A Novel Double Fault Diagnosis and Detection Technique in Digital Microfluidic Biochips

Sagarika Chowdhury, Rajat Kumar Pal, Goutam Saha

A Novel Double Fault Diagnosis and Detection Technique in Digital Microfluidic Biochips

Sagarika Chowdhury[1], Rajat Kumar Pal[2], and Goutam Saha[3]

[1]Dept. of Computer Science and Engineering, Narula Institute of Technology, Kolkata, India
[1]sagsaha2004@gmail.com

[2]Dept. of Computer Science and Engineering, University of Calcutta, Kolkata, India
[2]pal.rajtk@gmail.com

[3]Dept. of Information Technology, North Eastern Hill University, Shillong, India
[3]dr_goutamsaha@yahoo.com

Abstract. This paper presents a rigorous offline double fault diagnosis as well as a detection technique for Digital Microfluidic Biochips (DMFBs). Due to the underlying mixed technology biochips exhibit unique failure mechanisms and defects. Thus, offline and online test mechanisms are required to certify the dependability of the system. In this paper, the proposed algorithm detects double faults anywhere in the chip satisfying the dynamic fluidic constraints and improves the fault diagnosis time to an extent.

Keywords: biochip • droplet • LOC • micro-fluidic technology • fluidic constraints

1 Introduction

An integrated microfluidic device incorporates many of the necessary components and functionality of a typical room-sized laboratory onto a small chip [1]. These composite micro-systems, also known as lab-on-a-chip (LOC) (or bio-MEMS), offer a number of advantages over the conventional laboratory procedures and enable the handling of small amounts, e.g., micro- and nanolitres of fluids [2].

Droplet routing on the surface of the microfluidic biochip has been attracting much attention in recent years as it is one of the key issues to make use of the digital microfluidic device efficiently [3].

Microfluidic biochips have been characterized for the detection of faults [4], [5], [6], [7]. The test planning problem was formulated in terms of Euler circuit in [6], [7]. In [8], a functional testing has been proposed, referring to [4], [10], [11] that targets the functional operations of the microfluidic modules. Su *et al.* have proposed a defect tolerance based on graceful degradation and dynamic reconfiguration [9]. A network flow based routing algorithm has been proposed in [12] for the droplet routing problem on biochips [13]. An efficient diagnosis technique has been enhanced in [14] by Xu *et* *al.* such that multiple defect sites can be efficiently located using parallel scan-like testing. A more advanced multiple fault detection technique has been proposed in [16] by Chowdhury *et al.* in much less time compared to some previous techniques. However, it is not supported by the concept of Fluidic constraints, which has been explained in [15].

This paper addresses the issue of double fault diagnosis and detection technique in digital microfluidic based biochips through a graph-theoretic formulation.

The rest of the paper is organized as follows: Section 2 presents the preliminaries of microfluidic arrays, their defect characterization, and the graph-theoretic formulation. Section 3 explains the proposed technique. Experimental results are reported in Section 4. Conclusions are drawn in Section 5.

2 Preliminaries

2.1 Structure of a Microfluidic Array

In digital microfluidic biochips, each droplet can be independently controlled by the electrodynamic forces generated by an electric field [13]. Compared to the first generation biochips (analog), droplets can move anywhere in a 2D array to perform the desired chemical reactions, and electrodes can be re-planned for the different bioassays.

There are three key components in a biochip: 2D microfluidic array, dispensing ports/reservoirs, and optical detectors [13]. The 2D microfluidic array contains a set of basic cells, which handle droplet movement. The dispensing ports/reservoirs handle droplet generation, and the optical detectors are used for reaction detection.

2.2 Defect Characterization

As has been described in [4], faults in digital microfluidic systems can be classified as being either catastrophic or parametric. Catastrophic faults cause a complete breakdown of the system while parametric faults degrade the system performance. To detect a fault, we need to pass a droplet across the cells so that it can traverse the whole path and reach towards the sink. If there is any defect within the microarray, the droplet gets stuck there. Otherwise, it reaches the sink at a predefined time.

