, and higher sensitivity compared to conventional laboratory methods.

Earlier generations of microfluidics-based biochips were based on the manipulation of continuous liquid flow through fabricated microchannels [Su et al. 2005b]. Liquid flow was achieved through external pressure sources, integrated mechanical micropumps, or electrokinetic mechanisms, such as electro-osmosis. Recently, a novel microfluidics technology has been developed to manipulate liquids as discrete microliter/nanoliter droplets. Following the analogy of digital electronics, this technology is referred to as "digital microfluidics" [Pollack et al. 2002]. Compared to continuous-flow systems, digital microfluidics offers the advantage of dynamic reconfigurability and architectural scalability.

The level of system integration and the complexity of digital microfluidics-based biochips are expected to increase in the near future due to the growing need for multiple and concurrent bioassays on a chip [Su and Chakrabarty 2004]. However, shrinking processes, new materials, and the underlying multiple energy domains will make these biochips more susceptible to manufacturing defects. Moreover, some manufacturing defects are expected to be latent, and may manifest during field operation of the biochips. In addition, harsh operational environments may introduce physical defects, such as particle contamination, during field operation. Consequently, effective test techniques are required to ensure system dependability as biochips are deployed for safety-critical applications such as field diagnostics tools to monitor infectious disease and decode genes, as well as biosensors to detect biochemical toxins and other pathogens. In this article an efficient and effective fault diagnosis algorithm is

presented that:

—can detect and locate single, as well as multiple, faults;
—avoids flooding;
—pinpoints the actual fault, as opposed to the faulty regions that are typically identified by strategies proposed in the literature;
—renders itself well for built-in self-testing thereby eliminating the need for elaborate and expensive external test equipment; and
—can be used to test biochips during both the manufacturing phase and field operation.

The organization of the remainder of the article is as follows. Related prior work is described in Section 2. Next, fault modeling for digital microfluidic biochips and fault types is discussed in Section 3. Section 4 describes the proposed diagnosis methodology and its analysis. Finally, conclusions are drawn in Section 5.

2. PRIOR WORK

Although research in the design of digital microfluidics-based biochips has gained considerable momentum in recent years [Pollack et al. 2002; Srinivasan et al. 2004; Su and Chakrabarty 2004], to the best of our knowledge, only one group of researchers has reported work on the testing of digital microfluidic biochips [Su and Chakrabarty 2004, 2005, 2006a, 2006b; Su et al. 2003, 2004a, 20004b, 2005a, 2005b, 2006]. A cost-effective test methodology for digital microfluidic systems was first described in [Su et al. 2003]. Physical defects in such systems were analyzed and faults were classified as being either catastrophic or parametric. In all the proposed strategies, faults are detected by electrically controlling and tracking the motion of test droplets. An optimal test planning method for the detection of catastrophic faults in digital microfluidic arrays was investigated in [Su et al. 2004a]. It is based on a graph model of the array and a problem formulation based on Hamiltonian paths in a graph. An efficient concurrent testing method that interleaves test application with a set of bioassays was first proposed in [Su et al. 2004b]. Reconfiguration and defect tolerance techniques for biochips were described in [Su et al. 2005a; Su and Chakrabarty 2005, 2006b].

Prior work on the testing of digital microfluidics-based biochips is based on simplistic assumptions regarding the impact of certain defects on droplet flow. For example, a common defect seen in fabricated microfluidic arrays is a short circuit between two adjacent electrodes [Su et al. 2003]. It was assumed in Su et al. [2003, 2004a, 2006] that this defect causes a droplet to get stuck at one of the two electrodes, irrespective of the orientation of liquid flow. Based on this assumption, Hamiltonian paths are used to detect catastrophic faults in microfluidic arrays [Su et al. 2004a, 2006]. One of the problems with this approach is that although finding Hamiltonian paths in grid structures is well-known, checking the existence of a Hamiltonian path in a given graph is NP-complete. Thus, it would be expensive to determine such paths in the microfluidic array after it is reconfigured based on results of the fault diagnosis.

Furthermore, an attempt was made to experimentally validate the preceding assumption. Experiments suggested that the effect of such a short-circuit defect on droplet flow depends on whether the droplet flow path is perpendicular to the two shorted electrodes or aligned with them [Su et al. 2005b]. A test procedure for such defects should therefore not only test single cells, as in Su et al. [2003, 2004a, 2006] using Hamitonian paths, but should also focus on pairs of cells and the traversal of droplets from one cell to all its neighbors. A dfferent testing methodology based on the Euler path was developed to detect catastrophic faults, including those caused by electrode shorts [Su et al. 2005b]. Although this method is better than that based on Hamiltonian paths in terms of diagnosibility, it has one major shortcoming: When an electrode i is faulty, this fault cannot be distinguished from any other fault that may exist in a closed tour starting at i in the Euler path. The algortihm presented in this article can always uniquely diagnose such faulty electrodes.