With the use of electrowetting phenomenon, droplets can be moved to any location in the given 2D array. However, it should be taken care that each of the droplets is satisfying the observable fact of fluidic constraints described in [15]. Only then the incidence of droplet collision is reduced.

3 Proposed Technique

Detection of a single fault in the biochip has reached at its end. However, diagnosis and detection of multiple faults are complex. Thus, let us start with more than one fault. Assume that, there are at most two faults in the biochip. The proposed technique takes a greedy approach to solving double fault detection technique. It tries to visit all the boundary nodes of $G_{m\times n}$ during the first and second pass (that are P1 and P2) starting from source to sink. Remaining edges and nodes are traversed in subsequent passes P3, P4, and so on through certain movement patterns, which are explained below.

3.1 Movement Patterns

Column traversal from the source to the sink is based on one kind of movement expressed in Fig. 1 as Down-Left-Down-Right (DLDR). Let $v_{i,j}$ be the current node during a traversal.

See Fig. 1(a) to understand the movement Down-Left-Down-Right (DLDR) as follows, where the value of i (j) increases from top to bottom (left to right):

$$v_{i,j} \xrightarrow{\text{D}} v_{i+1,j} \xrightarrow{\text{L}} v_{i+1,j-1} \xrightarrow{\text{D}} v_{i+2,j-1} \xrightarrow{\text{R}} v_{i+2,j}$$

The sink can be reached from the source by the following appropriate sequence of movements mentioned above. Example 1 presents such a journey from source to sink.

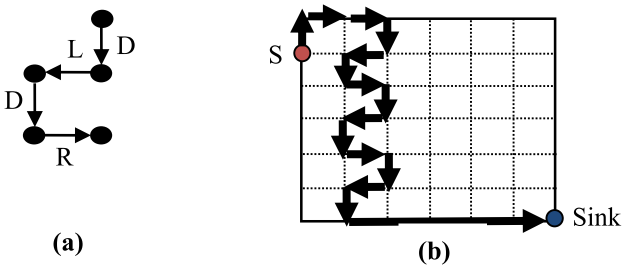

Fig. 1. (a) Movement patterns. **(b)** A path from source to sink following the proposed movement pattern of DLDR.

Example 1: Fig. 1(b) shows a graph dual $G_{7\times7}$ digital microfluidic array. The source is at position (2, 1), and the sink is at (7, 7). The graph has 36 nodes and 60 edges. The journey from source to sink with the proposed pathway is indicated with solid arrows.

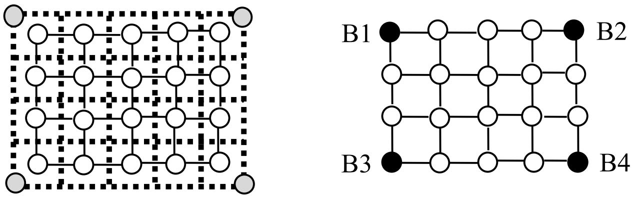

Fig. 2. Four base nodes at a graph dual of $G_{4\times4}$.

3.2 Strategy

We locate four fixed nodes as the 'base' nodes from where the journey with the prescribed movement patterns begins. For the graph dual $G_{m\times n}$ these base nodes are B_1 =

$v_{1,1}$, $B_2 = v_{1,n}$, $B_3 = v_{m,1}$, and $B_4 = v_{m,n}$. These four base nodes are shown in Fig. 2 for $G_{m\times n}$.

It is proposed that, irrespective of the source position, the traversal of the cells starts from any one of the nearest base point following the complete procedure.

3.3 The Complete Procedure

The complete process is as follows for a test array of size $M\times N$.

Procedure

Begin /* Assume source is at Base 1 and sink at (M, N) */

Step I: **Boundary Test 1:** A test droplet is dispensed from the source, and it traverses the boundary region clockwise and moves to the sink, as shown in Fig. 3(a).

Step II: **Boundary Test 2:** The second test droplet is dispensed with a delay of two units' time slice from the source and it traverses the boundary region anti-clockwise and moves to the sink, as shown in **Error! Reference source not found.**(b).

If BT1 and BT2 fail, it ensures at most two faults are at the boundary region. Hence, go for *Detect_Fault_UB* algorithm and *Detect_Fault_SB* algorithm. Skip steps three and four. Otherwise, proceed to step three.

Step III**: Row Test:** Two iterations of parallel scan-like test with one row shift are carried out, having the time delay of two units during dispense of each droplet. The test movement pattern of row test is shown in Fig. 4. Thus, in iteration 1, all the even rows are traversed and in iteration 2 all the odd rows are covered. After the first iteration, there should be a delay of two units' time.

After row test, the defected rows can be identified easily, following the arrival report of the droplets at the sink.

Now to further *reduce* the testing area of the chip, let us identify the region of the affected rows following algorithm *Reduce_Area* (Fig. 5).

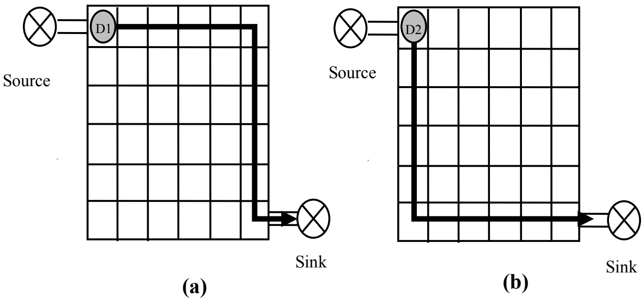

Fig. 3. (a) Boundary Test I, where droplet #1 and time slice T_1. **(b)** Boundary Test II where droplet #2 and time slice T_4.

Algorithm *Reduce_Area:*

1. If the affected rows are in the upper region, i.e. between row #1 to floor (M/2) of the chip, then divide it into two halves in such a way, so that, the divider passes just next to the one row, which is affected last (Fig. 5(a)).
2. Else if the affected rows are in the lower region, i.e. after floor (M/2) row number of the chip, then divide the chip into two halves in such a way, so that the divider passes just before the row, which is affected first (Fig. 5(b)).
3. Else divide the chip into two halves keeping faulty rows in both the halves (Fig. 5(c)), so that column test can be carried out in those two halves in parallel.

If the row test is satisfactory but BT1 and/or BT2 are not adequate, then consider Case 1.

Else if, row test is not agreeable, but BT1 and BT2 are satisfactory, then consider Case 2 and go to Step IV.

Else if row test is not acceptable and so for BT1 and/or BT2, then consider Case 3 and go to Step IV.

Else it ensures that the row test, BT1 and BT2 all are good enough. Thus, the chip is free from any fault.

Step IV: **Column Test:** Delay for two units. Repeat parallel scan-like test (for two iterations) for the columns, following DLDR movement patterns. Three units of time delay against dispersion of every droplet and four units of time delay during second iteration are required. These movements are explicitly shown in Fig. 6.

End Procedure

3.4 Analysis and Detection of Faults

The total time required for the entire process is the sum of the required time for Step I, Step II, Step III, and Step IV if they are carried out sequentially. Step III and Step IV are to be performed if only they are required.

As there can be double faults anywhere in the chip, including *boundary*, let us discuss their possible positions on the chip sequentially.

Case 1: Assuming two faults are anywhere at the boundary, i.e. the rest of the chip is fault free. Thus, the faults may be in the following locations:

- Two faults at the upper/lower boundary, or
- Two faults at the right/left side boundary, or
- One fault at the upper/lower boundary, other at the side boundary, or
- One fault at the upper boundary, other at the lower boundary.