Moreover, all the methods in the literature, including that proposed in Su et al. [2005b], can diagnose only single fault, and when the droplet fails to reach the sink (indicating the presence of a fault), these methods do not retrieve the droplet back to the source. This could eventually lead to flooding, an undesirable phenomemon in which droplets accumulate at an electrode.

A binary search method is used in Su et al. [2005b] to locate the faulty region (edge/electrode). This involves dividing the remaining graph into two halves at every stage of the search. The number of steps required to locate a single fault in an m x n array is $\lceil \log_2(m-1) \rceil + \lceil \log_2(n-1) \rceil$. The search is done in parallel.

When $m = n$, the number of edges incident to half of the nodes in the microfluidic array is $n(n-1)$. However, to ensure that the edges connecting the two halves are also tested, one of the two halves will have $n(n/2+1)$ nodes, and the corresponding number of edges that has to be traversed in this half would be $(n^2 + n/2 - 1)$. Thus, the corresponding recurrence relationship for the total number of edges traversed for an area of dimension nx $n$on the microfluidic array is:

$$T(n) = T\left(\frac{n}{2}\right) + (n^2 + n/2 - 1).$$

Hence, all solutions are of the form

$$T(n) = C_1 + C_2 n + C_3 n^2,$$

which implies that $T(n) \in O(n^2)$, where n is a power of 2.

However, $T(n)$ is eventually nondecreasing with the size of n, and so the asymptotic results obtained apply unconditionally for all values of n. Thus $T(n) = O(n^2)$.

As mentioned earlier, the method of Su et al. [2005b] has certain limitations such as handling multiple faults, and potential flooding issue in presence of any fault.

In this article, all the issues identified earlier are addressed by proposing a testing methodology that can not only detect and locate multiple faults, but also avoid flooding.

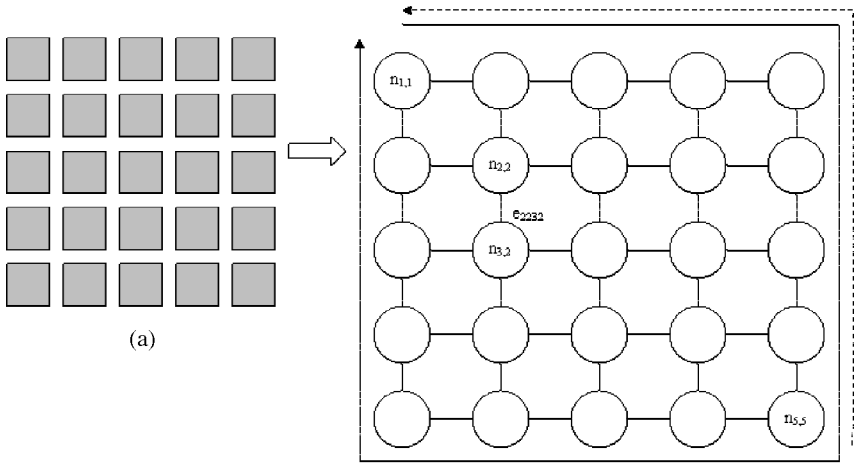

Fig. 1. A 5 × 5 microfluidic array and its corresponding graph.

3. FAULT MODELING

The microfluidic biochip discussed in Su et al. [2005b] is based on the maneuvering of microliter # nanoliter droplets using the principle of electrowetting-on-dielectric (EWOD) [Pollack et al. 2002]. Detailed descriptions of the different kinds of faults that may occur in a microfluidic chip are given in Su et al. [2003, 2005b]. In this article, it is assumed that the biochip uses the principle of EWOD and that the types of faults that could occur will be those identified in Su et al. [2003, 2005b].These faults include:

(1) A short between the droplet and the electrode. This typically happens due to the application of unusually high voltage, resulting in dielectric breakdown. Under this condition, the droplet typically undergoes electrolysis and further movement of the droplet is not possible.
(2) An electrode short due to metal connection between two adjacent electrodes caused by abnormal metal layer deposition and etch variation during fabrication. Under this fault condition, a droplet is stuck between the two shorted electrodes and further movement of the droplet cannot be achieved.
(3) An open circuit between the electrode and the control source due to variability in the manufacturing process. An electrode cannot be activated under this fault condition.
(4) Several conditions, such as particle contamination, cause the impedance between plates to become high, leading to fluidic open. A droplet cannot move across the obstacle under this fault condition.

Thus, it can be seen that an effective testing strategy should test each electrode and edge in the biochip.