If BT1 fails, but BT2 succeeds, then there may be one / two faults at the boundary. Consequently, go for Row Test. If Row Test succeeds, follow *Reduce_Area* algorithm described earlier. Then go for the following:

1. Consider sink at the end of the first row and perform boundary checking, starting from source to sink.

2. If the droplet does not reach, then the upper boundary region is faulty.

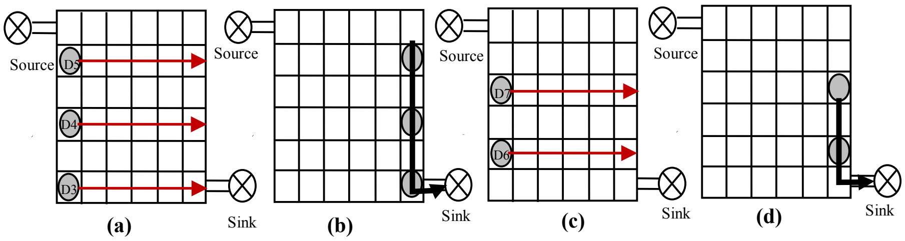

Fig. 4. Row_Test. **(a)** Iteration 1: droplet #3 at time slice T_7, droplet #4 at time slice T_{10}, and droplet #5 at time slice T_{13}. **(b)** Three droplets follow the last column to the sink. **(c)** Iteration 2: droplet #6 at time slice T_{18} and droplet #7 at time slice T_{21}. **(d)** Two droplets follow the last column to the sink.

3. Follow algorithm *Detect_Fault_UB*.
4. Next, delay for two units of time and send another droplet, from second row towards the end of that row and then downwards to sink, following side boundary. If it reaches properly, no fault in the right side region. Otherwise, there is a fault. Then follow algorithm *Detect_Fault_SB* .

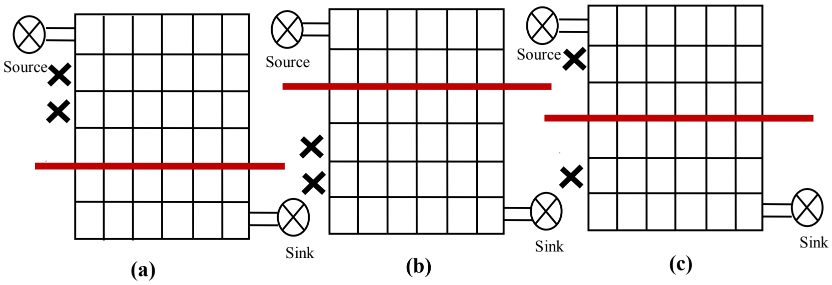

Fig. 5. Chip divider algorithm. **(a)** Affected rows are in the upper region. **(b)** Affected rows are in the lower region. **(c)** Affected rows are in any of the regions.

Algorithm ***Detect_Fault_UB: /* Detect Fault(s) in Upper Boundary region*/***
Assume the source at (1, 1) location and the sink at (M, N) location.

1. Disperse a droplet from source to the second column.
2. Go downwards by one row.
3. Go to the last column of that row and then to the sink.
4. After dispensing a droplet, a next droplet is dispersed after two units' delay.
5. Repeat Steps 1 through 4 incrementing the column number by one until a droplet fails to reach the sink in proper time.
6. When a droplet does not reach to the sink, then the interleaving column of the first row is detected as the *defective* cell.
   Once an affected cell is found, do the following:

7. Send the next droplet just before to the affected cell.
8. Go downwards by one row.
9. Follow the right way and go upwards, so that the very next cell to the affected one at the boundary get touched.
10. Go to the last column of that row and then to sink.
11. After dispensing a droplet, a next droplet is dispersed after two units' delay.
12. Repeat Steps 7 through 11 as shown in Fig. 7, until a droplet succeeds to reach the sink in proper time.

Following the last droplet that may fail to reach the sink in time, we can straightforwardly identify the *defective* cell.

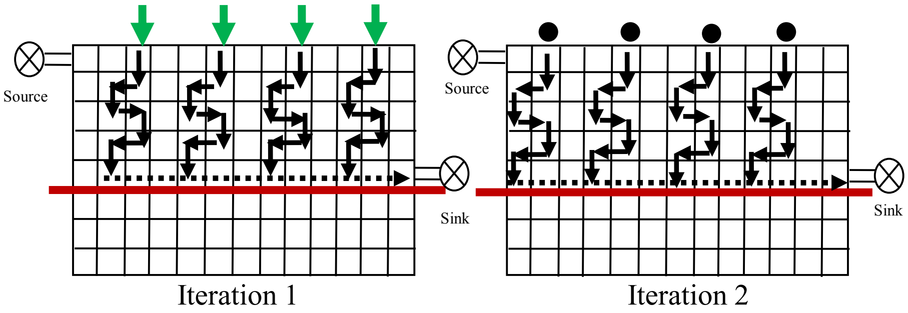

Fig. 6. Column_Test following *DLDR* movement patterns.

Algorithm *Detect_Fault_SB: /* Detect Fault in Side Boundary region */*

1. If the upper boundary is satisfactory, but the side boundary is not all right, then make source at (1, n–1) location and follow algorithm *Detect_Fault_UB* to identify fault at the side boundary.
2. Else if the upper boundary is faulty as well as the side boundary, then go for Row Test and detect the faulty row.

It is justified that the intersecting points of row and side boundary are treated as the faulty cell, as we have assumed that there can be at most two faults in the chip.

E.g., if the fault is at (M–3, N) location, then more than one droplet set for row test must fail to reach the sink. In that case, for the last droplet, which fails to reach the sink, the intersecting point of row path and side boundary can be detected as faulty (Fig. 8).

Case 2: Boundary is passable. Thus, two faults are anywhere in the chip, other than the boundary.

Go for the Complete Procedure as discussed in Section 3.3. After performing column test, we can have several scenarios to be discussed further.

1. If two adjacent column tests fail to pass droplets against one affected row, then there must be one common intersecting point, which is affected definitely (Fig. 9(a)).

2. Else, if two adjacent column tests fail to pass droplets, against two of the affected rows, then we are getting six considerable points among which two may be faulty (Fig. 9(b)). Thus, detection can be done afterward. Fig. 10 shows it clearly.

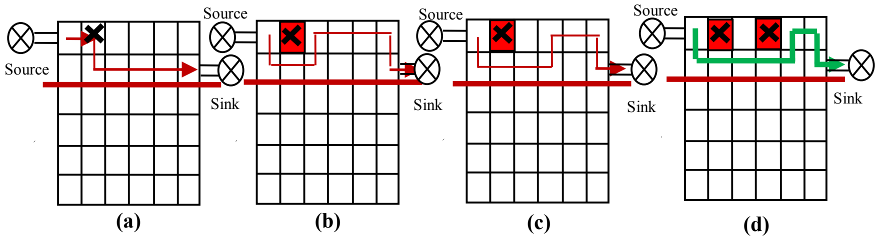

Fig. 7. Detection of double faults at the upper boundary of the chip. **(a)** First droplet fails to reach the sink, so the fault is at cell #2. **(b)** Second droplet fails to reach the sink, but cannot say whether the fault is on cell #3. **(c)** Third droplet fails to reach the sink, but cannot say whether the fault is on cell #4. **(d)** Fourth droplet succeeds to reach the sink. Thus, cell #4 is certainly faulty.

3. Else if one column test fails to pass droplet against one affected row, then two considerable points are there, of which one has not gone for column test. Thus, that non-traversed point is the faulty one (Fig. 9(c)).
4. Else if one column test fails, against two of the affected rows, then there are exactly two infected cells to be identified for sure (Fig. 9(d)).

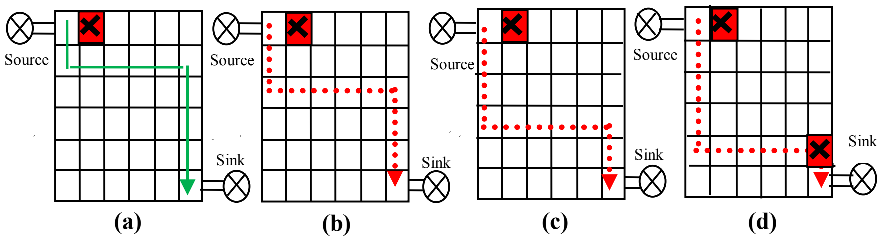

Fig. 8. Traversing through (a) to (d) while detecting the fault at the side boundary.