The microfluidic array is modeled as a graph (Figure 1) in which a node is numbered as $n_{i,j}$, where i and j denote the row and column numbers,

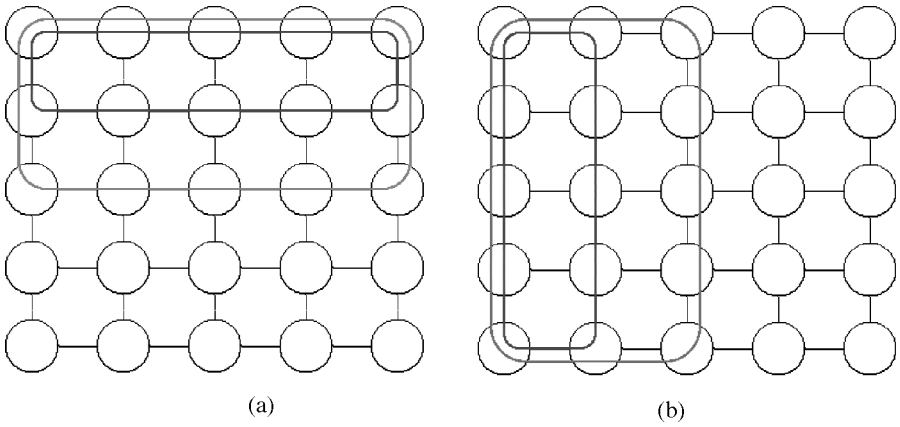

Fig. 2. Testing of nodes, horizontal, and vertical edges.

respectively, at which the node is located in the graph. Similarly, if there is an edge between $n_{i,j}$ and $n_{k,l}$, then this edge is labeled e_{ijkl}. It is assumed that, at most, one fault can exist in any rectangular loop (Figure 2) in which at least three of the sides of the rectangle are sides of the outer rectangular loop. Also, no failure takes place during any round of testing.

4. TESTING AND DIAGNOSIS

It was identified in Section 3 that test planning methods based on Hamiltonian paths [Su et al. 2003, 2004b, 2006] are not sufficient for fault diagnosis. In Su et al. [2005b], a diagnosis method based on the Euler path was presented. However, in this method, a closed tour at a node will lead to a nonunique fault diagnosis. Thus, a different fault diagnosis approach is required.

Furthermore, in the literature two types of testing strategies are proposed, namely, the offline testing that is used during the manufacturing phase to increase yield, and online, or concurrent, testing for field operations. In online testing, the cells are partitioned into two sets: a set of cells being used by the bioassays and the set of unused cells. The testing is done only on the unused cells. This approach has several disadvantages, including incomplete testing of the biochip, the need for sophisticated heuristics for test schedules that do not always guarantee optimal test paths from source to sink, and the fact that a test cannot be performed if a path cannot be found from the source to sink, depending on the set of cells that are being used by the bioassays. Furthermore, the two sets of cells that are created are static. The focus of this article is to design a test strategy that can be used during both the manufacturing phase and field operation, renders itself well for a built-in self-test (BIST), can test all the electrodes and edges, detect and locate multiple faults, and avoids flooding. It has been shown in Su et al. [2005b] that a droplet can be transferred from one cell to an adjacent one in milliseconds. Thus, the time required to test the biochip would be quite small. The biochip can be tested at any time when it is free of bioassays.

4.1 Basic Testing Strategy

The proposed strategy consists of two basic steps. In the first step, the outer loop (boundary nodes and edges) of the microfluidic array is tested. A test droplet is dispensed onto the array from the droplet source (i.e., on-chip reservoir and dispensing port). This is routed through the biochip-under-test, traversing all the electrodes and their respective edges of the outer loop. For example, in Figure 1, a closed tour could be$n_{1,1} \rightarrow n_{1,2} \rightarrow \cdots \rightarrow n_{5,5} \rightarrow \cdots \rightarrow n_{2,1} \rightarrow n_{1,1}$, assuming that $n_{1,1}$ is both the source and the sink. For consistency, it will be assumed that these tours are always in a clockwise direction. If a fault exists in the loop, then the droplet will get stuck. Otherwise, it will be eventually guided back to the sink. The sink electrode, which is also the source electrode in this step, is connected to a capacitive detection circuit that can determine the presence of the test droplet [Srinivasan et al. 2004]. Thus fault detection in the outer loop is achieved.

Once a fault is detected in the outer loop, it is necessary to locate this fault so that appropriate reconfiguration of the loop can be performed. An attempt is made to retrieve the droplet by applying voltages to the electrodes $n_{m-1,n}, , n_{m-2,n}, \ldots, n_{1,1}$ sequentially, as shown by the dashed line in Figure 1. It should be noted that the retrieval is done in a counterclockwise direction. If the droplet is retrieved, then the fault is present in that half of the loop. Otherwise, it is present in the other half. The segment of the loop where the droplet is present is further divided into two halves and an attempt is made to retrieve the droplet in a counterclockwise direction, as done before. This search, based on the binary search technique, is done until the droplet is retrieved at the sink. At this point, the smallest segment containing the fault has been identified. A droplet is sent through the source in a clockwise direction until the end of the segment in which the fault is present by applying voltages to appropriate electrodes sequentially. Once again, an attempt is made to retrieve the droplet by using the binary search technique. This process is repeated until the segment size finally reduces to one node that contains the droplet. The immediate clockwise neighbor and the edge connecting these two nodes are now considered to be suspicious.

Therefore, it is necessary to conduct further diagnosis to identify the fault. A droplet is sent in a counterclockwise direction to the immediate counterclockwise neighbor of the node that is currently considered to be suspicious. An attempt is now made to retrieve the droplet in the clockwise direction by applying voltages to electrodes sequentially, starting from the second neighbor of the suspicious node in a clockwise direction. If the droplet is retrieved successfully, it indicates that the droplet was stuck at the immediate neighbor of the suspicious node in the clockwise direction, implying that the suspicious node is faulty while the edge is still considered to be suspicious. Further tests to identify the status of the suspicious edge are not done, since it is incident to the faulty node and edges incident to the faulty node are no longer going to be used. If the droplet is not retrieved, a second attempt is made by applying voltages to the electrodes sequentially starting from the immediate neighbor of the suspicious node. Successful retrieval of the droplet implies that the suspicious node is fault-free, while the suspicious edge is faulty.

Reconfiguration based on the shortest path is done once the fault has been located. For an array with mx n nodes, if node n_{i1} is faulty, it is by passed by the following path: $n_{i-1,1} \rightarrow n_{i-1,2} \rightarrow n_{i,2} \rightarrow n_{i+1,2} \rightarrow n_{i+1,1}$. Similar reconfigurations are done for faulty nodes $n_{i,n}$, $n_{1,i}$, and $n_{m,i}$. The fault diagnosis and reconfiguration steps are repeated until a fault-free outer loop is identified or untill the array is rendered useless due to the unavailability of sufficient resources.

The second basic step involves fault detection and location among the nodes and edges that are contained within the outer loop. Faults in all horizontal paths (Figure 2) are identified and followed by the identification of faults in vertical paths. The fault detection and location procedure used for the outer loop is applied to all the loops in this step. In this step, the faults could exist only in paths that are not part of the outer loop, since the outer loop has already been identified as fault-free in Step 1.

4.2 Algorithm MFDL * Microfluidic array fault detection and location algorithm*\

Step 1

1. **While** not (done)
2. Dispense a test droplet from the source into the outer loop (initially boundary nodes and edges) in the clockwise direction.
3. **If** the droplet is retrieved at the sink after $|E|$ units of time then
4. outer loop is fault-free
5. \ $*|E|$ is the number of edges in the loop. It is assumed that a droplet takes one unit of time to traverse an edge *\
6. **Else**
7. edge = 0, node = 0
8. FaultLocation (loop, source, sink, edge, node)
9. ReconfigureOuter (edge, node)
10. **Endif**
11. **If** (outer Loop is fault-free) OR (microfluidic array useless)
    *An example of a microfluidic array becoming useless - a critical resource is disconnected from the rest of the system *\
12. done = true
13. **EndIf**
14. **EndWhile**

End Step 1

Step 2

1. Dispense a test droplet from the source to each of the horizontal loops whose three sides are a path from the Outer Loop (Figure 2(a)) starting from the smallest loop at an interval of 3 time units in the clockwise direction.
2. Mark each loop i from whom the droplet was not retrieved in $(|E_i| + d_i)$ units of time where $|E_i|$ is the number of edges in the loop i and d_i is the delay in the droplet dispensing * Faults are detected in horizontal loops *\
3. **For** each marked loop
   edge = 0, node = 0
   FaultLocation (loop, source, sink, edge, node)
4. **EndFor** * All the faulty nodes and all the faulty horizontal edges have been identified *\
5. **For** each node identified faulty
6. ReconfigureInner (node)

7. **EndFor**
8. Dispense a test droplet from the source to each of the vertical loops whose three sides are a path from the Outer Loop (Figure 2(b)) starting from the smallest loop at an interval of 3 time units in the clockwise direction
9. Mark each loop i from whom the droplet was not retrieved in $(|E_i| + d_i)$ units of time where $|E_i|$ is the number of edges in the loop i and d_i is the delay in the droplet dispensing * Faults are detected in vertical loops *\
10. **For** each marked loop
        FaultLocation (loop, source, sink, edge, node)
11. **EndFor** * All the faulty vertical edges have been identified *\

End Step 2

Retrieve (node_and_edge, path, firstNode, lastNode, done)

1. **If** (node-edge) then
2. Apply voltages to electrodes sequentially starting from the counterclockwise neighbor of the lastNode
3. **Else**
4. Apply voltages to electrodes sequentially starting from the clockwise second neighbor of the lastNode
5. **EndIf**
6. **If** droplet is retrieved then
7. done = true
8. **Else**
9. done = false
10. **EndIf**