Case 3: One fault is at the boundary and the other is anywhere in the chip.

Consider an example 2, where the graph dual of $G_{7\times7}$ depicts that it has an error at cell #2, i.e. on the upper boundary and at cell #25. Now, according the proposed method, BT1 will fail while BT2 will succeed to reach towards the sink in specified time. Therefore, we go for Row test and definitely, we will get row #4 as faulty. Hence, after performing algorithm *Reduce_Area*, it looks like the picture as shown in Fig. 11(a).

Now, perform algorithm *Detect_Fault_UB* and it will detect a fault at cell #2.

At this instant, droplet #D_1 of Column Test at Iteration 2 fails to reach the sink. Thus, the intersecting point of row #4 and the path traversed by D_1 during column test at iteration 2 is cell #25, as discussed in *Case 2(b)*. Fig. 11(b) shows this clearly.

The detection time is compared with some existing technique. It shows that the proposed one is much superior compared to the other. Moreover, the proposed technique satisfies the dynamic fluidic constraints as well.

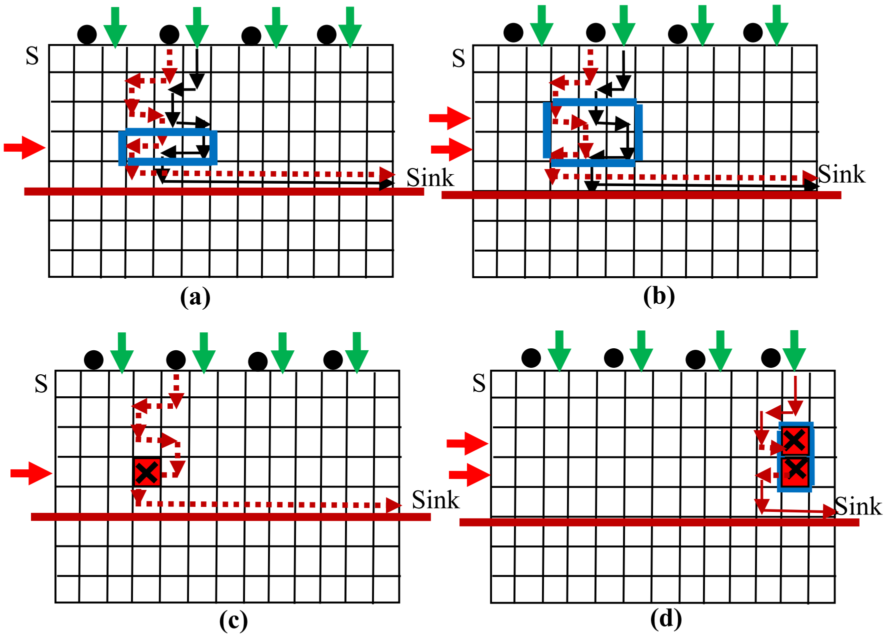

Fig. 9. Fault analysis for Case 2, where ➡ indicates affected row, ➡ indicates column traversed in the first iteration, and ● indicates column traversed in the second iteration.

4 Experimental Results

Extensive offline testing has been done with a large number of arrays varying from 4×4 to 60×60 electrodes. Table I reports the details of the microarray. Table II reports the performance of the proposed technique, viz. the existing technique [14]. The performance of the proposed technique is divided into two columns: Proposed (min) and Proposed (max).