End Retrieve

FaultLocation (loop, source, sink, faultyEdge, faultyNode)

1. located = false
2. retrieved = false
3. nodeEdge = true
4. path = first half of the loop starting from source
5. lastNode = sink
6. size = 0
7. **While** not (located)
8. Retrieve (nodeEdge, path, source, lastNode, retrieved)
9. **If** not (retrieved)
10. path = path $\cup$ p_1, where p_1 is the first half of the remaining loop
11. lastNode = last node in the path
12. size = number of nodes in p_1
13. **Else**
14. Dispense test droplet in the path by applying voltages to the electrodes in the clockwise direction starting from the source
15. path = first half of the path
16. **If** (size == 1)
17. Retrieve (nodeEdge, path, source, lastNode, retrieved)
18. located = true
19. FaultSet = {node in p_1, clockwise edge of the node in p_1}
20. **EndIf**
21. **EndIf**
22. **EndWhile**
23. Dispense test droplet from the sink in the counterclockwise direction until the candidate faulty node
24. lastNode = second clockwise neighbor of the candidate faulty node
25. nodeEdge = false
26. Retrieve (nodeEdge, path, source, lastNode, retrieved)

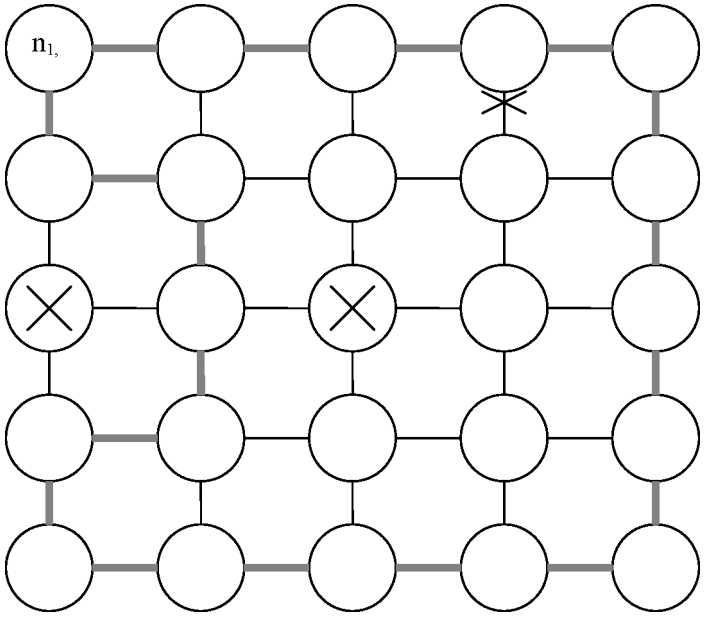

Fig. 3. A microfluidic array with three faults and a reconfigured outer loop.

27. **If** (retrieved)
28. faultEdge = edge between the candidate faulty node and the clockwise neighbor
29. **Else**
30. faultyNode = candidate faulty node
31. **EndIf**

End FaultLocation

ReconfigureOuter (edge, node)

1. **If** node is faulty then
2. Delete faulty node and the edges incident on that node
3. Reconfigure the loop * if $n_{i1}$is faulty then reconfigure $n_{i-1,1} \rightarrow n_{i-1,2} \rightarrow n_{i,2} \rightarrow n_{i+1,2} \rightarrow n_{i+1,1}$. Similarly for others *\
4. **EndIf**
5. **If** edge is faulty then
6. Delete faulty edge
7. Reconfigure loop to bypass the faulty edge
8. **EndIf**

End ReconfigureOuter

ReconfigureInner (node)

1. Delete faulty node and the edges incident on it
2. Reconfigure loop to form an inner vertical loop using the nearest neighbors

End ReconfigureInner

Example. Consider a 5×5 microfluidic array (Figure 3) in which $n_{1,1}$ is both the source and sink. In Step 1, the faulty node in the outer loop is identified and reconfiguration is done to form another outer loop, as shown in Figure 3. In Step 2, the new outer loop is used in conjunction with the horizontal and