The proposed method not only diagnoses double faults in the chip, but it also detects the location. To test N×N target array, Boundary Test 1 and Boundary Test 2 are carried out first, and the methods take 2N+2 units' time. Next, row test is performed; in each iteration, this takes N units of time. Thus, up to this step, the proposed technique takes 4N units' time slice. After that, the N×N target array be partitioned into two halves, and the column test is performed in parallel. Here, if one has to go for column test, then it takes 2N/2, i.e. N units time slice in each iteration. Hence, as a whole it takes 6N units time slice. Therefore, the total fault diagnosis procedure includes 6N steps, i.e. O(N), and compared to parallel scan-like test and multiple defect diagnosis [14], which has 8N steps, the time needed for this diagnosis is reduced.

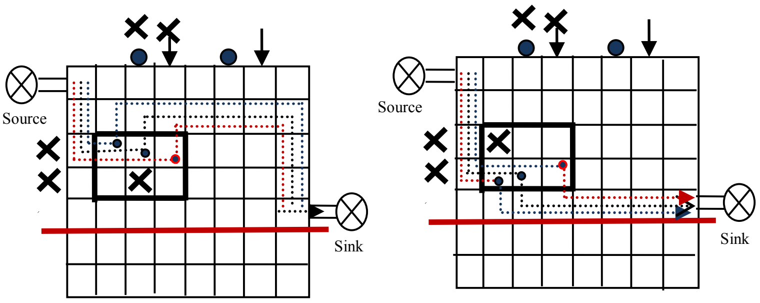

Fig. 10. Detection of double fault for Case 2(b).

Columns 6 and 7 of Table II show the percentage (%) improvement in each case. This is defined as:

% improvement = ((Existing Time – Proposed Time) × 100) / Existing Time

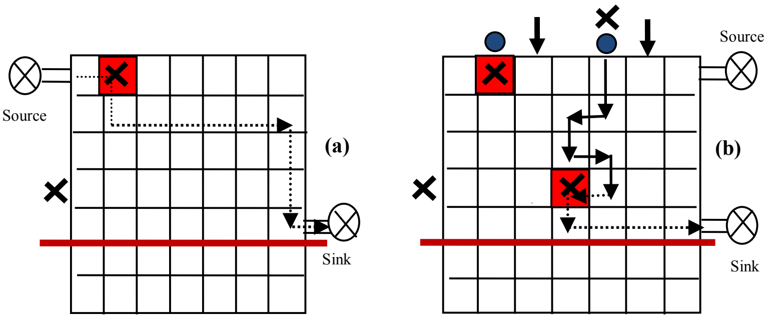

Fig. 11. Double Fault Detection through Example 2 for Case 3.

It can be seen that the proposed method has an improvement over time at about 12.5% minimum and 37.5% maximum. Table II shows the case-wise result, as discussed earlier. It shows that the proposed technique achieves improvement to a great extent.

Table 1. Test case details

Sl. No.	Size	Source	Sink	Total Edge
1	4×4	1, 1	4, 4	24
2	5×5	1, 1	5, 5	40
3	6×6	1, 1	6, 6	60
4	10×10	1, 1	10, 10	180
5	20×20	1, 1	20, 20	760
6	20×25	1, 1	20, 25	980
7	30×30	1, 1	30, 30	1740
8	40×40	1, 1	40, 40	3120
9	50×50	1, 1	50, 50	4900
10	60×60	1, 1	60, 60	7080

Table 2. Performance of the proposed technique

Sl. No.	Time Slice Required							% improvement in Diagnosis	
	Existing [14]	Proposed							
		Case 1		Case 2		Case 3			
		Diagnosis	Detection	Diagnosis	Detection (max)	Diagnosis	Detection	(Min)	(Max)
1	32	20	24	24	32	28	28	12.5	37.5
2	40	25	30	30	40	35	35	12.5	37.5
3	48	30	36	36	48	42	42	12.5	37.5
4	80	50	60	60	80	70	70	12.5	37.5
5	160	100	120	120	160	140	140	12.5	37.5
6	200	125	150	150	200	175	175	12.5	37.5
7	240	150	180	180	240	210	210	12.5	37.5
8	320	200	240	240	320	280	280	12.5	37.5
9	400	250	300	300	400	350	350	12.5	37.5
10	480	300	360	360	480	420	420	12.5	37.5

5 Conclusion

Efficient fault detection in a microfluidic biochip is an indispensable activity, as it is often used to operate in critical circumstances. In this paper, an advanced double fault detection technique has been proposed, which yields better results compared to some existing methods. Also, it satisfies the phenomenon of dynamic fluidic constraints. Thus, the chances of collision between two droplets are reduced. The proposed algorithm can detect any fault in the boundary region too. Therefore, further case studies can be done so that, not only double fault at the boundary, but more than two faults can be identified.