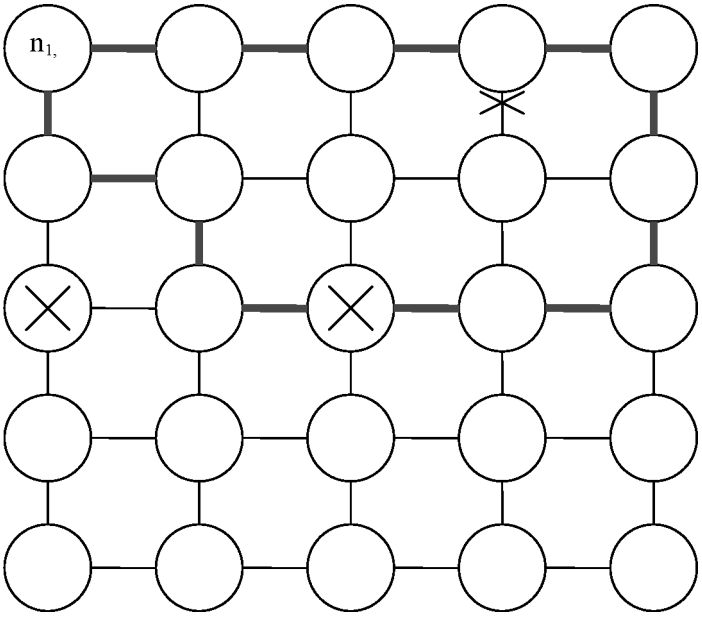

Fig. 4. Diagnosis of the second horizontal loop.

vertical loops. The internal faulty node is identified during the diagnosis of the second horizontal loop, as shown in Figure 4, and the appropriate reconfiguration is done. Figure 5 shows the second vertical loop without any fault, and the reconfigured loop once the faulty node has been identified. Finally, the faulty edge will be identified during the diagnosis of the third vertical loop.

The example illustrates the fact that it is possible to detect and locate a single fault in a closed tour in which at least three of the four sides are paths of the outer loop. Moreover, it also demonstrates the capability of the strategy to detect and locate multiple faults in the microfluidic array.

Now the complexity of the algorithm is analyzed. Let us assume that the size of the microfluidic array is m x n. It can be seen that in Step 1, if the outer loop is fault-free, then the number of edges traversed is $(2m + 2n - 4)$. If a fault is detected in the loop, then procedure FaultLocation is used a to identify the fault. FaultLocation essentially uses a binary search technique. The number of nodes (edges) in the outer loop is $(2m + 2n - 4)$. Thus, the number of steps executed in the worst case will be $\log_2(2m + 2n)$, while during each step, the number of edges traversed is bounded above by $(2m + 2n)$. Therefore, the complexity of fault detection and location in Step 1 is in $O((m + n)\log(m + n))$.

However, once a fault has been identified, a fault-free outer loop has to be determined to ensure the correct execution of Step 2. This reconfiguration is accomplished by the procedure Reconfigure and the new path that is included must be tested for any fault. It can be seen that in the worst case, the reconfiguration will traverse $O((m + n)^2 \log(m + n))$ edges. Thus, the number of edges traversed in Step 1 is in $O((m + n)^2 \log(m + n))$.

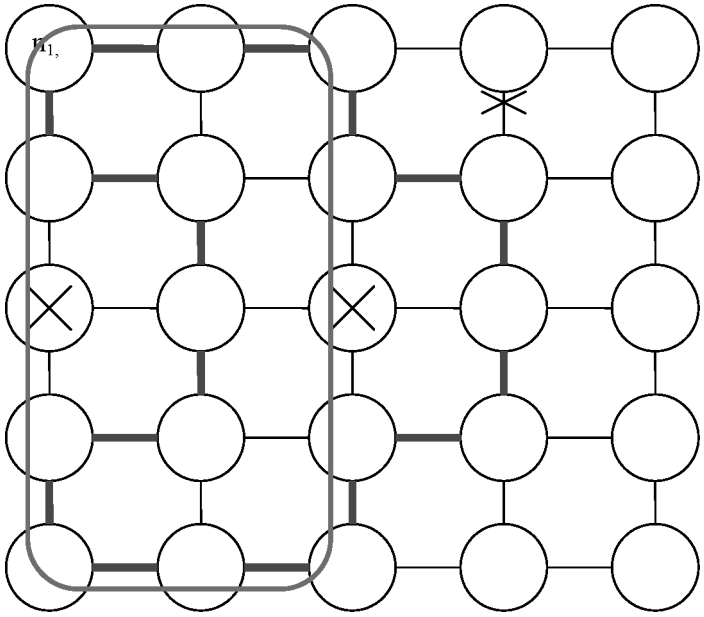

Fig. 5. Original and reconfigured second vertical loop.

In Step 2, since there is an overlap between fault detections among horizontal loops, the number of edges traversed is in $O(mn)$, but the number of time steps is in $O(m+n)$. Since the test droplets are introduced after every three units of time to avoid droplet mixing (due to the underlying physics of droplet transportation on the biochip), the last droplet will be dispensed after $3(m-3)$. Another $(2m+2n-4)$ time units are required for the droplet to traverse the loop. Thus, the total time units is in $O(m+n)$. Fault location in each loop is done sequentially. In the worst case, a fault has to be located in each of the m horizontal loops. It should be noted that the fault could be present only in the inner horizontal path, since the outer loop paths have been identified as fault-free. Thus, the binary search will be confined to only that inner horizontal path whose length is n. Therefore, the fault location procedure will be in $O(m(m+n)\log(n))$. Note that with appropriate scheduling of the test droplets, the fault location in various horizontal loops can be done in parallel, which will reduce the worst-case complexity of the procedure.

Now the performance of the proposed algorithm is compared with the strategy proposed in Su et al. [2005b]. Consider a 15×15 microfluidic array with a single fault at edge e_{2737}. While the proposed strategy takes about 296 units of time to identify the faulty edge, the strategy of Su et al. [2005b] takes 448 units of time to identify a fault region consisting of five nodes and four edges.

While these analytical results are based on worst-case analysis, it is critical to study the average time required to detect and locate faults. Computer simulations were conducted on a 15×15 microfuidic array and the results are shown in Figure 6. These results suggest that the average time required to detect and locate faults is linearly dependent on the size of the fault set. On

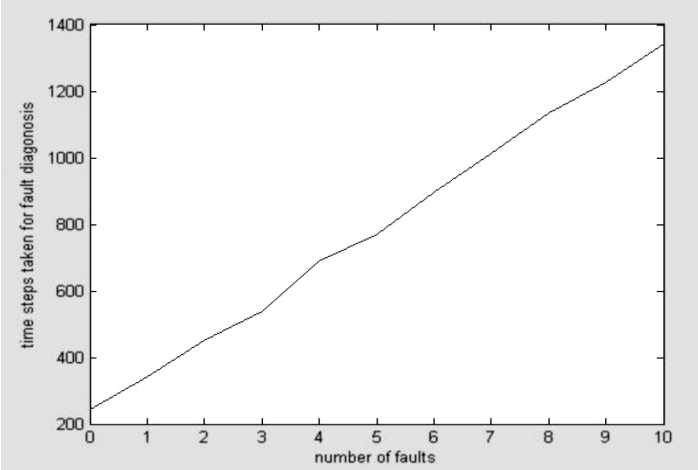

Fig. 6. Relationship between average time and fault set size for a 15 × 15 microfluidic array.

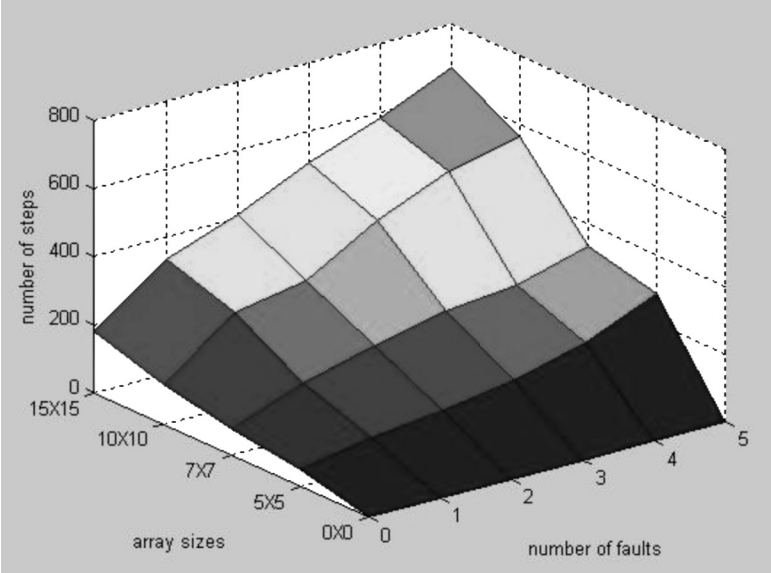

Fig. 7. Relationship between the microfluidic array size, fault set size, and average time.

average, the fault diagnosis of a faulty link takes a bit longer, since one more round of testing is required. For each size of, fault set, the time was averaged over 50 randomly generated fault sets. These fault sets included both node and link failures. Also, the faults were correctly detected and located in all cases.

Figure 7 shows the average time required to detect and locate faults for different array sizes. As in the previous case, for each size of fault set, the time was averaged over 50 randomly generated fault sets for each array size. It can be observed that for a fixed fault set size, the average time for fault detection and location increases with array size.

5. CONCLUSIONS

An efficient and effective fault diagnosis algorithm to detect and locate a class of multiple faults has been presented. One of main advantages of this algorithm is that it avoids flooding of the microfluidic array with test droplets. Flooding has been a critical problem in all strategies proposed in the literature. While retrieval of the test droplets is an integral part of the fault location strategy of the algorithm presented in the article, it leads to an efficient solution to flooding. In addition, the strategy is suited for BIST and highly applicable for quality assurance both at the time of manufacturing and during field operation. Furthermore, the algorithm can pinpoint the actual fault, as opposed to merely faulty regions, as typically identified by the strategies proposed in the literature.

Currently, efforts are being made to simultaneously retrieve the test droplets during fault location from several inner loops. This will further reduce the time complexity of the algorithm.

ACKNOWLEDGMENTS

Many thanks to Dr. Krishnendu Chakrabarty of Duke University for his valuable inputs during our discussions while this article was in preparation.