References

1. D. Erickson and D. Li, "Integrated Microfluidic Devices," Proc. Elsevier, Analytica Acta 507, 11-26, (2004).
2. K. Chakrabarty and F. Su, "Design Automation Challenges for Microfluidics based Biochips," DTIP of MEMS & MOEMS, Montreux, Switzerland, June 1-3, (2005).
3. X. Zhang, F. Proosdij, and G. Kerkhoff.,"A Droplet Routing for Fault-Tolerant Digital Microfluidic Devices," IEEE, (2008); The research has been partly funded by DfMM, FP6 NoE PATENT, in the framework of the BioDrop Flagship.
4. F. Su, S. Ozev and K. Chakrabarty, "Testing of Droplet based Microfluidic Systems," Proc. IEEE Int. Test Conf., pp. 1192-1200, (2003).
5. F. Su, S. Ozev, and K. Chakrabarty, "Test Planning and Test Resource Optimization for Droplet based Microfluidic Systems," Proc. IEEE Eur. Test Sym., pp. 72-77, (2004).
6. F. Su, H. Hwang, A. Mukherjee, and K. Chakrabarty, "Testing and Diagnosis of Realistic Defects in Digital Microfluidic Biochip," Proc. Springer Science + Business Media, (2007).
7. V. Srinivasan, V. Pamula, M. Pollack, and R. Fair, "A Digital Microfluidic Biosensor for Multianalyte Detection," Proc. IEEE MEMS Conference, pp. 327-330, (2003).

8. F. Su and K. Chakrabarty, “Defect Tolerance based on Graceful Degradation and Dynamic Reconfiguration for Digital Microfluidics based Biochips,” IEEE TCAD, vol. 25, no. 12, (2006).
9. T. Xu and K. Chakrabarty, “Functional Testing of Digital Microfluidic Biochip,” ITC (2007).
10. F. Su, S. Ozev, and K. Chakrabarty, “Ensuring the Operational Health of Droplet based Microelectrofluidic Biosensor Systems,” IEEE Sensors, vol. 5, pp. 763-773, (2005).
11. F. Su, S. Ozev, and K. Chakrabarty, “Concurrent Testing of Droplet based Microfluidic Systems for Multiplexed Biomedical Assays,” Proc. Int. Test Conf., pp. 883-892, (2004).
12. H. G. Kerkhoff, “Testing of microelectronic-biofluidic systems,” IEEE D&T for Computers, vol. 24, pp. 72-82, (2007).
13. Ping-Hung Yuh, Chia-Lin Yang, and Yao-Wen Chang, “BioRoute: A Network-Flow-Based Routing Algorithm for the Synthesis of Digital Microfluidic Biochips,” IEEE TCAD, vol. 27, no. 11, pp. 1928-1941, (2008).
14. T. Xu and K Chakrabarty, “Parallel Scan-Like Test and Multiple-Defect Diagnosis for Digital Microfluidic Biochips,” IEEE Transactions on Biomedical Circuits and Systems, vol. 1, no. 2, (2007).
15. P. Roy, H. Rahaman, C. Giri, and P. Dasgupta, “Modelling, Detection and Diagnosis of Multiple Faults in Cross Referencing DMFBs,” Proc. IEEE ICIEV, pp. 1107-1112, (2012).
16. S. Chowdhury, S. Majumder, K. Mondal, “Multiple Fault Detection Technique for Digital Microfluidic based Biochip,” Proc. IET & Int. Conf. ArtCom, pp. 117-125, (2013).