REFERENCES

Deb, N. and Blanton, R. D. 2000. Analysis of failure sources in surface-micromachined MEMS. In *Proceedings of the IEEE International Test Conference*. 739–749.

Deb, N. and Blanton, R. D. 2004. Multi-modal built-in self-test for symmetric microsystems. In *Proceedings of the 22nd IEEE VLSI Test Symposium*. 139–147.

Dhayni, A., Mir, S., and Rufer, L. 2004. MEMS built-in-self-test using MLS. In *Proceedings of the IEEE European Test Symposium*. 66–71.

International Technology Roadmap for Semiconductor (ITRS). 2003. http://public.itrs.net/Files/2003ITRS/Home2003.htm.

Kerkhoff, H. G. 1999. Testing philosophy behind the micro analysis system. In *Proceedings of the SPIE: Design, Test and Microfabrication of MEMS and MOEMS,* vol. 3680, 78–83.

Kerkhoff, H. G. and Hendriks, H. P. A. 2001. Fault modeling and fault simulation in mixed micro-fluidic microelectronic systems. *J. Electron. Test. Theory Appl. 17*, 427–437.

Kerkhoff, H. G. and Acar, M. 2003. Testable design and testing of micro-electro-fluidic arrays. In *Proceedings of the IEEE VLSI Test Symposium*. 403–409.

Kolpekwar, A. and Blanton, R. D. 1997. Development of a MEMS testing methodology. In *Proceedings of the IEEE International Test Conference*. 923–930.

Mir, S., Charlot, B., and Courtois, B. 2000. Extending fault-based testing to microelectromechanical Systems. *J. Electron. Test. Theory Appl. 16*, 279–288.

Pollack, M. G. 2001. Electrowetting-based microactuation of droplets for digital microfluidics. Ph.D. thesis, Duke University.

Pollack, M. G., Shenderov, A. D., and Fair, R.B. 2002. Electrowetting-based actuation of droplets for integrated microfluidics. *Lab on a Chip*, vol. 2. 96–101.

Srinivasan, V., Pamula, V. K., and Fair, R. B. 2004. An integrated digital microfluidic lab-on-a-chip for clinical diagnostics on human physiological fluids. *Lab on a Chip*. 310–315.

Su, F. and Chakrabarty, K. 2004. Architectural-level synthesis of digital microfluidics-based biochips. In *Proceedings of the IEEE International Conference on Coputer Aided Design*. 223–228.

Su, F. and Chakrabarty, K. 2005. Defect tolerance for gracefully-degradable microfluidics-based biochips. In *Proceedings of the IEEE VLSI Test Symposium*. 321–326.

SU, F. AND CHAKRABARTY, K. 2006a. Module placement for fault-tolerant micofluidics-based biochips. ACM *Trans. on Des. Autom. of Electron. Sys. 11*, 3, (Jul.) 682–710.

SU, F. AND CHAKRABARTY, K. 2006b. Yield enhancement of reconfigurable microfluidics-based biochips using interstitial redundancy. *ACM J. Emerg. Technol. Comput. Syst. 2*, 2, (Apr.) 104–128.

SU, F., OZEV, S., AND CHAKRABARTY, K. 2003. Testing of droplet-based microelectrofluidic systems. In *Proceedings of the IEEE International Test Conference*. 1192–1200.

SU, F., OZEV, S., AND CHAKRABARTY, K. 2004a. Test planning and test resource optimization for droplet-based microfluidic systems. In *Proceedings of the IEEE European Test Symposium*. 72–77.

SU, F., OZEV, S., AND CHAKRABARTY, K. 2004b. Concurrent testing of droplet-based microfluidic systems for multiplexed biomedical assays. In *Proceedings of the IEEE International Test Conference*. 883–892.

SU, F., OZEV, S., AND CHAKRABARTY, K. 2006. Concurrent testing of digital microfluidics-based biochips," *ACM Trans. Des. Autom. Electron. Syst. 11*, 2, (Apr.), 442–464.

SU, F., CHAKRABARTY, K., AND PAMULA, V. K. 2005a. Yield enhancement of digital microfluidics-based biochips using space redundancy and local reconfiguration. In *Proceedings of the Design, Automation and Test Europe Conference*. 1196–1201.

SU, F., HWANG, W., MUKHERJEE, A., AND CHAKRABARTY, K. 2005b. Defect-oriented testing and diagnosis of digital microfluidics-based biochips. In *Proceedings of the IEEE International Test Conference*, paper 21.2.

TEWKSBURY, S. K. 2001. Challenges facing practical DFT for MEMS. In *Proceedings of the Defect and Tolerance in VLSI Systems*. 11–17.

VERPOORTE, E. AND DE ROOIJ, N. F. 2003. Microfluidics meets MEMS. *Proc. IEEE 91*, 930–953.

Received May 2006; revised November 2006; accepted November 2006